AGILE: a methodology for Advanced Governance of Information services through Legal Engineering

Boer, A.; Sileno, G.

Publication date
2013

Document Version
Final published version

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
AGILE: a methodology for Advanced Governance of Information services through Legal Engineering

Alexander Boer, Giovanni Sileno

31 December 2013
Contents

1 Introduction 2
  1.1 The Agile project 2
  1.2 The Agile methodology 3
  1.3 Overview of this document 3

2 Basic concepts and principles 5
  2.1 Agility and law 5
  2.2 On agility 5
  2.3 Service, and service-oriented architecture 7
  2.4 Law and services 7
  2.5 Knowledge level agility 11
  2.6 Agility as diagnostic ability 15

3 Overview of the methodology 19
  3.1 Introduction: the real, actual, and empirical 19
  3.2 How agents use the law 22
    3.2.1 Normative agent roles 24
    3.2.2 Exercising legal power 27
    3.2.3 Information, evidence, and argumentation 28
    3.2.4 Operations, development, and policy making 29
    3.2.5 Context and knowledge engineering 30
    3.2.6 Summary of principles 31
  3.3 Maintaining the sources of law 32
    3.3.1 MetaLex and Bibliographic Identity 33
    3.3.2 Identity of Legal Concepts and Rules over Time 34
    3.3.3 Actions performed on Documents 34
    3.3.4 Applicability 35
    3.3.5 A demonstrator for MetaLex 36
    3.3.6 Summary of principles 36
  3.4 Maintaining models of regulated social systems 36
    3.4.1 Summary of principles 37
  3.5 Narratives and diagnosis 38
    3.5.1 Summary of principles 39
4 Legal Knowledge from an Engineering Perspective

4.1 Introduction .................................................. 43
4.1.1 Intention and Action ..................................... 47
4.2 Institutions and Rules ....................................... 50
4.2.1 Constitutive Acts, Speech Acts, and Intent .......... 52
4.2.2 Law as an Institution ................................... 54
4.2.3 Constitutive Rules and Ampliative Reasoning ...... 57
4.3 Norms, Normality and Normativity ....................... 58
4.3.1 Changing Legislation .................................... 62
4.3.2 Representing Normal Behaviour ......................... 64
4.4 Legal Normative Rules .................................... 66
4.4.1 Subjunctive Betterness .................................. 70
4.5 Agents and Action .......................................... 73
4.5.1 To Be or To Do .............................................. 74
4.5.2 Agent Causation and Time .............................. 78
4.5.3 Positions and Power .................................... 82
4.6 Interaction Between Agents ................................ 83
4.6.1 Transactions and Interests .............................. 85
4.6.2 Hohfeld’s Conception of Rights and Powers ....... 86
4.6.3 Everyone and Delegation ................................ 90

5 Managing a Corpus of Legislation .......................... 92
5.1 Introduction .................................................. 92
5.1.1 Auxiliary Rules and Facts ............................... 93
5.1.2 MetaLex ..................................................... 94
5.2 Sources of Law and Legal Rules ......................... 94
5.2.1 Representation of Legal Rules ......................... 99
5.2.2 Application of Rules .................................... 102
5.2.3 Representation of Applicability Rules ............... 103
5.2.4 Also-applicability and Legal Fiction .................. 104
5.2.5 Purposes of Rules ....................................... 105
5.3 Sources of Law and the Semantic Web .................. 108
5.3.1 Metadata and Legislative Occurrences ............... 110
5.3.2 MetaLex XML .............................................. 116
5.3.3 Individuation and Identification of Sources of Law .. 121
5.3.4 Citation, Inclusion, and Reference .................... 126
5.4 Knowledge Components and Sources of Law ............ 130
5.4.1 Knowledge Engineering Considerations .............. 132
5.4.2 Applicability and Knowledge Sources ............... 134

6 Modeling techniques for legal knowledge ............... 137
6.1 Narratives ................................................... 137
6.1.1 Operational definitions .................................. 137
6.1.2 Ontological layers ...................................... 138
6.1.3 Functional perspective .................................. 139
6.1.4 Narratives through critical realism .................. 139
6.1.5 Informational interest .......................... 140
6.1.6 Legal narratives ............................... 140

6.2 Argumentation ................................. 140
6.2.1 Argumentation as a narrative act ............. 141
6.2.2 Argumentation and inference to best explanation 141
6.2.3 Claims as manipulations of the explanatory space 142

6.3 Reasons and rules ............................. 142
6.3.1 Representation ................................ 143
6.3.2 Planning and projection ....................... 143
6.3.3 Justification ................................ 144
6.3.4 Obligation and power as reasons .............. 146
6.3.5 Subjective, objective perspectives ............. 146

6.4 Multi-agent systems ........................... 147
6.4.1 Agent-roles .................................. 147
6.4.2 Services and work flows ....................... 147

7 Model-based problem solving and the legal system 148
7.1 Introduction ..................................... 148
7.1.1 Spheres of activity ........................... 148
7.1.2 Generic Problems in Public Administration .......... 149
7.1.3 Cybernetic Control Models of Organizations ........... 153
7.1.4 Functions of Knowledge in Problem Solving Activity ........ 154
7.1.5 Model-based diagnosis ........................ 155
7.1.6 Multi-agent Systems and Coordination of Problem Solving Activity ........ 156
7.1.7 Conclusion .................................. 157

7.2 A formal model of problem solving with agent roles 157
7.2.1 Multi-agent system and agent-roles ............. 158
7.2.2 Scenario agent ................................ 158
7.2.3 Observation ................................ 158
7.2.4 Explanation ................................ 159
7.2.5 Modeling problem ............................ 159
7.2.6 Design or planning problem .................... 160
7.2.7 Monitoring problem ........................... 160
7.2.8 Diagnosis problem ............................ 160

7.3 Allocation and rule-based reasoning methods 160
7.3.1 Default rules ................................. 162

7.4 Probability-based reasoning methods ............. 163
7.4.1 Messages .................................... 163
7.4.2 Space of explanatory hypotheses ................. 164
7.4.3 Prior probability of explanations ................. 164
7.4.4 Probability of observations .................... 165
7.4.5 Surprisingness of observations ................ 165
7.4.6 Evaluating explanations ....................... 165

7.5 Reasoning about argumentation structures ......... 166
7.5.1 Disadvantages of argumentation structures ....... 166
Abstract

To address agility in public administration, the Agile project developed a reference knowledge acquisition infrastructure for legal knowledge, based on a dynamic and design-oriented conceptualization of the legal system. The main objective of the project was to reframe legal knowledge as a knowledge source in a design-oriented task ontology, building on insights from the CommonKADS methodology for intelligent system design. The methodology makes case law, and legal expert knowledge of critical incidents in organizations – two diagnostic knowledge sources underutilized in modern management and engineering – more accessible as a resource for design of agile organizational structures and intelligent systems.
Chapter 1

Introduction

1.1 The Agile project

The work reported in this report was performed in the context of the Agile project (acronym for Advanced Governance of Information services through Legal Engineering). Agile aimed to develop concepts helping administrative organizations to reduce the time from a request for changes based on a change in the relevant law to implementation in the organization. The project developed models that are intended to help administrative organizations in developing alternative implementations, estimating required effort and time, identifying risks and opportunities, providing useful and accurate feedback to the legislator, and generally to change from a project-centered approach to a maintenance-centered approach to dealing with change of the law.

Involved in the Agile project were the Dutch Immigration and Naturalisation Service (IND) and the Netherlands Tax and Customs Administration (NTCA). In both organizations, timely and efficient adaptation to changing legislation, case law, and patterns of behaviour, accommodating or evading law in the relevant environment, is seen as an important organizational objective. One whose realization is a constant cause of problems.

Immigrants and taxpayers are capricious clients – they quickly adapt to make the most of the rules in force – and regulatory policy making is a constant concern. The IND and DTCA have to reinvent themselves continually, and sometimes move to have the law changed, in response to problems or opportunities arising from their environment.

In the Agile project we aimed at developing a knowledge representation and design methodology, a distributed service architecture simulation environment, and supporting tools for legal requirements and knowledge representation. This document is the main result of the project.
1.2 The Agile methodology

The main deliverable of the Agile project, as documented here, is a methodology for legal knowledge management. A methodology is the formal documentation of a set of procedures for a given field, documenting and explaining the reasons for following them. It gives structure and leads us toward a result. The methodology should lead us through any point in the modeling process by constantly answering two questions: what do we do next and why are we doing it?

The methodology is strictly separated from the other, closely related, products of the project: for instance the reference architecture for service-oriented architecture, or the tools we have developed to animate legal knowledge in a multi-agent system.

The methodology is, moreover, not accompanied by a new special purpose modeling language, and real world deployment of knowledge models is not an object of the project. Actual technical implementation has to take into account the existing technical infrastructure of an organization, including the existing commitments an organization has to modeling techniques and languages, and the modernization of infrastructure or selection of delivery platforms is not the focus of the project. The suggestion that an organization that wants to adopt the methodology has to redesign its infrastructure harms the case for agility rather than helping it.

Agile, law-aware, modeling can be achieved through general purpose modeling languages for software engineering and business process modeling, like UML. The methodological guidelines in this document are therefore often discussed in terms of general purpose modeling techniques for scenarios, work flows, rules, and structures.

1.3 Overview of this document

The next chapter, chapter 2, lays the foundations for the methodology by addressing the main “why?” question: why is a knowledge acquisition methodology the key to organizing agility in organizations in their interactions with the law? It addresses our views of the agility issue, in relation to law and the implementation of services. It translates the general principles of agility in response to a changing legal environment to a matter of diagnosing narratives, and knowing implementations and uses of rules by intelligent agents.

In chapter 3 we give an overview of a methodology for agility through legal knowledge engineering, and introduces some important concepts and considerations from legal theory and computer science & law, directed at readers with backgrounds in software engineering, knowledge engineering, and business process modeling. We distinguish an 1) operational, 2) development and 3) policy making perspective on law, each suggesting their own perspective, and desiderata for modeling.

Chapter 4 is background material on modeling the legal system, without taking into consideration the analysis of written legislation or case law. Law
is mainly analyzed in terms of changes of legal positions. This is a powerful analysis technique, from the point of view of legal interpretation of individual cases, but it is a limiting perspective as well, more useful in the court room than in the development department. This chapter is added for the benefit of readers without previous exposure to modeling law.

Chapter 5 is background material on interpreting written rules found in legislation, and the more general problem of keeping organizational specifications synchronized with changing legislation. This chapter too is added for the benefit of readers without previous exposure to modeling law.

Chapter 6 has an auxiliary function. It presents a number of modeling techniques for use in the chapters following it, and addresses the reasons for choosing certain modeling techniques.

Having set up the techniques and concepts, chapter 7 presents a generic model of (legal) problem solving in organizations. The central concept of this chapter is model-based diagnosis. The chapter aims to show how legal interpretation, when performed by designers and policy makers, is a model-based diagnosis problem. The corresponding knowledge acquisition problem is the problem of systematically mapping out the diagnostic hypothesis space, and the information that can function as evidence distinguishing between hypotheses. The problem solving model explains the complementarity between the rules in the sources of law, as constraints and requirements in design, and narratives, from case law, or hard cases inside the organization, as a source of experience in the organization.

In chapter 8 we address the first half of the legal knowledge acquisition problem: managing sources of law and models of the rules found in them.

In chapter 9 we address the other half: the feedback we get from our efforts to design and implement our interpretations of the sources of law into the organization. This is the aspect of legal knowledge acquisition that is generally underdeveloped, and the major area in which the Agile methodology adds to the state-of-the-art in modeling the law.

Chapter 10 presents some implementations of Agile models in multi-agent system for animation. The first example, hunting foxes, is a classic in computer science & law literature, and therefore an interesting test case of the methodology. The second example, tax evasion schemes using real estate sales transactions, is a realistic example, based on a business concern of the NTCA in need for a better implementation. This domain model remains under development, and will be the subject of a future PhD thesis (Bernd Veldman).

Chapter 11 briefly summarizes the practical results of the project, and does not form part of the main body of this report.
Chapter 2

Basic concepts and principles

2.1 Agility and law

In the context of the present text, the agile organization typically belongs to public administration, and it delivers services based on, and constrained by, the law. The general principles of the methodology proposed here apply to any organization with non-negligible legal responsibilities and interests, however.

The law, in this setting, is manifested by language, distinguished (by form) into two main types: rules and narratives. Rules are prescriptive statements, of a general character, making legal sense of social interactions. Rules are generally found in formal legislation, and regulations addressing smaller social circles, like an organization, the parties of a written agreement, a sector of the economy, etc. Court decisions may formulate general rules too, abstracting from a specific interpretation of a case at hand. These rules are recognized because the regulatory policy maker that creates them has a recognized authority to do so.

Narratives share the legal interpretation of, and reaction to, a specific case. Court decisions are grounded in a narrative. Formal complaints and critical incidents are shared as narratives as well. Narratives are raw material. The job of generalizing the narrative into rules is left to the unauthorized interpreter.

Both may affect service delivery. Both originate, at least to some extent, in the environment, and may take an organization by surprise. Both may require an urgent response by the organization, calling for agility.

2.2 On agility

Agility, when applied to organizations, is the ability to adapt rapidly and efficiently in response to changes in the environment. The agility concept incorporates ideas of flexibility, of balance, and of coordination. As an ability, it refers
to a continuing state of readiness for action, but without an identified purpose. The agile enterprise is an extension of this concept, referring to an organization that utilizes principles of complex adaptive systems (CAS) to achieve agility.

**Proposition 2.2.1. Agility is the ability to adapt rapidly to changes in the environment.**

Agility means different things in the context of different systems. We therefore need to frame our use of the concept by differentiating some of these systemic contexts, and choosing our system boundaries.

In financial terms, agility for instance translates to the ability to convert current assets into cash quickly, by minimizing *keeping-the-lights-on expenses*. A financially agile business can make cash available to respond to changes in the business environment while keeping the lights on. Cash is treated as a proxy of the ability to act quickly. From this perspective, one should guard against optimism about the ability of an organization to address problems posed by the environment. Overinvestment in infrastructure based on overly optimistic estimates of the *return on investment* time window should be avoided from this perspective.

Obviously, infrastructure may itself be more or less agile, in the sense of being able to address problems the environment poses, and therefore more or less deserving of investment. In business process design (BPD), for instance, agility is operationalized as *intelligently managing degrees of freedom in executing a work flow by offering alternative execution paths to deal with problems and opportunities*. It can be addressed 1) as an operational problem, of foreseeing problems and opportunities, and selecting a potential execution path for such scenarios, and 2) as a design problem, of quickly adding an execution path for a newly discovered scenario, without interrupting operations.

In ICT, agility typically translates to a focus on continuity of availability of ICT-based services, and the use of shared resources as a service provided by third parties, governed by service-level agreements spelling out expectations about availability and performance of those shared resources. Agility in this narrow sense mainly services financial agility, and its reservations against over-investment in infrastructure.

In the context of agile programming, agility is about quickly adapting information systems in short development cycles, without interruption in service, in line with BPD definitions of agility.

In the Agile project we address agility in its basic sense, as a continuing readiness for action, based on knowledge about the service framework the organization operates in, and the threats to its function. The essential question:

**Proposition 2.2.2. The Agile knowledge acquisition question is “What knowledge structures serve the cause of organizational agility?”**
2.3 Service, and service-oriented architecture

The service-oriented architecture (SOA) paradigm, and its derivative cloud computing, often address agility in the BPD and ICT spheres as a mainly infrastructural problem, but the SOA service concept itself takes its meaning from a wider social and economic context. The Agile methodology addresses a specific form of that social and economic context of services: the social-legal framework that gives those services their legal meaning. It is therefore important to understand the meaning of service in its wider context, and to consider agility of service delivery in this wider context.

The service concept is the intangible equivalent of the economic good, one that is produced and consumed simultaneously. It shares some fundamental characteristics with economic goods: 1) a service must recognizably benefit the service consumer in some way, 2) its economic viability depends on sufficient demand for the service to justify resources invested in structures that create the potential for reliable production of the service on demand, and 3) the reliable availability of the service at some service point in the SOA framework must be known, and therefore advertised in some recognizable way, to potential consumers of that service in order for them to utilize it. If any of these conditions is not met, we cannot really speak of a service.

A fundamental non-infrastructural design challenge in SOA is the problem of defining useful standardized services and service characteristics, from the point of view of the service consumer, and to foresee and manage changes in the business environment that affect the economic viability of the service framework over time. This is obviously easiest for commoditized services supplied with only minor qualitative differentiation: data storage, and access to standard business applications, for instance, are the success stories of cloud computing.

If, however, qualitative differentiation and change are significant factors, the standardization and semantic definition of services become significant design and maintenance problems, mainly addressed by the semantic service-oriented-architecture (SSOA) concept. This is where we identify the key agility problem.

**Proposition 2.3.1.** Agility in SOA depends on qualitative differentiation in service delivery, in anticipation to a changing environment, and is therefore a problem of understanding and predicting the evolution of service characteristics in an environment.

2.4 Law and services

At first sight, law may appear to be at the wrong end of economic fungibility as a service to be a SOA success story. There are few areas of human economic activity where service appears as variable and ill-defined in its economic value, and differentiated in its forms of delivery, as in the area of law. The law thus presents us with significant conceptual problems, but is at the same time, because it is relevant to all meaningful social interaction between human beings,
of essential importance to SOA. Generally speaking, law may impact viability of, in principle all, services in a number of ways:

1. Demand for a service or willingness to supply the service may depend entirely on its being obligatory or permitted by the law, for either the service consumer or producer;

2. Demand for the service or willingness to supply the service may depend on the (re)allocation of legal responsibility for conformity with the law to the service consumer or service producer; and

3. Demand for the service or willingness to supply the service may depend on the counterpartys reputation for compliance with the law.

**Example** An online tax application submission service loses its value as a service entirely and immediately if paying the associated tax is no longer obligatory.

**Example** The value of access to a database may largely depend on whether it is legally permitted to use that data for marketing purposes.

**Example** The demand for financial statements by accountants may largely depend on their value as recognized legal evidence.

**Example** Legally permitted bypass of copy controls may reduce the economic value of online distribution of copyrighted materials.

**Example** Expert advice based on yesterdays law may be harmful rather than beneficial.

**Example** The value of free disk space in the cloud may significantly depend on personal data protection regulations applicable to those hard disks and the reputation of the company for (non)compliance .

For organizations, modeling the service characteristics of law-based services is a significant problem. On the other hand, a reason for optimism about the maintenance of law-based services, is that law is relatively easy to monitor for changes. Changes to the law originate from few, well-understood sources, are often announced well in advance, and are often communicated in the form of rules. Law is in this sense more well-behaved, and its evolution more predictable, than many sources of change in the business environment, if you invest in your ability to monitor it.

The Agile methodology addresses agility in the area of law as a knowledge management problem of maintaining correspondence between an evolving framework of legal rules and a framework of services. To understand its intent, it is important to understand that it does not aim for just regulatory compliance, as most work on law in BPD does. The Agile methodology aims for a comprehensive understanding of the relationship between law and the viability of service infrastructures.

Legal rules do not function just as constraints on service delivery, as the notion of compliance seems to suggest. As already noted, legal rules may have a
variety of relationships with services that significantly affect the economic value of service arrangements, negatively and positively.

**Example** Sales transactions with customers would be a lot harder if the law did not spell out when a contract exists, when a legal sale has been concluded and a duty to pay has come into existence, did not provide mechanisms to collect payment from uncooperative customers that have a duty to pay, etc. The law binds all sides in a service framework, generally to the advantage of the opposing side in service transactions. Change to the law may therefore create both urgent problems, of recreating regulatory compliance, and urgent opportunities, to utilize new rights efficiently.

The multi-faceted relationship between law and service is most obvious in public service delivery. Public bodies are peculiar organizations. The services they perform are often implementations of public legal acts, performed by public legal personalities, based in formal legislation. Legislation gives administrative organizations public personality, making them exist as legal persons with legal positions towards others, defines what the core functions of public organizations are, and what services they provide. It sometimes even guides how the organization subdivides itself into administrative units, constrains how it organizes business processes inside the organization, and constrains how the functions of the organization are realized by civil servants and computer systems.

Public bodies are also peculiar in often being a monopolist, i.e. the only organization of their category within a jurisdiction; Taxation, immigration, and social services, are often implemented by just one, unique organization with unique legal positions towards the environment.

This unique position creates a peculiar relationship with the legislator; The organizations interests do not directly compete with those of other stakeholders in their category, and its way of implementing the relevant law may easily acquire the status of the only thinkable way of implementing it.

The administrative organization is in a unique position when it comes to providing feedback to the legislator on the effectiveness of existing legal arrangements, and also shares responsibility for that effectiveness, but is at the same time uniquely exposed to the legislators whims.

The Agile methodology has therefore been developed in cooperation with two public administration partners: a tax administration and an immigration service. Both provide services that are clearly in demand, but whose economic value is entirely dependent on a dynamic framework of legal rules. Both are **hard cases** for the commoditization of services, and therefore for SSOA.

The Agile methodology addresses agility as a knowledge management problem of understanding and modeling complex service characteristics and complex service dynamics. Knowledge management comprises a range of practices to identify, create, represent, distribute and enable adoption of insights on, and experiences with, service delivery within an organization. In knowledge management, agility may mean a variety of things, for instance:

1. adaptive work teams on the operational level, that are equipped and empowered to respond directly on a case-by-case basis to unforeseen developments in the environment,
2. project management in development teams that aims to instill awareness of constant organizational change into people, processes, and technology, and

3. constant organization-wide attention for harnessing knowledge, learning, and collaboration to keep track of, and control over, what happens in the environment.

In the Agile methodology it means all of these things, but only when law is concerned. The Agile methodology broadly distinguishes two interesting sources of change in the organizations environment (Fig. 2.1): 1) change of the law, as evidenced by regulation, court decisions, etc., and 2) any other changes in society, in as far as these affect the current interpretation and implementation of law into the service framework that the organization depends on, or should lead to proposed changes to the law in support of the service framework.

Figure 2.1: Besides the law, there are other processes that make the organization move.

Fig. 2.1 makes this relationship visible: as the wheels turn, we can trace both a direction downwards (implementation) and upwards (feedback). Public administrations role is not just to implement the law by giving it hands and feet.
A tax administration or immigration service functions as a pair of eyes of the law too, providing feedback on the effectiveness of law, reinterpreting its meaning in a changing environment, signaling anticipated problems, and proposing changes to the regulatory policy maker.

The distinction between three layers of systems, with the public bodies service infrastructure in the middle, and the coordination of changes between these layers, is a key principle of the Agile methodology.

**Proposition 2.4.1.** In the context of law, the changing environment has two focal points: the law, changed by legislators to reflect policy, and processes in society that require a new interpretation and implementation of the law.

### 2.5 Knowledge level agility

To manage the knowledge an organization develops about 1) the regulatory policy it implements, 2) the resource-bounded rationality of its internal organization, and 3) relevant processes in society that impact the effectiveness of the implementation of regulatory policy, the Agile methodology builds on a philosophy of problem solving. The reasons to make this connection with problem solving are the following:

1. Key to understanding social systems as complex adaptive systems is the choice of basic forms that exhibit self-similarity (i.e. the whole shares forms with the parts): the basic form we choose is that of the embodied intelligent agent, representing persons, groups, and organizations.

2. We subscribe to the principle (from Newell (1982)) that knowledge is that which we ascribe to an intelligent agent to understand its actions.

3. We adopt the view that actions are undertaken to solve some problem. The effective organization of knowledge structures for some domain should therefore follow from an analysis of problems that can be posed and solved in a domain.

4. The noted problem of defining useful standardized services and service characteristics in SSOA can only be effectively addressed by referring to the assumptions and objectives with which intelligent agents approach the production and consumption of services.

5. To model this contextual aspect of services, it is only natural to address it as a problem of symbolic representation of the internal knowledge structures of the agents at both sides of the service interface. These knowledge structures include objectives and assumptions, about the typical behavior of others one has service-based interactions with, based on the law.

6. Only with well-defined problems in mind can one decide, for those specific problem definitions, on resource-bounded optimal strategies for solving
them at design time, using either analytic (game theory, decision theory, etc.) or empirical (statistical, experimental) means.

7. The agility concept is not found in optimal strategies, but in flexibility in adapting problem definitions as we gain knowledge about the challenges the problem solving agent is going to face in its lifetime.

**Proposition 2.5.1.** To understand service characteristics, one needs to understand the problems that the agents involved in the service are solving, and the associated knowledge structures they are using.

Agile problem solving is based on a theory of problem solving presented two decades ago in Breuker (1994), as a component of the CommonKADS methodology for the development of knowledge-based systems. This suite of problem types was based on an analysis of existing problem and task decompositions found in knowledge-based system literature at that time. The generic problem types distinguished in that paper describe recurrent design patterns found in individual and organizational activities, and have been applied to the field of law in the past. The theory captures key insights from embodied cognitive science, and exhibits some affinities with enterprise life cycle models.

In the CommonKADS approach, knowledge analysis consists of three phases: moving from 1) identification of an ill-defined problem to 2) a (well-defined) problem definition, with a problem space and an abstract solution, to 3) an executable task specification for finding solutions in the problem space.

In summary, Breuker (1994) states that 1) the availability of structural and behavioral models in a domain determines which problems can be meaningfully posed and solved, 2) there are recurrent functional/data flow dependencies between recurrent types of problems, in the sense that the solution to one type of problem is a knowledge source for another, 3) tasks typically package chains of functionally dependent problems, and 4) reusable problem solving methods, and associated knowledge representation strategies, match to typical knowledge-intensive tasks in the organization.

The problem solving methods are mainly the province of operations research. The CommonKADS Library relates problem types to known strategies for solving them, although it no longer represents the state of the art in this respect. The task, and its allocation to agents, is a starting point for BPD and SOA design activities. In this project we are concerned with the ability to turn ill-defined problems into well-defined ones, and not with the operational ability to solve them.

**Figure 2 Different problem vocabularies for design and planning**

The suite of problem types presents us with a generic problem solving cycle for intelligent agents, and distinguishes two different vocabularies for describing it, design-based and plan-based, depending on the type of model of the domain that the intelligent agent is working on (Fig. 2.2, 2.3):

**Design:** When the agent aims to be able to establish control in a domain by encapsulating processes into fixed structures, it decides on a design of a
system for dealing with a type of problem, and implements it. The overar-
ching objective: to develop, extend, and protect abilities. The domain of
interest is conceptualized as a system, with system boundaries and com-
ponents, and problem solving consists of 1) modeling a design space, 2)
configuring a system design based on component models, 3) implement-
ing the design in a system, 4) monitoring that system, and 5) diagnosing
faults in the system, feeding back the diagnosis into the modeling problem
(but now as a problem of repairing the system). BPD and SOA design
activities mainly work at this level.

Planning: When we feel design-based solutions are not rational, given the
complex and dynamic characteristics of the domain, or not required, given
the lack of systemic significance of a problem, we make situation-specific
plans to address problems once. The overarching objective: to use the
abilities we have effectively and efficiently. The domain consists of an
intelligent agent situated in an environment, and problem solving consists

Figure 2.2: The plan-oriented problem vocabulary.
of 1) understanding the environment, 2) planning interaction with that environment, using one's abilities, 3) executing the plan, 4) monitoring plan execution, 5) assessing failures, and feeding back the assessment into the problem of understanding the environment (but now as a problem of remediating a specific situation).

Example Many people will, over the course of their lives, sell something; for instance a house. The problem is, for most people, rare enough to approach it as a planning problem only, utilizing means which happen to be available at that time, for instance an auctioning service, or an agent specialized in selling, submitting to the constraints of using a service. The one-shot seller does not aim to establish control over the means of selling for future use. A commercial seller, selling commoditized goods, approaches the same problem as a design problem, setting up a system to support a generalized process of selling. The commercial seller establishes control over the means used in the process. Note that the buying/selling example makes clear that agents compete for control over parts of the process. Legal constraints do not determine whether payment should take place before delivery of a good, or vice versa, for instance. Commercial traders do often add such a constraint in the process, requiring that the customer do his part of the deal first. If the customer would impose the same constraint,
completion of a sale becomes impossible. The customer, who is willing to submit to the constraints of the other party, is therefore the party showing more flexibility in the interaction. The commercial seller, on the other hand, makes his services predictably available.

Example Some boundary cases: a dinner table arrangement has characteristics of a design problem (because time plays no role), but is not intended to be reusable, while a symphony for an orchestra is a plan with complex time constraints that is intended to be reusable.

Example The difference between understanding an environment and diagnosing a system: It is well known that gate logic, based on idealized components performing a single logical operation, can be used to describe a wide variety of electronic, fluidic, pneumatic, and mechanical systems. In design one decides on a gate logic specification and then one implements it, choosing one of the very many ways in which a gate logic specification can be implemented in a physical device. In monitoring and diagnosis of a designed system one compares the actual (electrical, fluidic, pneumatic, etc) behaviour with the previously decided logic specification, to detect and localize faults in the system. One can however use the same set of gate logic components to try to understand the (logical) function of an electronic, fluidic, pneumatic, or mechanical system, assuming that such a logic must be present. This is a very different problem: without a functional model of the system one cannot localize faults in the system. If one has such a working model of the functions of a system, one can however localize faults in one’s model if the actual behavior of the system deviates, assuming that the system is behaving correctly, and that one’s working model is therefore faulty.

**Proposition 2.5.2.** **SOA is a design problem.** We differentiate between the service framework design problem in the development sphere, and the operational planning problems that the operational users of the service framework, as service producer and consumer, are solving in the operational sphere.

The two problem vocabularies represent two alternative perspectives on what is in essence the same generic problem solving cycle that any embodied intelligent agent proceeds through (Fig. 2). A fully articulated model of a domain should in principle allow alternative and interchangeable description in both terminologies, but we tend to focus on one manifestation of problems at the expense of the other, depending on how we classified the systemic significance of the problem. Without this focus, we would be quickly overwhelmed with the complexity of the real world.

Example A development team obviously works with a project plan, a time schedule, and a human resource allocation, but its main deliverable is the far more elaborate design and implementation of some structure. The development team is solving a design problem, and associated planning problems are side issues. In contrast, a court in principle decides on individual cases in accordance with the law, but the argumentation used in a landmark case decision may, over time, acquire the status of a reusable theory about deciding a category of cases. The argumentation structure itself is a residue of strategic planning processes of the parties involved in the court proceedings, originally mainly directed towards an acceptable resolution of one specific case.

### 2.6 Agility as diagnostic ability

Explicit knowledge structures supporting problem solving are often incomplete. Problem solving activities in Fig. 2 may be distributed over coordinated agents, each dealing with a part of the required knowledge structures. Problem solving
activities may be partially or completely delegated to IT systems, to humans following some business process specification specifically designed for that purpose, or to humans exercising personal discretion and tacit knowledge in problem solving, depending on the natural agility of the human mind.

Monitoring and diagnosis tasks, and the remedial action following up on the diagnosis, are commonly delegated to an educated human user, instead of integrating them in the main business process.

The diagnosis process leading to a request for change to a development team often remains largely invisible in the knowledge assets of an organization. This is not necessarily a problem. Incomplete support of problem solving by organizational structures only leads to operational impasses if one fails to foresee the possibility of obstacles calling for diagnosis and remedial action in business process design and fails to equip and empower staff for dealing with obstacles to normal business operations. Not addressing a problem as a business process design problem is a workable solution for rare scenarios, and a BPD will never cover all eventualities.

Remark In our discussions with people with people from development teams about the organizational problem solving model we propose, we have heard complaints that the rationale behind changing legislation and internal guidelines is often not properly shared with development departments before the implementation, even if the reasons for the change are found within their own organizations. That such knowledge exists inside the organization, and led to a proposal for new rules, may be discovered only during user requirements analysis or user evaluation of a technical solution.

Agility does, however, require attention in knowledge management for knowledge structures used in monitoring and diagnosis problems, even if there is no justification for investment into developing BPD execution paths for rare occurrences. As noted, in BPD, agility means intelligently managing degrees of freedom in executing a work flow by offering alternative execution paths to deal with, what would otherwise be, failures. But this is only part of the story. The operational agility of an infrastructure may depend on the potential execution paths, but the operational agility of the organization is rooted in the quick and intelligent use of those alternative execution paths to avoid problems, and the quick adaptation of the infrastructure to newly found problems. Foreseeing and detecting problems is therefore at the core of agility. This is the problem of diagnosis, and diagnostic hypothesis-driven monitoring. The BPD includes active monitors for signs of failure, and connects these signs to a diagnostic hypothesis and a prepared remedy or repair.

Proposition 2.6.1. The ability of agile response functionally depends, in both the operational and the development sphere, on the ability to solve monitoring and diagnosis problems in those spheres, using appropriate structural and behavioural models of the system in which those problems arise.

We therefore devoted special attention to the knowledge structures required for monitoring and diagnosis, and the knowledge resources that are actually available in an organization for monitoring and diagnosis.
Whether we are considering relevant case law, or stories about critical incidents in the organization, as they are shared in a user requirements meeting, we analyze failure cases as

**narratives** involving agents and their interactions, as

**prototypical diagnostic agent roles** describing the rationale of certain actions (for instance fraud) from the perspective of the acting agent, and as

**argumentation frameworks** connecting evidence to arguments, and counter-arguments, about the actions and intentions of agents, and to appropriate remedies and repairs.

While argumentation frameworks are well known as a technique for the analysis of case law, and narrative frameworks have been used by others, we have added a number of innovations in the Agile project by situating the events and causal connections in a narrative in an underlying system, a multi-agent system, and addressing the function of case law and critical incident narratives as knowledge resources in a model-based diagnosis problem. This is an innovation in Artificial Intelligence & Law, and, even more so, in SOA.

This model-based diagnosis problem is, in the current approach to managing projects, often captured in the rationale underlying performance measures, and in use case scenarios exploring reactions to foreseen problems and opportunities. We believe that the model-based diagnosis concept is a more appropriate abstraction for this purpose, resulting, if one takes the right abstract components of the system, in knowledge resources that remain useful much longer than existing alternatives.

**Proposition 2.6.2.** The model-based diagnosis abstraction remains valid over time and context, regardless of changing business processes, performance measures, paradigms, modeling technologies, or the legal-institutional framework.

Diagnosis problems can be attacked with a wide variety of problem solving techniques, ranging from logic-based systematic hypothesis elimination methods like those of Reiter (1987) and de Kleer et al. (1992), logic-based frameworks for arguments and evidence like the work of Prakken (2004) and many others in computer science & law, classification approaches like the heuristic classification of Clancey (1985), to statistics-based approaches like Bayesian networks (cf. for instance Madsen (2010)). The risk management field is closely related.

The quality of a diagnostic process is determined less by the method of reasoning used than by the quality of the information used, and the grip one has on the diagnostic hypothesis space, i.e how well one understands the problem domain.

**Example** Many assume that the statistical methods of empirical science are superior over qualitative, abductive reasoning methods. The abductive method of Reiter (1987) will however work more reliably for analysis of electronic circuits based on gate logic specifications than bayesian networks based on statistical data will work for fraud detection, for the simple reason that we understand the system being diagnosed much better, and have better access to information about the behaviour of the system.
The *knowledge level* aspect of agility is the ability to monitor and diagnose. To develop that ability, the organization should have processes in place to create and maintain the knowledge structures needed for solving monitoring and diagnosis problems. In this document we describe a reference architecture for a knowledge acquisition infrastructure. This is a first step: prompt and accurate diagnoses need to be translated into prompt and accurate *alternative execution paths* as well. This is the take on the agility problem developed in Gong (2012), accompanying this methodological document.
Chapter 3

Overview of the methodology

3.1 Introduction: the real, actual, and empirical

To explain our problem solving framework, it helps to start with the way we ontologically structure reality, following a philosophy of science called critical realism. What we think of reality is stratified into the levels of the empirical, the actual, and the real (see figure 3.1). At the level of the real lie generative mechanisms defined as the powers and liabilities of things. In a social system we may take these to be the abilities and susceptibilities of agents towards each other. These mechanisms are the causes of events, some of which are experienced by us, as agents. Mechanisms exist even when not acting, and may act in their normal way even when not actualized, due to interference by other mechanisms, as a regular sequence of events, which may or may not be empirically experienced by humans.

Example Place yourself in the position of a decision maker who judges the eligibility of someone for some benefit. One of the conditions for that eligibility is having Dutch nationality. This is the real thing under study. In order to prove Dutch nationality someone handed over a Dutch passport in an intake meeting. The decision maker was not present. This is the event level, the actualization of Dutch nationality in this context: he could hand over a Dutch passport because he is a Dutch national. What the decision maker is looking at is a scan made of that passport, attached to the file he is deciding on. This is, in this context, the empirical level, the observation made by the decision maker. The decision maker takes the scanned passport as evidence of the event – someone, with looks matching the photograph in the passport, handed over that passport to a colleague – and the event is evidence of Dutch nationality because normally Dutch nationality is the mechanism that explains the ability of handing over a passport. Making this distinction is key to understanding both how legal-administrative services may fail, and how legislators regulate the decision making process. The legislator moves between these levels when prescribing how services should be delivered, requiring Dutch nationality, and/or production of a passport, and/or the keeping of a scan of a passport in the records.
There are two functions that scientists carry out when conducting experiments. First, the scientist triggers the mechanism under study to ensure that it is active, causing observable events, and second, the scientists must prevent interference with the operation, and observation, of the mechanism. Only in the limiting case of a carefully constructed, closed scientific experiment, the level of the real becomes directly aligned with the level of events, and with human experience.

In the Agile methodology the reconstruction of the real (social structures, and agent behaviour) from the empirical (the messages observed by a problem solving agent), and the design/planning problem of keeping grip on the real, take the center stage.

Figure 3.2 translates these three levels of reality as experienced by us into the three major forms of knowledge of the world that an agent has of social reality: the ability to

1. create models of agent behaviour & social structure,
2. predict from those models messages that will (not) be sent between agents, and
3. monitor observable messages to compare them to the ones predicted.

In figure 2.3 in chapter ?? shows a functional dependency chain of problem types that a generic intelligent agent should be able to solve. Figure 3.3 details the inputs of the different problem types; The output of one problem type is the input of the next one.
Figure 3.2: knowledge level representations of what we see and what is really there.

Compared to the cyclical figures in figure 2.3 it 1) misses the implementation step, and 2) does not cycle back to the modeling problem. There are two complications in displaying the process of problem solving that the reader should be aware of:

1. While problem solving is reasoning, that is mental action with mental effects only, the problems find their origin in real action, and unintended or unexpected real effects, and solutions eventually get translated into real action. Implementation is not reasoning, but the translation of an abstract solution into real world action. Implementation/execution is an odd duck in the pond of figure 2.3.

2. If the arrows between problem types denote functional dependencies between problem types, the arrow from diagnosis to modeling denotes something substantially different. While every problem type rests on all knowledge structures built up in the problem solving activities upstream, entering a new modeling phase means that the previous iteration of knowledge structures was judged faulty and will be adapted.

While diagnosis is central to organizational agility, the step from a diagnostic solution to a new model of agent behaviour and social structure is the ill-defined problem that defies decision making of business process design support.

In the next few sections we address important law-related components of the modeling problem:
3.2 How agents use the law

For the readers of this document, interested in knowledge management intelligent systems, we can think of legal knowledge as the knowledge we ascribe to an agent that acts with skill in the sphere of law, following the definition of knowledge in Newell (1982).

A legally proficient agent has working knowledge of this sphere, and of its interfaces with its environment, and uses that knowledge effectively to attain
his goals. The legally proficient agent is a norm subject, a normative agent that follows rules, but he is a stakeholder, and perhaps an investor, in the legal system too.

Large organizations often have strategies as investors and stakeholders in the law. In this chapter we consider what this means for legal knowledge engineering, exploring a concept of legal knowledge that may guide design in organizations.

Because knowledge guides action, legal knowledge is not limited to the sphere of law. It addresses all uses of the law. We distinguish the following four loosely connected legal knowledge representation domains, and try to distinguish them conceptually:

1. The organization of the sources of law;
2. The organization of abstract legal institutions;
3. Implementation and production of law in social structures; and
4. The application of law in individual decision making and the judgment of individual cases.

The first two domains, and in particular the second, are characteristic of a legal positivist perspective on law, and its boundaries. What is beyond them, is not strictly in the sphere of law, but still a matter of practical legal knowledge.

The first knowledge domain addresses the texts that are the sources of law. Society uses a wide variety of sources of law, and uses many labels other than legislation to describe sets of written rules: quasi-legislation; administrative rules; codes of practice; guidance; guidance notes; policy guidance; guidelines; circulars; framework documents; outline schemes, and statements of advice. A commonality of these documents is that they are understood to postulate or describe norms that guide or mandate conduct in a given type of situation. This guidance can pertain both to physical behaviour and to decision making processes.

Knowledge of the structural organization of the texts, and the structural organization of the corpus of texts that make up the sources of law is legal knowledge. This first knowledge domain deals with references between texts, discourse context, reuse of terminology, the use of standardized sentences to express legal-institutional design patterns, and the intentional use of legal principles like lex superior derogat legi inferiori (lex superior; higher law derogates lower law) and lex specialis derogat legi generali (lex specialis; specific law derogates general law) in the organization of the corpus of legislation.

The second knowledge domain addresses the abstract components of the legal system; its institutional structures, and rules, as posited in the sources of law. This domain is the core subject of positivist legal theory\(^1\).

\(^1\)The key knowledge management problem from this perspective is to uncover and document the structure of the legal system as a whole, based on the principle that there should be one legal system, it can be discovered in the sources of law, and therefore there should be one
A vast literature exists on abstract institutional and deontic design patterns derived from legal positivist theory for use in the development of intelligent systems. Automated normative reasoning with legal rules, in two-party interactions at least, may be considered a solved problem for engineering purposes. Many formal models that are good enough to guide normative legal reasoning exist. These do not, however, enjoy great success in business modeling, because they are hard to learn, computationally unattractive, and do not address the most important problems for organizations.

The third knowledge domain is characteristic of the legal realist perspective, with its focus on the sociological aspects of law. It covers the pragmatics of enforcement, legal service delivery, and judicial decision making. It covers the political arena, and policy making processes. This area receives less interest because it is at the edges of the sphere of law, but it is of central importance to large organizations. The agile methodology focuses on this knowledge domain.

The fourth knowledge domain is clearly the one most legal professionals act in most of the time: interpretation and application of the law in context, from a specific perspective, embedded in specific social structures, given certain knowledge, expectations, motives, a concrete problem, and concrete plans. It represents the individuals perspective on the real influence of the law on his actions, besides all those other things that guide action. This is what we typically refer to as legal expert knowledge, which is the kind of knowledge modeled in decision support systems.

Chapter 4 deals with the relation between the formalized rules, as we find them in the sources of law (in natural language) or business rules (in a knowledge representation language), and behaviour.

3.2.1 Normative agent roles

The second knowledge domain above is the most obvious starting point for the normative aspect of legal knowledge. Much of the terminology to describe institutions has been strongly influenced by the work of Searle (cf. generally Searle (1970); Searle and Vanderveken (1985a)). Searle describes the distinction between two types of rule, being regulative and constitutive rules.

Both the regulative and the constitutive rule are normative, and should be distinguished from other rules that merely generalize past observations in the hope that they generalize to future ones. The regulative and the constitutive rule affect events that happen after the rule comes into existence, and application of the rules by an agent is the force that makes the consequent of the rule happen. The regulative rule guides behaviour, while the constitutive rule guides interpretation of events in the terms of an institutional reality. The regulative rule can be violated.

We moreover classify some rules as auxiliary meaning that they are constitutive or regulative of the legal system itself without aiming to regulate some external social system.

right answer to any definite legal question about the existence of a legal rule or other legal entity.
The regulative function of law is generally taken as the defining one of the legal system in positivist legal theory. This is for instance apparent in Austin's adagium (cf. Austin (1954)) – later developed by Hart (cf. Hart (1961)) – of law as the command of the sovereign backed by the threat of punishment. Regulative rules embody the primary purpose of law, which is to define a function from behaviour to the qualifications \{allowed, disallowed\}. Constitutive rules merely have an assisting function, making the extent of the regulative rules clear by spelling out how to recognize and classify behaviour. Regulative rules are primary rules, and constitutive rules are secondary.

Regulative rules typically occur in three main variants in natural language: obligations, permissions, and prohibitions. Deontic logic dictates that something that is obliged is permitted, and not prohibited. Natural languages have special auxiliary verbs for them (may, must, ought, etc). The use of specific auxiliary verbs is however not indicative of whether some proposition is normative or not in law.

**Example** Some random examples of regulative rules, in a variety of forms: 1) Drivers keep to the right of the road. *(Dutch traffic rule)* 2) Once moving you should keep to the left, unless road signs or markings indicate otherwise. *(British variation)* 3) The taking of a good that belongs to another with the intent to disappropriate it, is theft, and is punishable with a prison sentence of at most one year. 4) The seller has a duty to deliver, and a right to payment.

In Fig. 3.4 the influential categorization of legal positions by Hohfeld is shown (but in a new tabular format that suits this methodology). This categorization of legal positions can be interpreted in terms of affordances. On the left hand side we find the positive categories, which may informally all be called rights. Having these may be said to create abilities to achieve certain effects one would not otherwise be able to achieve. On the right hand side we find their correlatives: These may be said to constrain freedom of action.

On top we find the primary, regulative, normative or duty categories, termed practical by us. These are conferred by regulative rules. They are practical because they directly concern reasoning from reasons towards intentions (to act or not act). If the patient is under a duty to do something under certain circumstances, then the patient has reason to adopt an intention to do something under those circumstances, and the agent 1) has good reason to expect that the agent has that intention to do something in the interest of the agent, 2) has

<table>
<thead>
<tr>
<th>Hohfeld</th>
<th>Agent</th>
<th>Patient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ability</td>
<td>Negation</td>
</tr>
<tr>
<td>Practical</td>
<td>Claim</td>
<td>No claim</td>
</tr>
<tr>
<td></td>
<td>Liberty</td>
<td>Duty</td>
</tr>
<tr>
<td>Epistemic</td>
<td>Power</td>
<td>Disability</td>
</tr>
<tr>
<td></td>
<td>Immunity</td>
<td>Liability</td>
</tr>
</tbody>
</table>

Figure 3.4: Hohfeld’s categorization of legal positions, or affordances of the legal system, from the agent and patient perspective.
reason to see to it that the patient does it, and 3) has the ability, created elsewhere, to seek punishment or damages if the patient does not. The liberty on the other hand, simply protects existing ability of the agent from imposition of a duty to the contrary. The patient of the liberty has no reason to expect anything. If one classifies a rule as a duty, one should be able to point at the agent’s ability to seek punishment of the patient.

**Example** The buyer has a duty to pay the seller is an example, where the seller is the sole, and explicitly named, beneficiary who has the correlated claim right. The seller himself needs to take action to see that buyers that do not pay are punished. Drivers yield to traffic from the right is another example of a duty; In this case all other traffic participants have a claim right. An ambulance running a siren is exempted from this rule, and other traffic participants need to be aware of the ambulance’s liberty in traffic. Punishment in traffic takes variable forms. Police officers may book violators, and violators will be held responsible in case of accidents.

In the bottom we find the secondary, constitutive, or power categories, termed *epistemic* by us because they concern reasoning from reasons towards conclusions. These are conferred by constitutive rules. The agent has a power, if the patient is liable to come to a certain conclusion in certain circumstances, and the patient may expect the patient to come to that conclusion. The patient may however have a specific immunity to the power. Powers are secondary in the sense of not being directly relatable to punishment. They take their force solely from their relationship with rights.

**Example** Ownership of a good is a typical example of a power, in that it creates liberties towards that good, and duties of non-owners towards that good. A driver’s license gives one the liberty to drive.

Although the legislator may phrase rules in these positional terms, it is important to realize that the list categorizes affordances of rules rather than rules themselves, and that these affordances have to be checked one by one per rule. The affordances may moreover relate to the initiation, continuation, or termination of the regulated behaviour.

**Example** If driver should keep to the right hand side of the road, then drivers have a duty to stay right and a duty to go to the right if they are on the left, and a right to expect the same from other drivers.

Figure 3.5 categorizes these in terms of four types of motive towards some behaviour description:

**Keep** means not to terminate a described obliged behaviour, by initiating a prohibited alternative one;

<table>
<thead>
<tr>
<th>Specification of duty</th>
<th>Obligation specified</th>
<th>Prohibition specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation specified</td>
<td>Keep</td>
<td>Cure</td>
</tr>
<tr>
<td>Termination specified</td>
<td>Acquire</td>
<td>Prevent</td>
</tr>
</tbody>
</table>

Figure 3.5: How duties are expressed in rules vs. types of motives.
Cure means to terminate a described prohibited behaviour, by initiating an obliged alternative;

Acquire means to terminate an prohibited alternative behaviour, by initiating a described obliged one; and

Prevent means not to terminate an alternative obliged behaviour, by initiating a described prohibited one.

Logically speaking at least some of the motives are redundant, since they simply define opposites and can be defined in terms of negation. In practice the four different motives are implemented differently in practice, and therefore it is recommended to categorize uses of rules in terms of motive.

3.2.2 Exercising legal power

Three basic mechanisms exist that explain how law and power get aligned.

1. Sometimes the means arise more or less automatically from an alignment of interests. One could for instance say that the driver of a car has practical, and pretty reliable, power to make other drivers coming from the left hand side yield in traffic. This practical power arises not from the effectiveness of enforcement of a norm that one ought to yield for traffic from the right, but from the mere expectation that traffic from the right will not usually yield, and an interest in avoiding collisions. The norm has a mere coordinating function, and drivers have a motive to keep to the right.

2. Not all norms are equally inspirational for spontaneous coordination of behaviour. Additional power must be generated. A second way to effectuate law is to rely on the interest of participants in norm violations in law enforcement. This mechanism is most obviously at work in private law, where a victim of norm violation is entitled to go to court, and usually has an interest in doing so, to have the case adjudicated by professional judges. In this case power is generated by private persons and the resources they are willing to invest in enforcement of the law, on an opportunistic basis. Victims must give claim rights practical effect at least sometimes for this kind of norm to work.

3. The third and most comprehensive link between the two spheres is in modern society clearly the state. State institutions, including a state-subsidized judiciary, act as reliable and omnipresent transmission belts between the sphere of law and the sphere of power. The state gives the law hands and feet, interprets it, and changes it, using resources appropriated for that purpose. Effectiveness becomes a policy-level motive.

A second way of looking at power in relation to the law, is to look at the types of power one can hold over the legal system. A classical categorization is the one in table 3.6, by De Mulder (1998), the Tetras Politica, an extension of
Montesquieu’s Trias Politica. The new addition, monitoring power, reflects a power to decide what events get attention and which do not. It is the power to determine what should be observable.

The powers relate nicely to the distinction between the practical and epistemic, already introduced in the context of rules and affordances (see figure 3.7). The power to legislate and the power to judge relate only to meaning. It is the power to determine what does and what does not logically follow from certain behaviour, and to have others adopt those opinions on meaning. The power to decide and observe are however practical, real world exercises of power, determining the actual effectiveness of law.

### 3.2.3 Information, evidence, and argumentation

Figure 3.7 may however be restated to a theory of problem solving, reflecting four types of knowledge, used in four types of problem solving ability (see figure 3.8). Restated this way, the four types of ability directly refer back to the four problems in Fig. 3.3. Modeling is an epistemic design problem, in which one designs a model of the world to understand it, to understand one’s abilities and susceptibilities. Design is a practical design problem, in which one designs specifications to be implemented, based on one’s motives, abilities and susceptibilities. Monitoring is a practical diagnostic problem, in which one decides what information to collect to find differences between expectations and observations. Diagnosis, or judgment, is the epistemic diagnostic problem, in which one comes to a fault localization in the model or the design specification/imputation, based on one’s knowledge of the range of possible behaviours and one’s diagnostic motives.

A rationalization of diagnostic reasoning is given in chapter 7, relating the problem of observation, and argumentation schemes used for enforcement, to model-based diagnosis problem solving.
3.2.4 Operations, development, and policy making

The four knowledge domains are leading in the stratification of law-related problem solving activities we have chosen. Fig. 3.9 shows problem-solving cycles in three spheres of law-based action in public administration: the legal-political policy making sphere, representing the first two knowledge domains, the design-oriented development sphere, and the work floor operations sphere. The law itself may address all spheres, regulating not only operations, but also organizational infrastructure, and the process of regulation itself.

Fig. 3.9 pictures the state’s transmission belt between law and power. In the lower, or deeper, two spheres the activities of the state organization are systematized. The organization is subjected to generally binding rules, which are explained in documentation, constrained by business processes, allocated limited resources, described in database management systems, etc. These are development activities, to be approached with a design problem solving model. At the top level primary government services are delivered, decisions on individual cases are taken, and individual cases are singled out for enforcement.

Operational activities are best conceived in a planning problem solving model. Problems regularly arise on the top level, and a diagnosis gives them a meaning, or possible meanings, on deeper levels. A problematic case on the operational level may mean that more resources should be allocated, or that documentation should be changed, or that the database tables should be redesigned, et cetera, or, finally, that the law should be changed to overcome friction with hands-on reality.

In private business, we usually speak of compliance of products and processes with the rules in the development sphere, and of action conform the rules in the operations sphere. Law is perceived in the first place as a constraint on designs and operations in private business. In the state’s organizations, law has a broader significance: it enforces, interprets, and it delivers the services implied by the law. The law makes things possible, determines the identity of the organization by stating its mission, etc.

From the perspective of the state organization, we are dealing with two mutually dependent design activities. On one side, the organization designs itself, using the rules as a source of requirements and constraints. On the other side, its limits, as perceived by itself, are a source of requirements and constraints in the design of the rules it proposes. As a stakeholder in policy making, it represents the sunk costs of the state in the implementation of the law in society. This does not necessarily mean that the organization narrowly represents its own interests in policy making, but rather that it represents its
own interpretation of the frictions that it observes, internally, in its network, and among its clients.

To react swiftly to problems, the organization needs to know the extent of its freedom to interpret and implement within the limits of the rules. It, in other words, needs to have traction on:

1. the alternative interpretations of the law, and of its effects, known to, and used in, the organization,

2. the organization’s formal account of the current implementation of the law in the organization and the service network around it,

3. its perspective on alternative possible implementations within the design space constrained by the current law in the organization and the service network around it, and

4. how proposed rules affect the potential performance envelope of the organization.

This distinction between the legal system per se, and its contingent alignment with a system of state exercise of power, as represented by its business process specifications and decision support system knowledge bases, can be reflected in knowledge representation.

The organization principle for this classification is functional dependency between knowledge roles, inspired by the typology of problems and views on problem solving first presented two decades ago in Breuker (1994). A complete library of problem-solving methods with associated knowledge assets, organized by generic problem types addressed by them, was developed, described in Breuker and de Velde (1994).

When compared to the actual activities of large organizations dealing with the law, the least developed tasks, from a knowledge management perspective, are usually the diagnostic tasks. Even on the operational level, there is no significant investment in ICT support of handling the problem cases, which are outside the happy flow. On the development level, legal diagnostic feedback from the operational level is even worse. People in development departments may even feel that they are out-of-the-loop completely, as legal professionals working in the operational sphere directly pass critical incident stories to legal professionals on the policy making level. In extreme cases, developers may read about it first in the newspaper. This is therefore the area calling for improved knowledge management.

3.2.5 Context and knowledge engineering

To help understanding our ontological stratification, consider what is being produced in each of the three spheres (see table 3.1). It is natural to consider the social structures first: the law gives us rules and institutions, the implementation by the state processes and products, which are operationally used to deliver services, and to perform public legal acts. Each of these is presented/represented
in information structures. The sources of law represent the legal system. Specifications, manuals, websites, program code, knowledge bases, communication protocols, standardized forms, and database schemas represent implementation. Formal decisions and judgments, database entries, filled in forms, and propositions in knowledge bases represent operations. As information scientists and knowledge engineers, we are on the interface between these products. We study the social objects indirectly, by studying the information about them, and propose new ways of structuring the information that represents the social objects.

All information structures that are produced belong to spheres of action. Knowledge representations, whether they are direct representations of the sources of law, decision support system knowledge bases, standardized decision argument structures, policy field simulations, jurisprudential theories of some domain, or organization-wide legal ontologies, are contextualized to their contexts of use. This is even true of representations of legal knowledge in the judiciary. In the judiciary, the development sphere is underrepresented in the organization but development for judicial decision making as a problem domain still exists. Work in jurisprudence on theory construction from case law fills in this gap, and this work, insofar as it is, or can be, formalized, mainly takes the form of reusable evidentiary argumentation schemes. Because the judiciary has only a rudimentary division of labour, insights in the nature of the work flow hardly translate into specialized business activities.

An example: assume a rule stating that a decision, which is favourable or unfavourable to a client, should be taken within two weeks. On the development level, this means firstly, that the decision process should designed to be fast, and secondly, that an operational resolution should be made available if it is not met, for instance by deciding favourably when the deadline is hit. This becomes the operational interpretation of the rule. On the development level, the percentage hitting the deadline is a diagnostic measure, and a simple repair is the allocation of more resources. On the policy level, we consider the dynamics as a policy theory: we may note the rule wastes organizational resources, and clients intentionally delay the decision making procedure to get their way, rendering the entire decision process largely ineffective, and therefore the deadline should be removed, or opportunities to waste time should be limited, etc. The policy effects that we believe we observe depend on the implementation.

3.2.6 Summary of principles

1. For each legal rule, one should be able to specify its use in table 3.2, for each relevant agent role, by stating motives in the form: prevent, acquire, cure, keep.
Table 3.1: Social and information structures

<table>
<thead>
<tr>
<th></th>
<th>policy making</th>
<th>development</th>
<th>operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>social information rules</td>
<td>processes, products specifications, manuals services, acts decisions, judgments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sources of law</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Fill in for each legal rule

<table>
<thead>
<tr>
<th></th>
<th>requirement</th>
<th>specification</th>
<th>observable</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>operations</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>development</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>policy making</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

2. Alternatively one may fill in information assets (rules, database records, etc.) related to the cell.

Chapter 6 specifies modeling techniques for describing motives.

### 3.3 Maintaining the sources of law

Chapter 5 details the requirements for managing a corpus of legislation. Some of the innovations we introduced in the project relate to the management of sources of law.

The source of law is a writing that can be, is, was, or presumably will be used to back an argument concerning the existence of a certain institutional entity in a certain legal institution (cf. Boer et al. (2006)): 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.

The sources of law function as a log book of relevant legislative changes to a legal institution. A well-picked body of sources of law may also be considered a blueprint or a snapshot of the rules and structures of a specific legal institution of interest in time, but this notion should not be taken for granted.

The legislative act belongs to a broader category of formal legal acts that are characterized by 1) the requirement that one intends to bring about a certain institutional change, and 2) that this intent is communicated in writing, i.e. the institutional change is represented.

The purpose of a logical representation is however to represent the state of the corpus at a specific interval in time. There are three kinds of legal rules, classified by purpose. The regulative rule corresponds with our practical requirements. The constitutive rule corresponds with our epistemic requirements. The auxiliary rule is an epistemic requirement on the management of the corpus of
legislation. These rules determine applicability in time, etc, and are interpreted relative to the context of production of the corpus.

3.3.1 MetaLex and Bibliographic Identity

To implement traceability from knowledge representation to sources of law, the Agile project builds on the results of our work on MetaLex XML (cf. for instance Boer et al. (2008b, 2006); Boer (2009a)), an XML metastandard for legal and legislative resources. MetaLex is a common document format, processing model, metadata set, and ontology for software development, standardized by a CEN/ISSS\(^2\) committee specification in 2006 and 2010.

MetaLex requires adherence to a URI\(^3\) based, open, persistent, globally unique, memorizable, meaningful, and “guessable” naming convention for legislative resources based on provenance information. This provenance information can be extracted in RDF form and used in OWL2 Boer (2009b).

MetaLex is especially useful for our purposes because it standardizes legal bibliographic identity. The determination of bibliographic identity of sources of law is essential for deciding on the applicability in time of legal rules presented in those sources of law.

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, as recommended by the Functional Requirements for Bibliographic Records (FRBR; cf. Saur (1998)).

MetaLex extends the FRBR with a detailed but jurisdiction-independent model of the lifecycle of sources of law, that models the source of law as a succession of consolidated versions, and optionally \(\textit{ex tunc}\) consolidations to capture the possibility of retroactive correction (errata corrige) or annulment after the fact of modifications to a legislative text by a constitutional court. In these cases the version timeline is changed retroactively: the conceptual time travel involved is an excellent example of the weird applications of constitutiveness. See for instance Governatori and Rotolo (2009) for an explanation of the practical ramifications of annulment, and more generally an overview of the complexities involved in change of the law. Note that while MetaLex permits the identification of versions in different timelines, the involved reasoning requires defeasibility.

The use of MetaLex identification and referencing solves one aspect of the traceability problem. In current organizational practice links are more often than not made to locatable items, often without formal agreements about the permanence of the used item identifiers \(\textit{even}\) between different departments of the same organization. Correct traceability to the right bibliographic abstraction (generally work or expression depending on the purpose of the reference) is – particularly at the levels below \textit{formal} law – a notable weak point in organizational practice, and \(\textit{ex tunc}\) change scenarios are not explicitly modeled.

\(^2\)http://www.cen.eu
\(^3\)Uniform resource identifier.
or even recognized. MetaLex makes this aspect of the traceability problem at least explicit, and provides some tools to address it.

In the MetaLex metadata set, specified in an OWL ontology, the realizes property between expressions and works represents the connection between the two ontological levels at which documents exist that are of relevance to their real world use. The source of law on the expression level for instance cites other rules on the work level, while the legal rules we represent knowledge about are necessarily identified by their representation in a discrete number of expressions Boer (2009a).

A citation (text fragment) w applies to (concept) C should for instance be read as each legal rule that is represented by an expression-level text fragment that realizes work fragment w applies to C. This representation technique plays an important role in the Agile project, and is observed to significantly cut down on rather pointless maintenance operations redirecting reference pointers. The idea of the MetaLex standard is of course that provenance metadata will be supplied by the publisher of the used XML manifestation, and is extracted from it in RDF form by organizations that use it.

3.3.2 Identity of Legal Concepts and Rules over Time

The MetaLex source of law at the expression level refers to a set of terms, and it represents a set of legal rules and other legal assertions. One might think of the set of terms and legal rules occurring in all known expressions of a work as the shared set of terms and rules at the work level. The shared work level set however only exists from a specific vantage point in time, or only once the source of law has become immutable after its repeal, and the shared work level set – which can no longer change – has become largely irrelevant. This abstraction cannot be safely used, since it is likely to lead to versioning problems.

In chapter 5 the subject of aligning version expressions of the knowledge representation with expressions of the sources of law, and the identity of legal concepts over time, is discussed in detail.

3.3.3 Actions performed on Documents

An important design feature of MetaLex from our perspective, is that provenance information is organized around actions performed on documents.

Provenance metadata more often than not exists of simple predicate-object statements about electronic documents, even though permitting different perspectives on the same action, because its identity was not made explicit, may yield incompatible metadata descriptions. This results in unnecessary duplication of metadata, and separate occasions in which to make mistakes. It therefore creates unnecessary maintenance, and, lastly, the loss of relevant references between documents Boer (2007).

An action generally plays the mediating role between relevant entities and the resource the metadata description is about. The natural coherence between
for instance author, publication date, and publication channel information (e.g. state gazette bibliographic information) is apparent to all: all are participants in the publication (promulgation) event. There is also a natural coherence between an old consolidation, the new consolidation, the modifying legislation, the modifying authority, and the modification date: the modification event links them together.

Because actions also play a central role on the other layers, we have chosen for a uniform representation of action inspired by MetaLex. Generally, we try to build on analogies between the legislative domain and the implementation domain, choosing the same representation solutions for both.

3.3.4 Applicability

Applicability plays a central role for knowledge engineers as soon as the reified legal rule and the interpretation of its logical meaning are distinguished. The logical rule must assert explicitly that the legal rule is being applied. The law also frequently identifies rules: a special class of legal rules, applicability rules (e.g. Prakken and Sartor (1995); Kowalski and Toni (1996); Boer (2009a), constrains the applicability of other rules, or make the application of one legal rule conditional on the application of another legal rule.

Applicability is wherever possible attached to legal actions as a methodological choice. Since we do not determine confluence of applicability rules from subsumption between propositions, or sets of them, we have to be sure that rules about the same subject apply to the same thing. Actions are the focal objects.

One of the great challenges in understanding applicability of legal rules is the distinction between its dispositional and categorical meaning (i.e. if the rule were applied to something, the result would have some quality that it wouldn’t have if it were inapplicable, vs. the rule has been as a matter of fact applied or not). In its epistemological applications (of which Kowalski and Toni (1996) is an excellent example) in defeasible reasoning, and in implementation resources, it is generally taken to be dispositional. Why make explicit the application of a rule at all, unless it is defeasible?

While the effect of switching rules on and off for specific cases or in specific task contexts may be realized by an extra condition to the rule styled as a form of dispositional applicability statement about the legal rule, this is an unnecessary epistemological commitment not explicitly warranted by the law itself. Such dispositional use of concepts in essence tries to capture it is consistent to assume that, which refers to reasoning strategy rather than to the meaning of the legal rule per se.

We also do not attempt to account directly for metalegal principles like lex specialis (more specific rules defeat general rules), and lex posterior (newer rules defeat older rules). Following Boer (2009a) we are of the opinion that lex specialis and lex posterior are based on generic principles of practical communication and cognitive function that give rise to temporal and logical defeasibility,
and do not as such have to be specifically accounted for in legal knowledge representation.

3.3.5 A demonstrator for MetaLex

In Hoekstra (2011); Winkels et al. (2013) we report on a demonstrator for the basiswettenbestand corpus of legislation of the Netherlands that 1) makes all metadata available, including versioning events, and 2) provides a simple user interface for accessing that corpus.

3.3.6 Summary of principles

1. There are two main sources of law: legislation and exposition of the meaning of law by way of narratives and conclusions (as in case law).

2. The body of legislation should be managed in accordance with MetaLex principles. MetaLex metadata is metadata that usually can not be found in the text itself: It represents what we need to know about the production of the text.

3. The main body of the text changes existing institutional structures.

4. In addition, auxiliary rules in the text apply to the text itself. Chapter 5 discusses some in detail.

5. the first representation task is to keep a correspondence between the text and a model of the institution. Implied ontological commitments (i.e. the existence of certain legal entities) in the text postulate the existence of institutional concepts, in a certain time interval.

6. Integrate and differentiate concepts with the same name found in the corpus, keeping an eye open for concepts changing into another concept, and the consequences of that change for existing data.

7. Isomorphism between text and logical representation is desirable, although such a logical representation cannot be useful for guiding behaviour.

8. Rules that guide action integrate legal rules with regulated social systems. Isomorphism is not required.

3.4 Maintaining models of regulated social systems

Although few would argue that large organizations are designed, with agents as the components, little has been done to address complex social systems as a model-based diagnosis problem. Localization of fault is however an important part of a stereotypical legal problem: we look for someone responsible for norm
violation. Enforcement involves a search for a reasonable responsibility assignment to one or more agents, and subsequent punishment of, or remedial action by, those agents. Problems that threaten public administration also involve localization of agents responsible for performance problems, and design-based remedies, like resource reallocation, redesign of software, revision of guidelines, etc, can be characterized as component repairs.

It is not in principle a good thing to build advanced planning or reason-ability ability into agents, as interesting as this may be from an AI research perspective. One of the reasons to keep agents relatively simple is the possibility of diagnosis. To this end, we introduce the model-based diagnosis problem of a diagnostic agent having a diagnostic model of an observable multi-agent system in its environment. We introduce the agent role as component of the multi-agent system, and agent role descriptions as models of normal, functional components and abnormal, faulty components.

Modeling techniques for agent behaviours are worked out in chapter 6.

### 3.4.1 Summary of principles

1. A social system is characterized by the agent role instances that participate in it as components.

2. The social system description and the social system diagnostic hypothesis can be characterized as MAS configurations from an objective point of view.

3. Externalized, “objective” description of the social system is undesirable, because agents are the participating components, and system breakdown often results from different subjective social system descriptions. The existence of social structures must be accepted by the participants in the structure, and should be treated as subjective. Cognitively speaking, a more plausible description of the diagnosis question is that the diagnostic agent(s) sort of participate in the social system being diagnosed, although a diagnostic agent has no fault models for itself.

4. For purposes of social system simulation, the MAS agent is equated with the agent role concept. Agent roles are instantiated as MAS agents; each role has its own agent thread and working memory, even if the roles are assigned to the same agent player.

5. Diagnosis of MAS configurations is hard because agent interaction is complex and topology flexible. Fault model-based reasoning is the only viable approach to automated diagnosis, and MAS configurations should be kept relatively simple.

6. The agent role description is characterized by (a set of beliefs shared by some group about) the events an agent can experience, its abilities to
create events, its beliefs, its goals, and its (goal-condition-actions and event-condition-actions) plans.

(a) The agent role is akin to a perspective, in the perspectivist sense. The perspective concept takes an internal stance towards the agent role.

(b) Complex combinations of agent roles with internal coordination mechanisms may be considered stereotypes of persons and organizations.

(c) Plans and goals create the possibility of assigning intentions to an agent, which turns their behaviour into intentional actions.

(d) If the agent role description is sufficiently detailed, its behaviour in response to the events to which it is susceptible become as predictable as that of a machine. Because the agent knows the agent role it is dealing with, it does not need to model the intentions and beliefs of the other agent explicitly because it has plans for all expected outcomes of interaction.

(e) The subjective agent role equivalent of a diagnostic structure description is a subset of the agent role belief set that deals with relations to other agent role instances in its environment that belong to certain agent role classes.

(f) The assigned agent role description may be a generalization of more specific agent role descriptions that each count as health modes of the agent role.

(g) Moreover, an agent may be aware of common fault modes of agent roles it is dealing with, and monitor for signs of these fault modes. The occurrence of such a sign is a reason to diagnose. Diagnosis entails a change of role towards another agent.

7. Other agents may not behave as expected, causing the agent’s actions to fail. This is in essence a diagnosis problem directed towards the environment. The fundamental difference to general diagnosis is that this diagnosis problem is subjective, as the agent perceiving it is itself a component of the system. This subjective point of view helps in keeping the number of MAS configurations to consider limited.

8. Modification of the structure description may be conceptualized as design activity (system) or modeling activity (environment).

3.5 Narratives and diagnosis

We have characterized the problem of adapting business processes and decision support software in public administration, in response to changing society and government policy, as a design & diagnose problem solving cycle, operating on agent roles as components. In public administration, legal knowledge in the form of critical incidents and, for want of a better word, noncompliance storylines
used for monitoring and enforcement has an important function in knowledge management. A central place in these stories is taken by stereotypical agent intentions like the intention to evade income taxes. These stereotypical agent intentions tend to resurface even if changes are made to the specific legal rules to stop abuse. This knowledge has no natural place in traditional forms of legal knowledge representation such as normative rules or legal argument schemes.

The model-based diagnosis problem, based on executable agent role descriptions, is a model for the acquisition of diagnostic compliance knowledge in public administration. In chapter 7, we apply a model-based diagnosis framework to the complex social systems in which large public administration organizations operate. The purpose of diagnosis is to identify problematic agent role instances in a multi-agent system (MAS). The diagnostic agent is part of the multi-agent system, has only limited access to the information exchanged in the multi-agent system, limited ability to control and change the multi-agent system, and actions to obtain information change the state of the system.

3.5.1 Summary of principles

1. A major objective of the methodology is supporting diagnosis of fault scenarios, and translating these diagnoses into revised designs.

2. A traditional model-based diagnosis problem can be viewed as a problem of a single diagnostic agent having a model of the whole system to be diagnosed as its environment.

3. Based on Reiter (1987), an externalized description of a system can be conceptualized as a pair (SD, COMP), where SD, the system description, is a set of sentences, and COMP a finite set of constants (uniquely?) identifying components.

(a) Typically, a system description describes how a system normally behaves, and it often distinguishes a description of structure from a description of function.

(b) The system description usually consists of a description of the behaviour of components and their interfaces, and a description of how interfaces are connected. Although there is a distinction between input and output terminals, the diagnostic system can use the functional description to predict output from output, and sometimes to postdict input from output.

(c) An observation of a system is a set of sentences OBS in (SD, COMP, OBS). If OBS and SD are inconsistent with the assumption that all components are normal, certain of the components behave abnormally. This test is usually implemented with some normality predicate.

4. A diagnosis is a conjecture that certain of the components are abnormal (AB ∈ COMP) and the rest normal.
5. The diagnostic process usually involves making additional observations to rule out hypotheses (monitoring). Analogous is the collection of additional evidence. Note that in social systems one cannot manipulate components as easily as in for instance electronic circuits. In electronic circuit diagnosis the components can often be treated as stateless.

6. Diagnoses can be generated on the basis of component fault models. The diagnosis in this cases conjectures alternative behaviour descriptions for the components that are behaving abnormally.

(a) The set of fault models of a component may be complete: in this case the component must be behaving according to the normal model or one of the fault models. Alternatively there may be unknown fault models.

(b) Additional knowledge may guide the exploration of the diagnosis search space. Diagnoses may be ruled out on grounds of impossibility. A probability distribution of fault models may be known that guides selection of hypotheses.

7. The social system description and the social system diagnostic hypothesis can be characterized as MAS configurations from an objective point of view.

8. Externalized, “objective” description of the social system is however undesirable, because agents are the participating components, and system breakdown often results from different subjective social system descriptions. The existence of social structures must be accepted by the participants in the structure, and should be treated as subjective. Cognitively speaking, a more plausible description of the diagnosis question is that the diagnostic agent(s) sort of participate in the social system being diagnosed, although a diagnostic agent has no fault models for itself.

9. Diagnosis of MAS configurations is hard because agent interaction is complex and topology flexible. Fault model-based reasoning is the only viable approach to automated diagnosis, and MAS configurations should be kept relatively simple.

10. Other agents may not behave as expected, causing the agent’s actions to fail. This is in essence a diagnosis problem directed towards the environment. The fundamental difference to general diagnosis is that this diagnosis problem is subjective, as the agent perceiving it is itself a component of the system. This subjective point of view helps in keeping the number of MAS configurations to consider limited.

11. The richness of messages exchanged between agents makes a model-based diagnosis approach based on normal models and unspecified unknown fault models unfeasible. A catalog of fault models, and actual simulation of fault model based hypotheses, is therefore in our view essential for success.
12. Modification of the structure description may be conceptualized as design activity (system) or modeling activity (environment).

13. Agents in a MAS cannot feasibly run MAS simulations to resolve diagnosis problems. This is a design phase problem, solved by a planning solution. Fault model detection is hardcoded into monitoring functions that detect abnormalities and forward these to process design, resource allocation, or enforcement processes. These monitoring functions are design solutions based on a diagnostic strategy applied in the design phase.

14. A design objective for the agent role methodology is to explain logical nonmonotonicity in legal rules as logical nonmonotonicity in the MAS but not in the individual component. The agent role will however have a belief revision ability (temporal nonmonotonicity).

(a) As pointed out in Reiter (1987), there is a close relationship between diagnosis and default reasoning.

(b) The distinction between executive agent roles and diagnostic agent roles should explain contrary-to-duty patterns in the legal rules.

(c) Burden of proof/production/persuasion should be explained in terms of communication between agent roles.
Figure 3.9: The operations, development, and policy making problem solving cycles.
Chapter 4

Legal Knowledge from an Engineering Perspective

4.1 Introduction

Society uses a wide variety of sources of law, and uses many labels other than legislation to describe sets of written rules: quasi-legislation; administrative rules; codes of practice; guidance; guidance notes; policy guidance; guidelines; circulars; framework documents; outline schemes, and statements of advice. A commonality of these documents is that they are understood to postulate or describe norms that guide or mandate conduct in a given type of situation. This guidance can pertain both to physical behaviour and to decision making processes.

Before one can turn to the question of how to represent legislation and similar documents, one must consider the role these documents play in guiding behaviour.

This chapter deals with the relation between these formalized rules and behaviour. It does not address the problems legislators encounter in managing a large body of such rules, the solutions they have found for these problems, and the consequences this has for knowledge engineers who try to represent the meaning of these rules in logical form: this is the subject of the next chapter. This chapter is based on Boer (2009a), and its essentials are boiled down in the methodological instructions in chapters 3 and 6.

This book takes law to be an institution whose primary purpose is to create normative order by way of formalization – the sources of law specify a formal, institutionalized normative order (cf. MacCormick (1998)) – usually wherever spontaneously arising, informal normative order fails to achieve the desired results. The following rationales are usually associated with the formalization, i.e. the writing down, of norms:

**Accumulating Knowledge** Accumulated knowledge from conduct in the past is written down in the form of norms to guide conduct in the future.
The norms set out the relevant criteria, guide collection of information, decrease the amount of mistakes, and generally allow relatively stupid people to solve complex social problems.

**Consistency** Writing down norms supposedly encourages consistency, fairness, equality of treatment of persons, groups, and organizations in different places and at different times. Norms limit the discretion of the decision maker in treating specific persons, groups, and organizations differently, and therefore reduces bias and corruption. Written norms also make behaviour of others predictable, which reduces conflicts.

**Democracy** The process of writing down norms allows for greater public involvement than the mere making of a decision. The use of norms is in this sense a precondition for effective democracy on a large scale.

**Legitimacy** Written norms contribute to the perceived legitimacy of decisions, because of the reasons above, and allow the decision-maker to cite the source to justify the decision.

Much of the terminology to describe institutions has been strongly influenced by the work of Searle (cf. generally Searle (1970); Searle and Vanderveken (1985a)), although the notion of institution and the closely related notion of constitutive rules were introduced earlier. Searle describes the distinction between two types of rule, being normative – or regulative – and constitutive rules, as follows:

Some rules regulate antecedently existing forms of behavior. For example, the rules of polite table behavior regulate eating, but eating exists independently of these rules. Some rules, on the other hand, do not merely regulate an antecedently existing activity called playing chess; they, as it were, create the possibility of or define that activity. The activity of playing chess is constituted by action in accordance with these rules. Chess has no existence apart from these rules. The institutions (emphasis mine) of marriage, money, and promising are like the institutions of baseball and chess in that they are systems of such constitutive rules or conventions (Searle (1970), p. 131).

This distinction has been taken up by many in computer science & law (viz. Boella and van der Torre (2004)), although often with a very limited application: constitutiveness is taken to be a characteristic of a limited type of legal rules or legal acts. In legal theory in general there is a bewildering array of conceptions of constitutiveness (cf. for instance Mazzarese (1999)), some of which are irreconcilable, and this book will add yet another one.

Both the regulative and the constitutive rule are to be distinguished from rules that merely generalize past observations in the hope that they generalize to future ones. The regulative and the constitutive rule on the contrary only
affect events that happen after the rule comes about, and application of the rules by an agent is presumably the force that makes the consequent of the rule happen. The regulative rule guides behaviour, while the constitutive rule guides interpretation of events in the terms of an institutional reality.

The regulative function of law is generally taken as the defining one of the legal system in positivist legal theory. This is for instance apparent in Kelsen’s notion of a Grundnorm (cf. Kelsen (1991)), which is clearly of the regulative type, or in Austin’s adagium (cf. Austin (1954)) – later developed by Hart (cf. Hart (1961)) – of law as the command of the sovereign backed by the threat of punishment. Normative rules embody the primary purpose of law.

The next section of this chapter will kick off with a discussion of constitutive rules and institutional facts, since the regulative function – even if it is in the end the really important one – is secondary to the function of constitutive rules. Law institutionalizes normative order by way of institutionalizing recognition and evaluation of conduct. The institution and the normative order it institutionalizes are two separate things. This point of view will be worked out further in sections 4.2 and 4.3. Attention to only one of these two aspects, or the insistence that rules are to be either conceptualized from an institutional point of view or a regulative one, will lead to unbalanced knowledge representation.

Normative rules occur in three main variants in natural language: obligations, permissions, and prohibitions. The use of specific auxiliary verbs is however not at all indicative of whether some proposition is normative or not. Both a legislator and a sociologist can utter, for example, the sentence “Tanks adjacent to the hull are not used to store fuel oil”, and in neither case the sentence would be semantically ambiguous. In the first case, the legislator sets forth a rule prescribing how a tank adjacent to the hull should be used. In the second case, the sociologist verifies a regularity, analyzes the relation between the norm and social reality. A legal norm is a norm by virtue of its function and the way it came about, not of its propositional content.

Clearly, normative rules only address conscious human choices. The eruption of a volcano may very well be undesirable, but it makes no sense to try to tell it ought to behave. It is also not very sensible to prohibit making mistakes. Normative rules are information items intended, by their author, the legislator, to influence other people’s choices. The normative rule describes a choice, and the preference the author holds for one of the involved alternatives, and it imposes that preference on the addressees by way of some social mechanism: it is this social mechanism that makes it a norm.

This social mechanism is generally speaking of little importance to knowledge engineering. It becomes relevant in as far as it affects the effectiveness and efficiency of legislation. Since the comparison and evaluation of legislation is one of the considered contexts of application of knowledge components, this report gives this question more attention than most literature in the field.

Normative rules can only work if they are known to the agents addressed by them. They are therefore made explicit in writing (contract, regulation, jurisprudence, etc). This written document prescribes behaviour to agents assigned a certain role, for instance the owner and user of a road, or posits norms
for the creation of artefacts, for example ships, works of art, and tax forms. In other cases it constrains or defines procedures for actions or transactions by agents, for example a survey, hearing, or purchase.

In the world around us, we see that norms are usually adopted, provided they are known. There are two fundamental explanations for violation of norms, assuming in the first place that it is based on wrong choices:

**Unwillingness** The agent prefers the wrong alternative. Conditioning (reward and punishment) is the universal solution for this problem, and according to some influential voices the fundamental function of law.

**Ignorance** The agent failed to understand the law, failed to conceptualize possible choices, or failed to foresee possible outcomes of a choice. If the agent drives on the wrong side of the road, for instance, without the intent to cause a crash, the problem is apparently that the agent failed to foresee the likely crash that would follow. Law also helps solving this problem by commanding people to drive on one side – in this case the side is irrelevant – of the road.

In some cases an agent may be forced to choose contrary to the rules because a legal alternative was not available. This case is usually either not considered a violation, or it is a special case of violation without adverse legal consequences.

Because normative rules are supposedly used by agents other than the legislator, they must be interpreted. The use of normative rules therefore presupposes an understanding of the norm entailed and legal systems only function if these norms are understood. Understanding the basic normative rule alone is unfortunately not enough, as we will see in chapter 5. It is necessary to understand which documents contain valid normative rules, but it is also necessary to understand in which order they are applied. It is necessary to understand when and where they are valid, and exactly whom is addressed.

Rules in the sense discussed address agents by role. The agents addressed by the norm are described, and not identified. The rule applies to you because you are (recognized as) a citizen, driver, legislator, civil servant, judge etc. Not every command is a normative rule. A command that addresses a role filled by one person at the time, for instance the monarch or the prime minister, is still a normative rule. A command that directly addresses you cannot possibly be one.

Role is the most useful way in which legislation is organized. Tax law addresses taxpayers, criminal law addresses state and citizen, private law addresses interaction between citizens, labour law addresses interaction between employer and employee, etc.

Note also that there is no strict distinction between the legal system and the domain it regulates. The application of codified norms by a judge to others, and the imposition of sanctions, is also regulated by normative rules.

The legislator also usually limits itself, although there are accepted limits on the legislative power of the legislator to restrict his own legislative power (viz.
Suber (1990)): logically speaking, there are laws the legislator cannot pass, or there are laws the legislator cannot repeal.

To understand the use of normative and constitutive rules, one must first distinguish between three main roles one can adopt towards these rules:

**Legislators** use normative and constitutive rules to change the behaviour of addressees and bystanders.

**Addressees** adopt the rules in the decisions they make in a variety of roles: the rules both constrain and create alternative ways to achieve one’s goals.

**Bystanders** interact with addressees and use the rules to explain and predict the behaviour of addressees.

The position of bystanders is very often not explicitly considered, but the use of the rules to predict other people’s behaviour is as important in decision making as its direct use by addressees. A neutral reasoning agent can be conceived of as a bystander.

Additionally, to describe the use of normative and constitutive rules we have to spell out the relation between someone’s decisions and someone’s behaviour.

### 4.1.1 Intention and Action

We distinguish between an internal and external perspective, and an ex ante and ex post perspective on decision making. The ex ante and internal perspective of the decision maker who considers a plan, is very different from the ex post and external perspective of the observer who describes someone else’s behaviour in terms of actions, i.e. ascribes a plan to others based on their behaviour.

This relation is mediated by the concept of intention. It has two distinct senses, depending on whether we take an internal or an external perspective. That is, we intend to do things and we do things intentionally. The relation between what we intend to do and what we do is too weak to reduce these senses of intention straightforwardly to each other (cf. for instance Bratman (1984)). This is also recognized in law, which uses certain concepts and fictions to deal with this problem. Ideally, we want to connect intentional action, and its conceptualization in law, to the planning problem type of section ??.

To understand what intention is it makes sense to focus on its ex ante sense: Bratman gives good practical and philosophical reasons for the methodological priority of ex ante intention over ex post intention (cf. Bratman (1984)). Ex post ascription of intention can simply be considered an explanation of someone else’s behaviour in terms of his (ex ante) intentions. Alternatively, we may consider the ex post recognition of intent as the primary one, and simply consider the recognition of your own intentions as a special case of recognition of intentions in general: this interpretation is more in line with Newell’s point of view that knowledge is something that is ascribed to intelligent agents to explain their behaviour (cf. Newell (1982) or section ??).
Observe that intention cannot be simply explained in terms of preferences, desires, goals, etc. One does not usually say that one has the intention to become a famous football player just because one dreams of being one, and one does not generally intend to have the same cake and eat it even if one desires to do so. Intentions are closely linked to specific actions: intention is the commitment to a plan one decided to execute or are executing, or to a design one decided to implement or are implementing (cf. generally Boella et al. (2005) who have a similar view on intentions). One has the intention to do something when one is doing it. Whether the intention started just before execution of the plan, when one “committed” to it, or just after it when one “recognizes” it, is of minor impact.

Cohen and Levesque list the following requirements on a satisfactory representation of intentions in Cohen and Levesque (1990):

1. Contrary to desires and preferences, intentions held by an agent are consistent with each other and with the agent’s beliefs.

2. An intention poses a problem for an agent: the agent must have a plan that he believes realizes it under certain conditions, and he must be committed to executing that plan.

3. The agent monitors the success or failure of his attempts to realize his intentions: failure constitutes a new problem.

A description of behaviour in terms of actions is grounded in the recognition of behaviour as a plan execution. It describes a structure in two different strata: a plan in the mind of the agent that he is attempting to execute, and observable behaviour of the agent that succeeds or fails to execute that plan. It is based on the presumption that the agent, whose reasoning capabilities are limited, must commit to a plan and remains committed to it for a longer period of time, and moreover, that we can predict both intentions and plans with reasonable accuracy.

The agent only takes decisions once he has completed an attempt to execute a plan, either because the plan failed, or it succeeded, or because something interesting came up. Note also that the agent can interleave the execution of multiple plans, and that some plans may involve monitoring things. One may for instance intend to buy something as soon as one gets the chance: plan execution starts immediately, with a monitoring activity, but it will be difficult for other agents to recognize the plan execution.

The other case which raises questions is when the plan requires us to schedule the start of its execution. If the public transport planner tells me that I have to leave in five minutes to catch the bus, I already sort of commit to it by a hasty visit to the toilet. But if I used it the evening before, then I may even have forgotten my plan the next morning.

It is in these cases that one may feel that one had an intention long before one started execution, and these are of course the hardest cases for criminal
courts to crack when they have to judge whether an attempt to execute a plan was initiated.

One cannot simply say, that the agent “has a plan”. The term plan is used in too many different senses. There is a difference between the plan being executed by the agent, plans under consideration by the agent, and “primitive” plans the agent knows of that are normally executable and that the agent uses as components for the composition of plans. Since mainstream planning literature calls all of these plans (viz. i.a. Bratman (1984); Blythe (1999)), or alternatively tasks in the case of hierarchical task network (HTN) planning (cf. Erol et al. (1994)), it is not very helpful in differentiating between a plan being executed, and the plans manipulated in the mental process that resulted in it. Here we distinguish between a plan, which is simply the mental object we manipulate when we are planning, and the task, which does seem to suggest commitment and is the thing we eventually execute.

The following breakdown, loosely based on the description given in Cohen and Levesque (1990), is somewhat arbitrary but helpful:

**problem** (objective, goal) is desired change

**task** can be executed to solve a problem (achieve and objective, goal)

**competence** is to know a method for performing a task, which may involve decomposing it into subtasks

**plan** is a specification of a method for performing a task

**action** is executing a task, by causing events (occurrences of changes), resulting in success or failure

**intention** of an action is the task one executes: it simply refers to execution

The same breakdown should in principle also be possible for designs, which should be treated in analogy with plans. Also addressing designs in this chapter would however add little.

Most of these things belong to the mental realm, except for action, which bridges the gap between observable behaviour in the realm of the senses and the task it performs:

**Proposition 4.1.1.** An action executes a task.

The action is an occurrent: it happens against the canvas of space and time. The classification of intention as a relation between action and task may be contested. Some people may prefer to attach intentions to agents instead of actions, and to the (intended) result or goal of the task instead of to the task. These can however be interpreted as cases of metonymy. It is obvious to connect the intention to the action because of the temporal concurrence of holding the intention and executing the action. The concept action already presumes the deliberating agent. To say that an agent has an intention means nothing when separated from the action.
The preoccupation with goals is a similar case of metonymy. When one blows up a car with a bomb in the hope of killing its driver, one intends to place a bomb in a car, and to terminate the existence of a car, of a bomb, and of the life of a driver: all of these follow from the specified task. The term goal should be limited to the desired death of the driver that was an input to the planning process. The action has an intention, not a goal.

It is arguable that a more detailed representation of intention would demand its reification and connect it to action, task, goal, and agent.

The most complicated ingredient is perhaps the competence to perform a task: presumably this is based on some organization of episodic memory which predisposes the agent to believing it can perform certain tasks, and furthermore is so predictable that other agents recognize task execution. Examples of such “primitive” tasks that can be performed, or recognized, without further ado according to planning literature are such complex procedures as buying a quart of milk, painting a table, taking a plane to Melbourne, etc. To explain where these primitives come from we can appeal to Schank and Abelson’s scripts (cf. Schank and Abelson (1977)), a concept developed for automated story understanding. Section 4.3 connects these scripts to normality, and discusses its relations to normativity.

In law, even buying a quart of milk can however become a complicated affair, because it involves a transaction between two agents. If just action is already complex, transaction adds the complication of their interaction. On the level of transactions we find a new vocabulary – associated with Hohfeld (viz. Hohfeld (1919)) who ironically thought of them as fundamental legal concepts – to gain traction over our problem. This new abstraction level is the subject of section 4.6.2.

4.2 Institutions and Rules

Institutions are, according to a common definition, “structures and mechanisms of social order and cooperation governing the behavior of two or more individuals”, for instance law, marriage, money, democracy, the marketplace, church, school, etc.

Law is an institution whose primary purpose is to create normative order by way of formalization – i.e. a formal, institutionalized normative order (cf. MacCormick (1998)) – usually wherever spontaneously arising, informal normative order fails to achieve the desired results. A similar characterization is already found the work of Geiger, who speaks of law as a mechanism to create normative order monopolized by a central authority (cf. Geiger (1969)). The notion of law as an institution, based on so-called constitutive rules, has been a productive concept in legal theory and computer Science & law, in particular to justify the use of logic programming rules to represent law (cf. generally Boella and van der Torre (2004); Mazzarese (1999)).

An institution is a collective intentional or social entity, i.e. an entity that exists merely because a collective (i.e. group of natural persons) recognizes and
intends its existence. The institution itself is a good example of intention applied to the existence of artifacts.\footnote{See my earlier observations about the interchangeability of the planning and design problem type in section 4.1 and \ref{section:planning_design}.}

The structures of the institution are defined by the institutional facts that make up the institution, and its mechanisms of change are the constitutive rules that specify what constitutes, or counts as, an institutional fact. Conversely, the institutional fact has a constitution base (following Hindriks in Hindriks (2005)), which consists of the application of a constitutive rule to the constituting facts, which are brute facts, yielding an institutional fact.

**Proposition 4.2.1.** A constitutive rule of an institution derives an institutional fact from one or more constituting facts, at least one of which is a brute fact.

**Proposition 4.2.2.** A brute fact, relative to some institution, is a fact that has no constitution base in that institution.

Brute facts are pre-existing and external to the institutional reality constituted by the rules. This does not however mean that they are in any sense “natural”, or non-institutional. The institutional facts of one institution can be the brute facts of another one. If we for instance state that checkmate constitutes winning, checkmate is an institutional fact of chess, and winning is an institutional fact of games. If we consider “winning the chess game” to be part of the institutional language of chess, the relation between checkmate and winning can no longer be considered truly constitutive, unless one expands constitutive to include any terminological axiom. On this point this book follows observations by Hindriks on Searle’s work (in Hindriks (2005)) and not Searle (cf. Searle (1970); Searle and Vanderveken (1985a)), who does insist on separating social reality from a pre-existing one. Hindriks identifies these rules as essential rules of the institution: the ontology of the institution. An institutional fact that merely rephrases other institutional facts using the institutional ontology shares the constitution base of the institutional facts constituted by brute facts.

**Proposition 4.2.3.** An institutional rule of an institution derives an institutional fact from one or more other constituting facts, all of which are institutional facts.

The distinction between institutional and brute facts is very similar to the role of the Breuker’s legal abstract model (cf. Breuker and Den Haan (1991)): a layer of “legal” concepts and relations built on top of a large layer of commonsense knowledge (Valente and Breuker (1995), p.57). In Breuker’s original conception this world knowledge could ground different institutional normative orders in a single consensus reality. This is an attractive proposition for the purposes of comparison of two institutional realities that do not intersect at all, as both would presumably be grounded in the same consensus reality, giving us something concrete to compare.
The (institutional or brute) fact is simply a reified statement about something. One may think of the constitutive rule as a simple institutional qualification of something – the predicate or concept is institutional – or as giving rise to a new institutional thing. This is usually a question of modeling style and requirements, and not a quality inherent in the constitutive rule.

**Example** In an auction, raising a hand constitutes a bid; We can speak of the institutional concept $\text{Bid}$, some institutional object $o$, which is institutional because it is of type $\text{Bid}$, and the institutional fact $\text{Bid}(o)$.

From an ontological point of view the act of raising one’s hand and the act of bidding a certain amount cannot be one thing, because they do not share essential properties. But the constituting act of raising a hand to make a bid and the supervenient institutional act of making a bid are in this case inextricably linked to each other. The same is the case for a constituting act of taking that constitutes a theft. The case is often less clear cut when we are talking about states, objects, substances, etc. In many cases we see that although the constitutive rule states that some brute state constitutes some institutional state, the act to initiate or terminate that state is labeled constitutive. This is not entirely surprising, since we are dealing with intentional things brought about by actions.

A constitutive act in this sense is an act of representation: its function is standing for, or representing, something else. Functions of things are not inherent in the thing: they are assigned by (intentional) agents, and the recognition of the function depends on the observer. Note that assigning a function always introduces the possibility of assessment or evaluation, as things can be better or worse at realizing the function.

This creates an ambiguity, however. We raise our hand at the auction because we intend to make a bid: the action has no other rationale. The thief on the other hand has no intent per se of being labeled a thief: he merely takes that risk. To resolve this ambiguity, we have to distinguish the constituting act from the constitutive act.

### 4.2.1 Constitutive Acts, Speech Acts, and Intent

In law, a legal act\(^2\) is an act that creates a legal fact i.a. because it was intended by the actor to do so. In analogy, consider a constitutive act an act intended to constitute an institutional act. A constitutive rule may specify that an act (for instance raising one’s hand) constitutes an institutional act because it was intended to do so: the constitutiveness of the act is a condition of the constitutive rule. To unambiguously create the institutional fact we have to make a demonstrable declaration of will, and, conversely, to proof the institutional fact one must provide evidence of the intent to produce an institutional fact.

**Proposition 4.2.4.** A constitutive act is an act intended to constitute an institutional act.

\(^2\)Rechtshandeling in Dutch doctrine, Rechtsgeschäft in German doctrine.
As stressed by for instance Jaap Hage in Hage (2007) intent has a central role in bringing about institutional facts, but only in its indirect relation with action: the institutional fact is brought about by performing a constitutive act, and – normally speaking – an action presumes intent to bring about the product of the action. To describe behaviour in terms of actions is to presume intentionality to produce the products of the action: this is why actions can fail. If a driver turns right we are talking about an action, and if the driver happens to hit a bicyclist while doing it, this is an event, and it is an (unintended) side effect of the action which is not part of the essence of the action itself. Without the action the event could not have happened, but hitting the bicyclist is not the action itself, which was aimed at turning right.

Raising one’s hand at an auction constitutes a bid. Raising one’s hand to a friend may however also be an attempt to draw attention, and conventionally constitutes a greeting. Raising one’s hand to a friend walking in on an auction in the central lobby of a cruise ship you were watching constitutes a greeting, but may be recognized by the auction master as a bid, which is an unintended side effect of the action. The auction master’s problem is to determine who participates in the auction, and the setting is in this case not conducive to this determination.

Legal theory has a whole battery of concepts to deal with intent, the failure to bring about what was intended, and the fact that not all effects of actions are intended ones: declaration of intent, intent and conditional intent, attempt, recklessness, negligence, mens rea, etc. Still the operative principle behind constitutive rules and institutional facts is that people to a large extent have control over what institutional facts they bring about: this is the whole point of trying to regulate behaviour by way of declaring sanctions in advance.

A constitutive act is an act of representation that creates some institutional thing that wasn’t there before, by virtue of the intent to create the institutional product of the act alone.

Collective intentionality – Searle’s original requirement for performing constitutive acts – does not necessarily presuppose that collectives are intentional: it is sufficient that its members can make a distinction between I-intentions and we-intentions, since the interpretation of institutional reality happens in each individual mind without recourse to a collective one. I intend to undertake some action because I expect that we interpret the action as representing something else. Searle specifically defends the thesis that it is unnecessary to explain how we-intentions reduce to I-intentions. This is an internalist point of view, and also the one we would usually take on behalf of (the user of) an expert system.

Collective acceptance, an alternative concept introduced by Hindriks (cf. Hindriks (2005)), derived from Searle’s collective intentionality (cf. Searle and Vanderveken (1985a)), can be taken to mean actual acceptance by at least one person besides the actor. It answers a variant on an old philosophical question: when nobody’s around, can your actions be truly constitutive, regardless of your we-intentions? According to Hindriks this is only the case if they are recorded and later accepted as constitutive by the required audience that accepts the act.

This distinguishes the constitutive act from other acts of representation in-
herent in speech acts: if one for instance requests someone to do something, the request represents an attempt to make someone do something, regardless of whether it is accepted by anyone. Similarly, a promise to do something represents an intention to do something, and only when it has a constitutive character gives rise to a duty to do that thing. Think for instance of a threat: a duty only follows from the threat if there is an audience for whom a threat has the conventional effect of producing a duty, which is nowadays not normally the case. On the contrary: for most people the threat constitutes a crime. The only thing that distinguishes threats from promises is that the thing you promise to do is not appreciated by the target audience.

Although there are some types of speech act typically associated with producing certain conventional effects – declarations, directives, promises, etc. – constitutive acts should not be equated or confused with specific types of speech act. Secondly, the fact that some act constitutes another act does not mean that the constituting act is also constitutive: this is only the case if the actor himself intends to perform the institutional act. The constitutive rule can be applied by bystanders to someone else’s actions.

4.2.2 Law as an Institution

Law is an institution, and therefore can be analyzed in terms of constitutive rules and institutional facts, backed up by an institutional ontology. When specifically applied to law we can speak of legal rules and legal facts, backed up by a legal ontology. The legal act, however, is an institutional act only because it was intended by the actor to be one: we cannot properly make the distinction between the constituting act and the constitutive act. Confusingly, theft is therefore merely a legal qualification of a natural act and not a legal act, but that some act constitutes theft is a legal fact, if we follow prevailing doctrine.

**Proposition 4.2.5.** *An act that constitutes a legal act is constitutive: it is intended to constitute the legal act.*

To position the role of intent in institutional reality I will distinguish (roughly following Sartor’s account in Sartor (2006a)) between legal rules that:

1. merely state that some event or situation constitutes an institutional fact,
2. state that some act intended to constitute some institutional fact constitutes that institutional fact,
3. are intended (by the legislator) to enable people to create institutional facts, and

---

Note that this account however lacks the component of ascribed intention: intention is at best implicitly available in the distinction between productive and behavioural characterizations (cf. section 4.5.1) made there.
4. are intended (by the legislator) to enable people to create institutional facts in order to pursue their own interests.

The distinction between the second and third type is tenuous: in some cases the legal rule only confers the benefit of legal recognition to actions that would also take place without it (for instance buying and selling), while in other cases the legislator creates a rule solely for the purpose of creating a recognizable way to achieve a certain legal effect (for instance a permit application procedure for gaining permission for constructing a shed in your garden). The legislator sometimes explicitly has to create a recognizable way of achieving certain novel legal effects.

Each category adds another constraint, and is as such a proper subset of the previous one. The classification is however purely by function: in essence they are all the same rules. If the legislator intends to enable people to create some institutional fact, one usually speaks of (attribution of a) power or competence, and if the power or competence is attributed to help people pursue their own interests one speaks of a potestative right in Sartor’s terminology. The link between pursuing one’s own interests and the introduction of the concept of a right will be treated in more detail on section 4.6.

Pursuing one’s own interests is however a problematic concept. It is easier to approach matters from the angle of pursuing someone else’s interests. Civil servants for instance often have powers – linked to their role of civil servant – that they can only use as part of a predefined administrative task realizing an objective of the administrative body, and outside it they don’t. Within the context of the task they have the power. In KBS this notion is rarely important, as we are usually dealing with routine tasks with a predefined structure.

Sartor discusses these rule variants as types of normative conditionals, and discusses the constitutive or “counts-as” rule as a subtype of them. Although I do not adopt this conceptualization, I do concur with his observation that all of them are simply typical variations of a single underlying pattern: that of a rule connecting antecedent to institutional fact. Sartor gives a more limited interpretation to constitutive and “counts-as” rules, which are in his opinion non-deontic of character. This distinction is problematic: the unlawful taking of a good that belongs to somebody else clearly constitutes theft, but the same formulation also – as a side effect – prohibits the unlawful taking of a good that belongs to somebody else if we understand the concept of theft to be evaluative in character, in this case being unambiguously negative.

Intuitively one would like to arrive at the conclusion that criminal law prohibits theft, but at least the Dutch, German, Italian, and English formulations do not explicitly command anyone not to commit theft. In the next section I will discuss the relation between the constitutive and normative character of rules: here we only come to the conclusion that neither theft, nor the act that constitutes theft is a legal act.

Constitutive acts – in general – can be informal or formal, and by this we usually mean: backed by text. Text is taken to mean any representation preserved in a form whose existence is independent of both sender and receiver.
Formal acts play a central role in law. Law uses formal constitutive acts to formalize constitutive rules, enact constitutive rules, and to repeal constitutive rules. We can therefore also speak of formal rules, constitutive rules created by a formal act, and in the case of the law these are fortunately explicit and unambiguously prior to their effects.

**Proposition 4.2.6.** A formal act is an act of representation, preserved in a form whose existence is independent of both sender and receiver, of an institutional fact, with the intent to be constitutive of the represented institutional fact.

**Example** In a literate society, the legislator often chooses for formal legal acts, since these make it easier to recognize intent (i.e. task performance). Instead of bidding by raising hands, auctions may also accept formal written or electronic bids. An action that results in a message representing a bid to an auction master, with the intent to perform a bid, constitutes a bid.

**Proposition 4.2.7.** A legislative act is a formal act, that creates institutional rules, constitutive rules, and/or institutional facts required for correct interpretation or functioning of institutional rules or constitutive rules.

A powerful criticism of the notion of the constitutive rule is to point out the suspect causal mechanism: if the rule backs the institutional fact, then obviously the rules exist prior to the institution, and if their results are recognized by an audience, then that audience should be able to verbalize the rules in effect.

**Example** MacCormick in MacCormick (1998) discusses the example of queueing – a behaviour that a large part of the world population engages in – to show that the distinction between institutions and non-institutional conventions or normative orders are hard to draw using the concepts of constitutive rules. It is not obvious that the participants in a queue will be able to verbalize the rules in effect. What is clear, however, is that queueing behaviour will be increasingly formalized – in the form of a visible and recognizable queue – as more people start to participate in it and queue-jumping starts to occur.

While we can doubt the construct validity of the constitutive rule in general, we can rest assured that the analysis of institutions should at least work for the law already in operation, at least to the extent that it formalizes its constitutive rules. Only constitutional law itself, which gives the rules for creating and changing the rules, usually came about in a rather messy fashion (cf. for instance Suber (1990)).

Law of course is not a monolithic system. There are many institutions that assume legislative power, and there is no guarantee that these institutions cooperate in harmony in creating their legal ontologies, legal facts, and constitutive rules, or even that they recognize each other. A legal institution can selectively recognize the institutional facts of another institution by way of constitutive rules, which may be formal or informal. It can also adopt the ontology of another institution without recognizing its institutional facts altogether: they merely use the same terminology, but ignore each other’s institutional facts completely. Clearly marking off the boundaries between different institutions, or deciding what it is that is being recognized, is not always trivial.
Proposition 4.2.8. A legal institution can recognize the institutional facts of another institution by way of constitutive rules.

Example A simple example is the recognition of driver’s licenses. The law in many countries regulates who is allowed to drive, in order to allow only people who are competent to drive on public roads. When the legal system finds someone competent to drive it issues a drivers license. This (constitutive) act is intended to create the institutional fact that this person is licensed to drive, and it is formal: it results in a physical representation of this institutional fact in the form of the drivers license. The drivers license is prima facie evidence for the license to drive, but the license to drive may be retracted (constitutive act), in which case the drivers license becomes void if this happens in his absence.

A country may recognize the institutional fact of another country by way of a constitutive rule, or it may simply directly recognize the foreign physical driver’s license as proof of competence by way of a constitutive rule. The two mechanisms have different results in relation to voided licenses, but to find out that the driver’s license is voided the two countries obviously need to exchange institutional information.

Institutions may use very similar, or even exactly the same, institutional ontology without sharing constitutive rules. The similarity is in this case very shallow, as there is no link at all between both institutional realities.

4.2.3 Constitutive Rules and Ampliative Reasoning

The question is whether the use of constitutive rules should be represented with standard material implications, or as a kind of ampliative reasoning. Institutional rules define the institution’s ontology and are properly modeled by logical entailment. Constitutive rules however perform a mapping from one ontological stratum to another ontological stratum. Each of these strata have their own space of possible models as described by their ontology, and there is no guarantee that constitutive rules always perform an unambiguous mapping from one stratum into the other one. Consider the following simplified real world example:

Example Let us say that 1) the taking of something constitutes theft, and that 2) theft must have been performed with the intention to appropriate (cf. generally Sartor (2006a) for the example). Observe that this example illustrates that although we are “deriving” institutional reality from brute reality the inference arrows seem to point the wrong way sometimes. The first rule is a prima facie a sufficient condition for theft, and the second one a necessary one. The inevitable conclusion is that constitutive rules may indeed be “defeated” by brute reality. If you take something from the shelf in the supermarket, you do not have the intent to appropriate it: there is no acceptable mapping from brute reality to legal reality if we interpret the constitutive rules as entailments. A brute reality and a legal reality that don’t match is simply not part of the model set described by the source of law; it describes the part of brute reality that matches with it, not the part of it that doesn’t. The legislator is not being illogical, but simply has another purpose for logic than the legal knowledge engineer.

There are several tricks to avoid this conclusion, but we will run into trouble if in the future we would have to add 3) illegal copying (also) constitutes theft.

We can combine the rules into one by making the necessary condition part of the sufficient condition, but now the appropriation is not necessary.

---

4The Dutch version has a “with the intent to” sentence fragment in their variant of rule 1, leading to a different formalization.
This is a common approach. The rule now settles for an unknown intent to appropriate. The problem is also that the rule has to be changed to accommodate rule 3 about copying. This is the solution chosen by Sartor (2006a) and undoubtedly many others.

We may argue, quite plausibly, that constitutive rules only map from legal reality to the brute one, and take the position that all constitutive rules therefore specify necessary conditions.

If we add rule 3 theft would require taking a good and copying information instead of or, which can’t be the intended interpretation, so we still have to modify a rule to accommodate a new rule.

Reformulating the rules in this way is not a good idea. Both reformulations above try to answer the question which thesis is defeated, taking into account ontological stratification: the first one presumes a settled brute reality and explains it in terms of legal facts, while the second one assumes a context in which one is explaining a settled legal fact in terms of a matching brute reality.

These approaches are a correct interpretation of the use of stratification: one takes one stratum as the independent variable and the other one as the dependent variable. But the source of law should not be directly represented in these ways: the theory put forward by the source of law is after all perfectly consistent in itself.

The solution, based on the assumption that legal facts only come into existence if they are consistent with the settled (brute and institutional) facts, is to guard the derivation of the legal fact with the requirement that this derivation should be consistent. The rule must, in other words, be defeasible to be useful.

**Proposition 4.2.9.** A legal fact only comes into existence if it is consistent with the settled facts.

For purposes of application in a KBS one also needs to decide whether one assigns a burden of proof to the user. In this case one might for instance want a separate argument for the ascription of intent instead of allowing intent to be justified by default by the taking itself, since the intent is one of the constituting facts. Also this constraint does not break the isomorphism requirement: it clearly refers to rule 2 only. These constraints are not directly relevant for the reusable representation of the source of law: one does not automatically assume the burden of proof just because one applies the rule to explain people’s behaviour. Burdens of proof must be met by formal justifications, not by reasoning in general.

**Proposition 4.2.10.** A burden of proof can be represented with a constraint.

### 4.3 Norms, Normality and Normativity

We commonly make a distinction between normality and normativity. Normality refers to the norm from a descriptive perspective: it merely describes an observed regularity in the behaviour of instruments or agents. Normativity only applies to the behaviour of agents, and communicates something in addition to the mere regularity of behaviour: the behaviour is motivated by the recognition of the norm by those whose behaviour is consistent with it.
Normality and normativity are different things, and obviously should be treated as such. We learned long ago from Hume in Hume (1739) that “ought” does not follow from “is”. No constellations of facts can ever prove the truth of a norm. The norm can be communicated between agents, and it can be recognized and adopted, but it is never the spontaneous result of logical reasoning from facts.

There are however a number of ways in which normality and normativity, different sides of a same coin, interact. In the commonsense conception of justice this relation is clear, and fairly well-understood. Montaigne already pointed this out in de Montaigne (1580):

I am prepared to forgive our own people for having no other model or rule of perfection but their own manners and behaviour, for it is a common failing not only of the mob but of virtually all men to set their sights within the limitations of the customs into which they were born.

The very word morality derives from mos or mores (in plural) maiorum, the ways of our ancestors. That custom is the basis for what is moral seems to involve a shift from what is to what ought to be.

Normativity as a source of behaviour is the traditional province of Computer Science & Law and Legal Knowledge Engineering. Normality is the province of descriptive sciences, and in Knowledge Engineering is implicit in notions like defaults (cf. generally Kolovski et al. (2006)) in reasoning, world knowledge and common sense knowledge (cf. Breuker and Den Haan (1991)), or Schank’s scripts (cf. Schank and Abelson (1977)).

Not all norms have something to do with behaviour of intentional agents. The norm is an epistemological concept identified by its epistemic role in a type of problem solving and not something that exclusively belongs to the vocabulary of the legal domain, or to that of morality. A norm is a standard of performance, a measurement scale. It is used whenever we are assessing something, regardless of domain (cf. Valente and Breuker (1995)).

The norm is conceptualized differently in different contexts: When assessing readings from a broken instrument – for instance some circuit board – a norm is for instance the distribution of scores obtained from a correctly functioning norm group, or some margin around the specified ideal functional mapping from input to output that describes the expected and intended behaviour of the instrument. The expected behaviour of the instrument is also intended: the instrument was designed by an intentional agents with some intended purpose in mind. This process shares little with law except the epistemological roles (assessment, norms, case, qualification) in the problem solving process.

In sociology, a social norm is a pattern of behavior expected of an agent within a particular society in a given situation. In this context the norm is interpreted as directly reflecting some preference of the involved agent; Given that an agent is aware of a number of behavioral alternatives and made a choice between them, we can infer that the agent revealed a preference since choice
and preference are interdefinable. If we observe a pattern of agents revealing the same preference and suspect that there is some social mechanism that explains how that shared preference arose, we may infer that there is a social norm that explains that preference. What makes the behaviour normal is simply the fact that it is expected.

There is a presumption, for instance in predicting the consequences of legislation, that people will normally do what is normative. In other words, it is normal to comply with the norms.

In legal theory we also find mention of custom as a “source of law”. The unlawfulness of taking a good (theft) is for instance judged against a cultural context. Is it normal to take something from a shelf in a supermarket? Is it normal to take your neighbour’s ladder to get your cat out of a nearby tree? The assumption is generally that what is normal is lawful, and legislation prohibiting what is normal will be less easily accepted.

Not recognizing this can lead to serious overestimation by legislators of their power to change society. A legislator that likes to be in control should limit himself to instructing people to do what they are willing to do.

Normality and normativity are just two ways of explaining patterns in behaviour: something is normal according to the observer because the observed pattern happens to reveal it to the observer, and it is normative if it motivates (or “causes”) the choices that lead to the pattern. Normality appears as a social fact and a social value, because it is used as a standard of reference against which deviations from normality are evaluated. The normal has both factual and normative force. Associated with the former is custom which is revealed through observation as a matter of fact. Associated with normativity are morality, ethics, and the law.

To explain how normality and normativity relate, and what effects it has that are relevant for representing and evaluating legislation, I will introduce the concept of status quo bias. It is central to the formation of common sense standards of justice or fairness, and can for instance be taken into account in comparison and evaluation of legislation on behalf of the legislator to judge the effectiveness and efficiency of legislation.

These concepts are all related to a version of economic theory called prospect theory (as opposed to utility theory, which is almost universally considered a standard for rational decision-making), credited on Kahneman et al. in Kahneman et al. (1986). His account, based on results from experimental psychology, revolves around the reference transaction, and its role as a yardstick for what is fair. That is, what we become accustomed to (the status quo) attains a normative status against which deviations to our disadvantage are considered unjust.

Status quo bias refers to the finding that in games an option is more desirable if it is the status quo for no other reason than that it is so designated (cf. Kahneman et al. (1991)). Status quo bias is often discussed in relation to the endowment effect and loss aversion (cf. Kahneman and Tversky (1979)). Status quo bias differs from the other two in that it does not even depend on framing changes in terms of (possible) losses and gains.
The endowment effect refers to the differential weight placed on the value of an alternative depending on whether one “possesses” the alternative and is faced with its loss or whether one does not possess it and has the potential to gain it. Losing a given alternative, which is part of one’s endowment, is felt to be a greater loss than the corresponding chance of gaining it when it is not part of one’s endowment. Losses are more heavily weighted than foregone gains.

Furthermore, a certain degree of inertia is introduced into a choice process since things that are already included in the individual’s endowment will be more highly valued than those not held in the endowment. Kahneman and Tversky introduce the term loss aversion to capture the relative steepness of the loss-portion of this value function relative to that on the gain side of it.

Kahneman et al. in Kahneman et al. (1986) developed the idea of a reference transaction: a relevant precedent that is characterized by a reference price or wage, and by a positive reference profit. In the course of life, certain practices and patterns establish themselves and attain the status of normality. If someone disrupts this normal course of affairs, then the disruption will be considered unfair if:

- it is to the advantage of the other and to your disadvantage, and
- if there is insufficient justification for the disruption.

The involved parties are entitled to the terms of the reference transaction. The reference transaction, as a custom, is part of one’s endowment.

Example Consider the following example from Kahneman et al. (1986): A hardware store has been selling snow shovels for $15. The morning after a large snowstorm, the store raises the price to $20. The reference transaction, given by precedent, is marked by a price of $15. The price increase transgresses the reference transaction. In a telephone survey, 82% of the 107 respondents deemed the price increase to be unfair.

Kahneman et al. (cf. Kahneman et al. (1986)) claim that individuals consider the reference transaction to be fair because it is normal: individuals do not have any anterior criterion of justice which informs their judgements of fairness. Kahneman et al. even suggest that any stable state of affairs tends to become accepted as fair eventually. The reference transaction consists in what people have become accustomed to but it also becomes a normative yardstick for assessing the fairness of deviations from itself. This is perhaps philosophically suspect from Hume’s point of view at least, but has also been recognized as an important factor in the sociology of law (cf. for instance Geiger (1969)).

Kahneman’s work in psychology has had quite an impact on economics, eventually resulting in a Nobel Prize for economics in 2002, because it attacks the fundamentals of utility theory as a predictor of actual human behaviour in a systematic and empirically well-founded way.

The notion of a reference transaction is naturally connected to the script in knowledge representation, and my earlier tentative specification of what competence is: to know a method, gleaned from episodic memory, to perform a task. If the snow shovel is made significantly more expensive, or the quart of milk is
no longer for sale, etc., this is unfair because it takes away something from us: the potential to do certain things following a predictable script.

The champion of scripts in knowledge engineering is Roger Schank (see Schank and Abelson (1977)). The key idea involved in scripts is that our knowledge of concepts, events, and situations is organized around expectations of key features of those situations. Many choices are made in the context of a script: you do the same thing every time as long as the salient features of the scripted situations don’t change.

In Schank’s theory, memory is episodic, i.e., organized around personal experiences rather than semantic categories. Generalized episodes are called scripts. Schank’s ideas were developed in the context of story understanding, but have also played a key role in the development of hierarchical and case-based planning algorithms (the elusive store of skeletal plans and primitive plans; viz. Erol et al. (1994)). Scripts are also organized around the social roles (buyer, employee, customer, student, citizen, father, etc.) one plays in life (see for instance Masolo et al. (2004) for more on social roles).

It only makes sense that institutional settings also get internalized as scripts, as reference transactions, i.e. become the normal way of doing things, and a guide for predicting the behaviour of others.

The legal normative rule is a constitutive rule with the purpose of creating normative order. It does so by way of positive or negative qualifications of behaviour. The unlawful taking of a good that belongs to somebody else constitutes a theft, theft negatively qualifies the taking, and gives license (through other rules) to other agents to do things to the thief that are generally speaking not in the thief’s interest. This regulative function of a fraction of legal rules is however not the only way in which the law promotes normative order.

Both normative rules and other constitutive rules of the institution of law tend to affect what is perceived as normal. Legislation, in so far as it induces certain types of behaviour, has an impact on habit formation and hence on the formation of the standards of justice of the future. If the legislator sets a standard for legally recognized sales transactions, actual sales transactions will tend to conform to the prescribed model as it gradually becomes the new reference for sales transactions.

Normative rules qualifying undesirable behaviour in sales will be phrased in terms of a deviation of these reference transaction: in the interest of predictability it is therefore a good idea to follow that script, and if someone deviates from the script to your disadvantage and their advantage, it is only natural to presume that what he does must be illegal.

4.3.1 Changing Legislation

As an observation about changing legislation the reference transaction is very relevant. Firstly, it predicts, as noted, that the addressees of changed legislation will generally speaking cling more fiercely to existing reference transactions than utility theory would predict.
Secondly, it gives a basis for understanding norms of analysis as understood by Hettich and Winer: the norms used to evaluate proposals for legislation (cf. Hettich and Winer (2000, 1988)). We can expect the following from the combination of Schank’s scripted behaviour in planning and the formation of standards of justice around Kahneman’s reference transactions:

- Stakeholders will usually evaluate a change in legislation by considering roles – buyer, employee, employer, etc. – they typically or regularly play in reference transactions, and not their future position as a rational agent that could change roles. The buyer of a snow shovel will not contemplate becoming a seller of snow shovels. Political lobby organizations that try to influence legislation obviously do so from the conception of a role the members they represent play.

- The legislator will also usually take into account the reference transactions created by the existing legislation and evaluate the new one as a deviation from it. It is possible to extrapolate roles in these reference transactions to populations filling the role, certainly in macro-economic modeling of taxation and social security\(^5\). The estimated number of people playing each role will often be considered constant in the legislator’s evaluation, except when “social movement” from role to role is the purpose or a salient aspect of the change.

- It is easier to change non-salient aspects of the reference transaction than salient ones. Making people pay more often by making the transaction occur more often for instance works better than making them pay more in the transaction.

The appropriate normative yardsticks for evaluation of legislation measure deviations from abstract reference transactions. The norms and concepts by which a new version of legislation is going to be judged are probably largely based in the existing version. Changes made will be Pareto improvements – at least one individual better off, without making any other individual worse off – but in terms of social roles identified in the relevant reference transactions and not of individuals, which would be an impossibly hard problem. In Tanghe (2004), Tanghe also makes this distinction in political discourse focusing on inequality and its relation to injustice: this concept in its most widely accepted sense revolves around equality of ability to assume social roles, Tocqueville’s equality of conditions, but this form of equality has arbitrary results when extrapolated to specific populations of one’s choice.

Peacock (2004) makes a more ominous prediction about the legislative process: the reversal of policy measures is more difficult than their introduction, particularly if the losers of a reversal are clearly identifiable. Consequently, a government should consider this effect of its policy measures before implementing them. If a policy is intended as a short-term measure, it may,

\(^5\) Dutch readers will be familiar with the continually updated koopkrachtplaatje (roughly purchasing power table) and the population segment they are in.
when it comes to reversing it, have become so customary that reversal is very
difficult or nearly impossible.

If adoption of legislation depends on habit formation, then there is a differ-
ence between the immediate effects of legislation, and the effect after some time
has passed. Whether and how quickly agents will change behaviour depends
on how salient the changes in legislation are. Those for whom the changes are
salient may even capitalize on the change process itself.

It is therefore reasonable to assume that evasive behaviour will set in after a
number of relevant decision points where the changed legislation has an effect.
In a comparative exercise involving different versions of legislation this can be
approximated by an iterated comparative analysis: one for the initial situation
after legislation has changed and several more representing stages where a cer-
tain percentage of the involved agents have changed their behaviour to evade
negative consequences of persisting in making the same choices in the same sit-
uation. At some point, we assume, all will have adapted and the envisioned
positive effect of the change in legislation comes into effect.

Example People may not adapt at all unless the government invests a lot in enforcing com-
pliance. A recent example in the Netherlands was the introduction of a obligation to identify
yourself if the police asks for identification when you are a possible suspect of a crime or
misdemeanor. Over 40,000 people were fined for not being able to identify themselves in less
than 8 months, and of those 40,000 more than 12,000 went to court to appeal against the fine.
These 40,000 people probably represent a significant percentage of the people that were asked
to identify themselves in the first place, mostly because of traffic violations. Does the change
simply lack salience for people that don’t envision themselves in the role of criminal suspect?

Example People may however also adapt their behaviour before a change sets in. In some
cases the majority of the population are alerted to this possibility, because of media attention.
An interesting example is the tax deductibility of interest payments on mortgages in the
Netherlands. Every time this subject is discussed in the political arena it causes a scare,
since the expected effect of abolishing tax deductibility is a collapse of the real estate market.
The expected consequence, given that incomes before taxes remain the same, is that after the
change many people will be able to afford only roughly 70%-80% of the mortgage payments
they are making now. Many people will therefore be forced to sell their house, and the new
buyers will be looking for a house of about 70%-80% of the price they would have been willing
to pay before the change. Therefore the value of houses will drop to about 70%-80% of the
value before the change. Every homeowner will start considering whether it is a good idea to
realize income from the capital invested in their house now, and move to a cheaper, or rented,
house.

It is probably not a coincidence that this example again comes from taxa-
tion, and the previous one from criminal law. Taxation appears to be governed
more by the financial rationality prescribed by economists, presumably because
it is explained in these terms by the media and professional advisors. Criminal
law appears to be more inert, presumably because there is a strong moral un-
dercurrent in criminal law and criminal law lacks salience for people that don’t
envision themselves in the role of criminal.

4.3.2 Representing Normal Behaviour

Kahneman’s reference transactions directly relate to the perceived competence
to perform mundane or “primitive” tasks. The effect of constitutive rules is to
create a script through which one can achieve certain legal effects, i.e. recognized by others. If these effects are believed to be beneficial by the agent, the agent will generally speaking use the script in planning. Conversely, other agents will ascribe plans to the agent based on the same scripts and assumptions about his beliefs and desires, and recognize the legal effects he brings about. Normal behaviour is simply expected behaviour.

If legal effects of constitutive rules are believed to be detrimental by the agent, the agent will use the script in planning to recognize bad plans. The most important rule of this kind is obviously the normative rule. Agents will however also evade for instance liability to taxation, or legal effects that create costly duties, liabilities, or disabilities.

**Example** For instance: if you need a building permit for a shed higher than 2.5m, this is an argument against designing such a shed. Another nice example comes from the CLIME project (cf. ?): There is a maximum size constraint for ships allowed entry into the Panama canal called PANAMAX. This rule is however irrelevant for operators of ships: ship designers all over the world use it as a hard constraint for ship design.

We apply the notion of “normal” behaviour both to objects and to other people (and perhaps, following Dennett (1987a), to ourselves). Certain categories of objects, tools, machines, artifacts generally, have a function or instrumental role. We have designed them for the purpose of exhibiting certain predictable and useful behaviours when we interact with them in certain ways, and this allows us to perform certain tasks. We also classify them by that function. If their behaviour deviates from the expected one, usually in a detrimental way, they are broken and need to be repaired or replaced.

When interacting with other people we do exactly the same. To buy a quart of milk, you have to find someone who can fill the agent role of seller of milk, etc. We judge deviations from expected behaviour in the same way, except that the remedies are of a different character. Masolo et al. (2004); Hoekstra et al. (2007a) point out that social roles are identified by the set of actions that can be performed in that role. The social role is also a separate object whose existence depends on the “brute” object in the stratum below. This is for instance shown by the application of terms like “good” to roles like seller, student, or cook: the properties apply to the agent in the role, but not to the agent separately (cf. Hansson (2002)). Legal roles are simply social roles in a legal institution. Roles are functions when applied to objects, and agents when applied to persons as the active and efficient cause of events. Agent roles that depend on recognition, and are therefore institutional, are social roles, and if they depend on recognition of the law they are legal.

Sets of actions connected to agent roles can be composed into scripts with interacting participants, who each take a role. To recognize a series of events as an instance of the script, is to expect that the other agents will act according to their role. If they deviate from the script, a new explanation of the situation is needed. Executing a task is to play your role in a suitable script you selected to solve a problem or achieve a goal.
4.4 Legal Normative Rules

There is a simple criterium to distinguish normative rules from other constitutive rules: the normative rule represents a norm that can be violated or that can cancel a violation of another norm. The normative rule can be defined in terms of the institutional legal fact it can create: the violation of a norm, or the cancellation of the violation of a norm.

As already indicated in section 4.1, normative rules come in the three major flavours of obligation, prohibition, and permission. The significance of the normative rule of the obligation or prohibition type in legislation is in its effect as a constitutive rule: by stating that one ought to do or bring about \( x \), it states (as a matter of terminology) that not doing or bringing about \( x \) is a violation. It is the institutional qualification violation, which is inherently evaluative in character, that opens up the interpretation of the act as wrong in the deontological sense. It is also this interpretation that makes deontic logic an obvious tool for representation of law. The effect of a permission is to cancel what would otherwise be a violation: it only means something in the presence of other, conflicting obligations or prohibitions.

**Proposition 4.4.1.** A legal normative rule is a constitutive rule that derives its violation, or the cancellation of a violation, from one or more constituting facts.

The institutional ontology of law incorporates inherently evaluative concepts like violation, duty, crime, theft, etc. because it is intended to evoke this deontological interpretation, just like the institutional ontology of chess appeals to concepts like winning and losing because it is intended to evoke competition, and a manufacturer of hammers calls its products hammers because it intends to evoke an evaluation of their usefulness for delivering blows, which would not be conveyed by calling the hammers paper weights. The purpose of law is to institutionalize normative order.

The concept of violation also evokes the concept of punishment. Not because violations are always followed by punishments, but because punishments are by definition preceded by violations. The preceeding violation is the terminological conditio sine qua non of punishment\(^6\) Punishments are obviously not just disagreeable consequences of one’s actions, but intentional and predictable counteractions by others intended to suppress violations.

As already suggested in section 4.2, the following should also be considered (inter alia) a normative rule: “The taking of a good, that wholly or partially belongs to another, with the intent to unlawfully appropriate it, constitutes theft and will be punished with a prison sentence of at most four years”. It is not the verb used or any other structural regularity in the expression that identifies the normative rule, but its inherent evaluativeness. It works as a standard constitutive rule for determining a legal fact of theft, negatively qualifies it

---

\(^6\)Hart and Honoré (1985): “Legal responsibility is the liability of a person to be punished, forced to compensate, or otherwise to be subjected to a sanction by the law.”
by using a term – theft – that’s inherently negatively charged\(^7\), and concludes with an institutional rule capping the length of prison sentences for an act that constitutes theft\(^8\).

In discourse we encounter the use of normative notions like obligation both in a generic sense, as an identification of a type of rule, and in a context relativized to specific objects. Consider the following two examples:

1. If (seller) offers to sell some (good) to (buyer), and (buyer) agrees to buy the good then (seller) has an obligation to sell the (good) to (buyer).

2. I have an obligation towards John, to sell him the painting I offered him for sale if he agrees to buy the painting.

The first one is a rule. The second one is partially filled in script based on the rule that describes a legal “position”: it indicates that some of the conditions for deriving a violation of the rule have already been met, and that if John agrees to buy the painting, which is outside my control, not selling the painting to him would constitute a violation. One is in a “state of obligation”.

Proposition 4.4.2. A legal normative rule is a constitutive rule that derives a state of obligation to do something, or the cancellation of such a state of obligation, from one or more constituting facts.

Section 4.2.3 already showed an example where a single constitutive rule is represented by multiple logical rules. The normative even seems to create multiple institutional facts, depending on its context of use.

Obligation is traditionally represented by an operator \(O\), ”it ought to be that”. Depending on the kind of logic and the type of problem considered, knowledge engineers choose for different conceptualizations of normative rules. Logical accounts of obligation explain certain properties of a family of concepts revolving around subjunctive betterness (cf. Dayton (1981); Makinson (1999)): they mainly express that certain entities would be better than others.

There are several rule-based approaches that try to approximate norms in rules with notions like violation or obligation (cf. for instance Jones and Sergot (1993)). A standard translation of obligation \(O\alpha\) to violation \(V\) is \(\neg\alpha \rightarrow V\), expressing the intuition that \(\alpha\) is obliged if and only if we consider it a violation to do \(\neg\alpha\). This works for trivial cases, but amongst others fails to account for permissions. It’s dyadic variety \(O(\alpha | \beta)\) means that \(O\alpha\) if \(\beta\), hence \(\beta \cap \neg\alpha \rightarrow V\).

More advanced is the functional approach of Valente in Valente and Breuker (1995), that depends on a value function \(v\) mapping from descriptions to the ordered set \{allowed, silent, disallowed\}, where violation obviously maps to disallowed:

---

\(^7\)See for instance Hansson (2002) for more about the relation between concepts and value statements.

\(^8\)To represent this last rule transparently, one would first need to fix an interpretation of the more general rules of criminal procedure.
\[ v(c) = \begin{cases} 
\text{disallowed} & : O(\alpha \mid \beta) \land c \in \phi \setminus \beta \land \neg \alpha \\
\text{disallowed} & : F(\alpha \mid \beta) \land c \in \phi \setminus \beta \land \alpha \\
\text{silent} & : O(\alpha \mid \beta) \land c \in \phi \setminus \beta \land \alpha \\
\text{silent} & : F(\alpha \mid \beta) \land c \in \phi \setminus \beta \land \neg \alpha \\
\text{allowed} & : P(\alpha \mid \beta) \land c \in \phi \setminus \beta \land \alpha \\
\text{silent} & : P(\alpha \mid \beta) \land c \in \phi \setminus \beta \land \neg \alpha \\
\text{silent} & : \text{otherwise} 
\end{cases} \]

c \in \phi \mid \text{means: } c \text{ is a member of the extension of } \phi \text{ (i.e. } \phi \mid \text{ is the set of worlds } w \text{ such that } M, w \models \phi). \text{ Its strength is in dealing with permission and handling conflict between norms (cf. for instance Winkels et al. (1999)). A weakness is that it cannot be used to detect a contrary-to-duty obligation. Valente and Breuker (1995) is satisfied with the observation that this approach does not lead to the contrary-to-duty paradox.}

More flexible is the modal deontic logic approach, which turns obligation, prohibition, and permission into modal operators and works from the observation that obligation and prohibition are interdefinable \((O\alpha \equiv F\neg\alpha)\), and obligation implies permission \((O\alpha \rightarrow P\alpha)\). What deontic logics are often good for is capturing the uses of obligation in a positional sense, as explained above.

The modal interpretation of \(O\) is a nonempty model set of deontic alternatives in such a way that what ought to be is the case in its deontic alternatives. These alternatives ideal in the sense of being subjunctively best worlds. Thus if \(O\alpha\) is true of an object, it means that it would better if \(\alpha\) were true. It does not mean that it is obligatory for anyone to do something which might make \(\alpha\) true.

The dyadic \(O\) operator is similar. \(O(\alpha \mid \beta)\) is true of an object if the condition given by \(\beta\) determines some non-empty class of deontic alternatives in which \(\alpha\) is true. It means that given that \(\beta\), it would be better if \(\alpha\) were true. It does not follow from this that given that \(\beta\) it is obligatory for anyone to do something which would bring about \(\alpha\).

Orthodox deontic logics do not deal with the circumstance that the only way to make \(\alpha\) true involves forbidden actions, that making \(\alpha\) true is impossible, or that although bringing \(\alpha\) about would certainly be better, it is also not my obligation but somebody else’s\(^9\).

In the literature this discussion is known as ough-to-be vs. ought-to-do (cf. d’Altan et al. (1996)): they are widely assumed to be mere reformulations of the same thing, even though it is wellknown from practical applications that they are not. The assumption that if it ought to be that \(\alpha\) (given \(\beta\)) is equivalent to saying that it would be better if \(\alpha\) were true (given \(\beta\)) is built into all orthodox deontic logics.

Deontic logics also generally don’t represent violations directly. Legal obligations do not express the goodness of something but the given that the decision to perform or not to perform the action referred to as obligatory has consequences. Interpretation of the violation of the obligation cannot be separated

---

\(^9\)This clearly separates legal obligation from ends-oriented accounts of morality.
from a theory of decision making and action that answers questions such as what was within the power of the decision maker, what did the decision maker intend, what did he foresee, and what did he try to do? Deontic concepts should be explained in terms of betterness, but also of, firstly, habit formation, and of choice, of beliefs, of decisions, of violations, of sanctions, etc.

Since deontic logics appear to capture the same properties of betterness as preference logics do, but without reference to the social mechanism of norms, there have also been attempts to relate the two concepts (cf. for instance Hansson (2001); van der Torre (1997); Tan and van der Torre (1996); Boer et al. (2005a,b)). Preference logic (cf. van Benthem et al. (2005); Doyle (2004); Doyle and Wellman (1994); Doyle and Thomason (1997)) captures the use of preference (here strongly $\prec$, and weakly $\preceq$) and indifference statements $\equiv$ between propositions. Against preference logic the same criticism applies: it should be explained in terms of choices, intention, beliefs, action, and not just in terms of other preferences.

This connection to decision making and action will be made in section 4.5.

Figure 4.1: An entity-relationship diagram describing the salient structure of obligations and prohibitions.

**Proposition 4.4.3.** A legal normative rule is a representation of a legal norm with the intent of being constitutive of that norm, regardless of whether it is phrased in terms of violation or in terms of a state of obligation (or the cancellation of those).

Putting the axiological interpretation in terms of betterness, and the institutional interpretation in terms of constitutiveness of legal facts together results in the structure described in figure 4.1. A legal norm is an a priori obligation or prohibition (i.e. not a positional one), it applies to a certain case, allows a certain case – the allowed case – and disallows a certain case – the disallowed case. Instead of saying that the norm disallows the case, we can also say that
the case violates the norm. The allowed and disallowed case are both subsumed by the case to which the norm applies. Besides that they by definition form a complete partition of the case to which the norm applies, i.e. all cases to which the norm applies are either allowed cases or disallowed cases. This is true of the obligation and the prohibition: they are simply two different ways to put the same thing into words: the obligation explicitly mentions the allowed case, and leaves the disallowed one to common sense inference, while the prohibition makes the disallowed case explicit.

The nature of the betterness relation between allowed and disallowed case is from the perspective of the addressee justifiable from the consequences of violation of the legal norm, while the legal norm is itself an expression of a preference of the legislator.

The permission, shown in figure 4.2, is different. The permission allows something, but it doesn’t disallow anything. The logical complement of the mandated case is here simply called the opposite, following existing practice by some authors (cf. Valente and Breuker (1995); Hohfeld (1919). It is the complement of the allowed case subsumed by the case to which the norm applies. Note that the obligation is subsumed by the permission. One can easily imagine a more generic subsuming concept that just “applies to a case” for constitutive rules in general.

4.4.1 Subjunctive Betterness

As noted a very important aspect of normativity can be captured by the notion of subjunctive betterness (as used by Dayton (1981)) to make clear that both deontic and preference logics take the same stance towards the concepts that are
being formalized, i.e. that both express the axiological concept of subjunctive betterness, and that both are incomplete formalizations of the thing they claim to formalize. Subjunctive refers to the grammatical mood of the type of expressions covered by expressions of obligation and preference. The subjunctive mood is used for subjective expressions, indicating wish, command, possibility, counterfactuals, etc.

In the case of deontic logics this means that an obligation “given $\beta$ it ought to be that $\alpha$” is considered a paraphrase of “given $\beta$, $\alpha$ would be better than $\neg\alpha$”.

Central to deontology is the notion of choice:

**Proposition 4.4.4.** Deontic choice $O(\alpha \mid \beta)$: if an agent has the choice between $(\alpha \land \beta)$ and $(\neg\alpha \land \beta)$ then the agent should choose $(\alpha \land \beta)$.

Beware of interpreting $\alpha$ as an action: the alternatives may concern both descriptions of actions and situations, as long as situations can be conceived of as productive characterizations in the sense that social and legal norms only speak about situations controlled by human action.

My preferences are supposed to be revealed by my actions. If “I prefer $\alpha$ over $\beta$” then “$\alpha$ would be better than $\beta$” to me. I may prefer not to violate my obligations, but I can choose to do so; I cannot choose against my preferences. I can pretend to, but by doing so I simply reveal my real preferences: the notion of preference is purely tautological. If the creation of a normative rule is intended to be constitutive of a norm, then the legislator by definition expresses the preference captured in the rule.

A notable difference between deontic statements and preference statements is that the deontic statement (given $\beta$, $\alpha$ would be better than $\neg\alpha$) usually neatly partitions the better and the worse things, and limits its scope of application, while preference statements ($\alpha$ would be better than $\beta$) often don’t. The preference statement $\alpha$ would be better than $\beta$ for instance tells us nothing about the relative ordering of $\alpha \land \beta$ relative to $\alpha$.

The uniform transparency of deontic statements comes at the expense of a limitation in methods that can be used to combine deontic statements: the only method of combination over conflicting expressions is the choice between conflicting deontic statements, while more general theories of preference and utility have other combinative methods at their disposal.

The deontic operators may be reduced to preference statements as follows:

\[
O(\alpha \mid \beta) : \beta, \alpha \succ \beta, \neg\alpha \\
F(\alpha \mid \beta) : \beta, \neg\alpha \succ \beta, \alpha \\
P(\alpha \mid \beta) : \beta, \alpha \succeq \beta, \neg\alpha
\]

These can form the basis of preference-based reasoning system that can meet at least the following desirable characteristics of a deontic knowledge representation:

**Proposition 4.4.5.** What is obligatory is permitted: $O(\alpha \mid \beta) \rightarrow P(\alpha \mid \beta)$.
Proposition 4.4.6. The impossible and the meaningless are not obligatory: 
\[ \neg O(\alpha \mid \alpha) \text{ and } \neg O(\neg\alpha \mid \alpha) \] are axioms.

Proposition 4.4.7. There are no conflicting obligations. The obligations \( O(\alpha \mid \beta) \) and \( O(\neg\alpha \mid \beta) \) are inconsistent: 
\[ \neg(O(\alpha \mid \beta) \land O(\neg\alpha \mid \beta)) \] is an axiom. Idem for \( O(\alpha \mid \beta) \) and \( P(\neg\alpha \mid \beta) \).

Proposition 4.4.8. If \( \phi \) is the set of worlds \( w \) such that \( M,w \models \phi \), then the sentences \( O(\alpha \mid \top) \), \( O(\beta \mid \alpha) \), \( O(\neg\beta \mid \neg\alpha) \) are only satisfied by the ordering 
\[ \neg\alpha \land \beta \prec \neg\alpha \land \neg\beta \prec \alpha \land \neg\beta \prec \alpha \land \beta \].

One of the attractive features of the representation in the form of preferences is that it produces triangles between a context of applicability \( \beta \), and good (\( \alpha \land \beta \)) and bad (\( \neg\alpha \land \beta \)) alternatives that are complete partitions of the context of applicability. It naturally fits in a graphical representation of taxonomies, and knowledge acquisition methods like the repertory grid that combine development of a taxonomy and sorting along evaluative dimensions (cf. for instance Gaines and Shaw (1993); Ford et al. (1991)).

A point of contention may be the translation of a permission to a statement of weak preference (\( \alpha \gg \neg\alpha \)) instead of indifference (\( \alpha = \neg\alpha \)), as has been argued by for instance Opalek and Wolenski (1991); Royakkers (1997). Indifference is however incompatible with the intuitions of deontic logicians (cf. Alchourrón and Bulygin (1981)); If something is obliged, then it should also be allowed. The asymmetric statement (\( \alpha \gg \neg\alpha \)) leaves room for a prohibition or obligation (\( \alpha \succ \neg\alpha \)) and retains the information that the represented permission explicitly allowed \( \alpha \) and not \( \neg\alpha \), on which it was silent.

It is perhaps extended exposure to alethic modal logics and law that causes the intuitions. There is empirical evidence that children generally attribute an attitude of indifference towards \( \alpha \) and \( \neg\alpha \) to others who express \( P(\alpha \mid \top) \) (cf. Keller et al. (2004)).

A norm system that only contains \( P(\alpha \mid \top) \) is not a well-formed norm system at all since it serves no purpose in guiding and evaluating behaviour. Operator \( P \) serves no real purpose in a deontic reasoning system that does not allow for conflicts between norms in guiding and evaluating behaviour. The operator \( P \) cannot be understood in any other way than a superfluous utterance stating indifference towards \( \alpha \) and \( \neg\alpha \) if it is evaluated without other normative expressions as context. The \( P \) only becomes relevant if:

- it conflicts with an obligation or prohibition existing in the context of discourse, and
- it is used to cancel the conflicting obligation or prohibition.

The explicitly stated permission clearly has another function than the dispositional permission inferred from the absence of knowledge of a prohibition. Since permissions are usually uttered with the explicit intention of amending a specific obligation, and are often found nearby in the same legislative text, it makes sense to ‘localize’ them to some extent by adding the asymmetry. The
asymmetry is also necessary for using the representation in combination with common deontic reasoning systems.

Some broad permissions can perhaps be interpreted as symmetrical. It is for instance generally accepted that freedom of expression i.a. includes a strong permission to keep your opinion to yourself. We might add a fourth dyadic operator for freedom of choice (liberty) with some limited applications in the representation of legislation:

\[ L(\alpha|\beta) : \beta, \alpha = \beta, \neg\alpha \]

Logical theories of subjunctive betterness can be investigated on their own, without any further presumption on how norms work, and a number of theoretical issues in AI & Law can be addressed by considering the norm as a statement of subjunctive betterness alone.

4.5 Agents and Action

Obviously, the function of norms cannot be considered without appealing to the notion of agency. The norm does not merely express relative goodness of certain states of affairs, but also directs agents (not) to bring them about.

Agency is not a subject that should be considered part of AI & Law: most accounts of action and planning are written outside the field of AI & Law, and are often designed with automated planners – that construct a plan to achieve a certain goal state from some initial state for a domain with certain properties – in mind (cf. generally Blythe (1999); Boella and van der Torre (2004); Erol et al. (1994); Bratman (1984); Cohen and Levesque (1990); Allen and Ferguson (1994); Kowalski and Sergot (1986); Allen (1984); Hansson (1994); Pacheco and Carmo (2003); Suchman (1987)). From a Semantic Web perspective it makes sense to combine a generic mechanism for deontic reasoning with a theory of action, events, or plans designed for some other purpose.

These theories can vary considerably in how they deal with time, with change, etc. Still, we cannot satisfy ourselves with an account of deontic reasoning that abstracts the thing being qualified to some opaque proposition. Doing so gives rise to odd analyses of what norms mean for human action.

We have to make at least some assumptions about agency and change, and relate these to normative positions like being in a state of obligation to do something. An important concept in this respect is the distinction in legal theory between ought-to-do and ought-to-be representation, discussed in the next section.

In a nutshell, this chapter tries to account for two major uses of legal rules:

**Planning** to perform a task one set oneself; to bring about beneficial legal facts, while avoiding detrimental ones; and

**Situation and action recognition** to infer one’s own legal position, and what legal facts others bring about, are going to bring about, and unsuccessfully attempted to bring about.
Planning involves generating and comparing alternative plans, while situation and action recognition involve generating and comparing explanations. The planning perspective also prominently involves time and change.

The purpose of this section is to point out some constraints that the conceptualization of intelligent behaviour implicit in the rules themselves impose on the ways in which we can conceptualize agency and change. It is not intended to set a golden standard for the conceptualization of planning and change itself.

4.5.1 To Be or To Do

A central subject in the representation of obligations is the ought-to-be versus ought-to-do debate (cf. Castañeda (1970)): does the obligation prohibit a state of affairs or the bringing about of a certain type of change? What role does intention play in this? Important for knowledge engineering: is there a logical relationship between these two possible formulations that allows us to translate between them? If obligation cannot be reduced to mere subjunctive betterness, as for instance Castañeda (1970); Dayton (1981) argue, we have to account for what an obligation tells us to do.

Valente (1995); Lehmann (2003); Hoekstra and Breuker (2007) however argue for a distinction between:

1. the norm proper, which simply derives \{allowed, disallowed, silent\} from a situation description,
2. the causal connection between the conduct of a person and the qualified situation,
3. the fault implied by the conduct of that person, and
4. the attribution of responsibility, usually but not always to that person\(^{10}\).

The discussion in Hoekstra and Breuker (2007) gives the best overview of the mechanisms involved, and stresses that causation itself is the outcome of ampliative inference explaining the occurrence of an event by a process explained by an earlier or simultaneous event. This causal model is an abstraction of underlying processes. Agent causation simply extends this to mental processes.

Since the end result is an explanation in terms of agent causation, or the lack of it, and the legislation itself can be understood in terms of agents “bringing about” i.e. causing certain occurrences, this mechanism of explanation is of minor concern here: legislation typically addresses the explanation, and not the underlying processes.

There are two markedly different “styles” of legislative drafting that are both encountered in legislation: the productive style, which focuses on unwanted results, and the behavioural style, which focuses on ease of recognition of unwanted behaviour. Note that the productive style does obviously attribute it to behaviour: the ought-to-be style describes the goodness of the product of our

\(^{10}\)For instance liability of a parent for the conduct of a child.
actions. It deals with states of affairs generally caused by humans, but omits specification of their role in it. The ought-to-do style directly describes the action. As pointed out in section 4.1, action recognition presumes recognition of the scripted role the agent assumes and ascription of intent.

Generally speaking, in settings involving interaction between a human and a machine, or a design of one, it is easier to give a productive characterization of what the legislator doesn’t want: e.g. the carrying capacity in kg of the elevator should be at least 100 kg times the number of people you can reasonably fit into the volume of the elevator. It is of course possible to give a behavioural characterization of the same constraints, but this approach can be clumsy and unintuitive.

In these settings we are usually dealing with what is often called a classical environment that is fully observable, deterministic, with a finite number of discrete alternatives at any point, and where the system or design being manipulated is only changed by the planning agent. If the system or design violates a rule, it is its owner, operator, or designer that did it.

In environments where one or more of these features are not present there is a less predictable interaction with the environment. In this kind of setting we have failed attempts, unforeseen products of acts, and ambiguity about who caused something. Because the intention of the law is to a large extent to influence people’s behaviour, it is more practical from a compliance and enforcement perspective to give behavioural characterizations of behaviour and to point directly at the seller, the driver, etc. Behavioural characterizations are generally speaking easier for the addressee to learn and adopt, and violations of norms with behavioural characterizations are often easier to recognize and prosecute.

Because the abstract goals of the legislator often have a productive nature (for instance reducing the number of people that die in traffic, or reducing conflicts over the possession of goods, or raising money for some purpose), behavioural norms often feel like approximations of what the legislator really wants of the addressees of the legislation. Behavioural characterization increases precision in enforcement and compliance, by making the things to which it applies easier to foresee and recognize, but at the expense of precision in describing the policy goals of the legislator.

Productive characterization creates problems for the addressee if the addressee does not foresee the consequences of his actions reliably in an environment that is only partially observable and stochastic. This is for instance often the case if other human beings are involved. If the addressee perceives a lack of control over the situation, enforcement of legal norms has little effect.

The full mechanism in Hoekstra and Breuker (2007) appears to be conceptualized explicitly in the sources of law only in cases of grave importance, for instance responsibility for someone else’s death. Here the legislator not only addresses unintentionally causing death, but also failed attempts at causing it.

In addition to behavioural and productive characterization, the attempt forms a third category: characterization of display of undesirable intent in itself, regardless of the form in which it is executed and whether it is successful. This category is problematic, and therefore usually avoided if possible. There is a
strong taboo on “ruling over the consciences” of citizens, and a practical reason to avoid it is obviously that it requires mind reading on the part of the court.

Still legislation with a strong moral tone, in particular criminal law, does involve a certain degree of “mind reading” when dealing with things like attempts (to kill someone), premeditation, or the distinction between intentional and unintentional killing. Because the court cannot actually read minds, it has to infer apparent intention from actions. To do this, it has to judge what the agent foresaw as the consequences of his actions, and this requires ascribing knowledge to agents. This isn’t really helpful, as it just moves the mind reading problem to another aspect of the operation of minds.

The solution is unsurprisingly found in the ambivalent nature of norms: it is possible to set standards for what reasonable people should know. There is also a related legal concept: the knowledge of the man on the Clapham Omnibus (cf. Montrose (1958)). How to realize this in KBS is hardly a problem: it is usually taken for granted that the dispositional belief set of some agent (cf. section ??) is the logical closure of its individual beliefs with some universal body of shared knowledge. The idea that we share a body of knowledge is an assumption behind ontology modeling. It is indeed a necessity that people who understand each other share a body of knowledge. In addition to shared ontological knowledge, people must also be assumed to share knowledge about what normally happens, as alluded to in section 4.3.2.

The fundamental difference between an ought-to-do characterization and an ought-to-be characterization is in the omission of an agent identified by role (e.g. the driver coming from the right, the thief who is taking a good that wholly or partially belongs to another) in the description. The problem with the ought-to-be characterization is therefore in identifying the agent who caused the state of affairs or failed to prevent it from occurring, while in ought-to-do characterizations this is merely a matter of action recognition. While it is the state of affairs that is considered to be disallowed, it is an agent who will be, for his actions or his failure to act, held responsible for its occurrence. For the agent this means that when he foresees that his action may end up in an prohibited state of affairs, the problem is whether responsibility for the state of affairs will be attributed to him or to someone else.

This is not trivial. It is not necessarily the case that the one held responsible for an undesirable situation should also be the proximate cause of the situation in which the violation arises. You can’t for instance simply reduce ought-to-do to ought-to-be. An example: Imagine someone offers you a painting for sale, with the condition that the offer is valid until next tuesday. You have a claim-right to buy the painting, in the sense that the counterparty has a duty, towards you, to sell you the painting if you accept the offer before next tuesday. The counterparty then sells the painting to someone else. After he sold it, you accept the sale. You are now in the situation prohibited by the obligation, and you are the proximate cause of the situation. Still you are not responsible, because the duty wasn’t yours but the seller’s: the fault is with the seller.

Proposition 4.5.1. The mere fact that a situation is prohibited does not nec-
essarily mean that a plan that brings it about is a bad choice for a specific agent. It depends on whether you will be held responsible for it. If however the agent conceives of the plan as an alternative in a planning problem, and the plan is based on the assumption that other agents will behave normally, i.e. following the prevailing norms, then it is very likely that choosing to execute that plan is a bad choice.

Even if we consider the attribution of responsibility to agents as unproblematic, and subscribe to a simple deterministic model of states and transitions, the relation between behaviour and product is not immediately obvious. d’Altan et al. (viz. d’Altan et al. (1996)) for instance offer the following hypotheses for the relation between ought-to-do characterizations and ought-to-be characterizations:

1. A state is obligatory if and only if it is the result of an obligatory action.
2. A state is obligatory if and only if all the actions that lead to the state are obligatory.
3. A state is obligatory if and only if it is forbidden to undo it.
4. A state is obligatory if and only if all the actions that are necessary to bring about the state are obligatory.

None of these alternatives is without counter examples.

For the representation of legislation, in particular the kind of legislation that lends itself to routine decision making processes amenable to automatization by KBS, it is safe to assume that it can be understood in terms of agents “bringing about”, i.e causing, certain occurrences, with or without execution of a plan to that effect. The subject of the norm is therefore naturally the action.

**Proposition 4.5.2.** The subject of a normative statement is an action.

One can however also be held responsible for one’s non-action. Of this scenario we know two variants:

1. One has an obligation to act and doesn’t; and
2. something that is disallowed happened because one did not act to prevent it (negative causation, cf. Lehmann (2003); Hoekstra and Breuker (2007)).

The first case is a standard feature of any combination one makes of a deontic logic and causation.

The second case only happens if one is presumed to have been aware of the possible undesirable occurrence and didn’t attempt to prevent it, i.e. demonstrated a lack of intention to prevent the occurrence through one’s choices. Assuming that agents are always executing some task they set themselves, the

\[11\] I.e. performance of the actions necessarily leads to the state, and non-performance of the actions necessarily prevents the state.
undesirable occurrence was “on the radar” during the planning of that task and therefore part of the conceptualization of the situation for which one made the plan: here too one is in the end responsible for what one did do instead of what one ought to have done.

4.5.2 Agent Causation and Time

By acting, an agent either initiates or terminates certain fluents, for instance a:

1. state of an object; for instance the state of being married, or the state of obligation of the counterparty to deliver certain goods,

2. situation, i.e. a configuration of objects; for instance two people being married to each other, or

3. perdurant object; for instance a marriage, or a private company limited by guarantee.

Sartor in Sartor (2006a) (table 2) lists a number of normative conditional patterns based on the distinction between initiation (action initiates fluent), termination (action terminates fluent), and emergence (fluent entails fluent) where the consequent is always a legal fact. These facts are further classified into normative positions (e.g. my obligation to terminate the traffic violation initiated by the traffic participant overtaking me), qualifications (e.g. being a married person), and the existence of legal things constituted by something else (e.g. the marriage that exists between me and my wife).

All of these are obviously constitutive rules. The difference between qualification (married) and existence (a marriage) in this sense is primarily a difference of descriptive vocabulary and not of substance: reification of relations in a knowledge representation and nominalization of verbs or adjectives in natural language are motivated by technical and presentation considerations, not ontological ones.

Initiation and termination are generally encountered in real life as pairs – a change operating on the same substrate, which can be an object or some configuration of objects (a situation, scenario, plan, system, etc. The change terminates a state and initiates a state of the same thing. State (for “stative”) is taken to be the generic type subsuming the previous categorization of ways in which fluents are described.

Observe that the combination of the deontic operator ‘it is obligatory that’ ($O$) from section 4.4 with a simple intentional action operator ‘agent $i$ brings about’ ($E_i$) would give rise to four atomic types of obligations (cf. generally Sergot and Richards (2001a) for a discussion):

1. it is obligatory that $i$ brings about $C$;

2. it is obligatory that $i$ brings about not $C$;

3. it is obligatory that $i$ does not bring about $C$; and
4. it is obligatory that \( i \) does not bring about not \( C \).

Let’s for the moment set aside the problem of how \( i \) is going to (know whether \( i \) can) bring about \( C \), and how one determines whether \( i \) did, didn’t, attempted it, etc. The difference between ‘\( i \) brings about not \( C \)’ and ‘\( i \) does not bring about \( C \)’ requires some further analysis.

Obviously bringing about not \( C \) is only meaningful if \( C \) is the case: we can reformulate this to terminating \( C \) (whenever it may arise), while not bringing about \( C \) is about not initiating \( C \), and \( C \) is a state, some occurrence that occupies a time interval or period on the canvas of time, that is initiated by some change and terminated by another change. Similarly, we can talk about not terminating \( C \) (whenever it may arise), and about initiating \( C \).

The agent can be held responsible for the fluents he initiates and terminates: this is the easy case. He is also responsible for the fluents he was able to initiate or terminate but didn’t. In the Netherlands it is for instance permitted for a pair of bicyclists to ride next to each other. If a third traffic participant overtakes them, this traffic participant initiates a traffic violation, against the obligation to keep to the right. At the same time jurisprudence shows that any one of the three traffic participants can be held responsible for the traffic violation, since each of them is able to terminate the traffic violation. Current jurisprudence apparently interprets the obligation as pertaining both to initiation and to termination, but this does not follow from the phrasing of the involved obligation and permission which simply addresses the stative pair of bicyclists riding next to each other (in other words, an “ought-to-be” characterization).

This should be kept in mind when interpreting ought-to-be descriptions: does it pertain to initiation, to termination, or to both.

Normative rules are usually about bringing some change about. There is a general pattern explaining how legislators will typically phrase normative rules:

\[ O : \text{some change} \succ \text{the continuation of the sketched situation}; \]

\[ F : \text{continuation of the sketched situation} \succ \text{some change}; \text{and (less markedly)} \]

\[ P : \text{some change} \succeq \text{continuation of the sketched situation}. \]

Norms are only concerned with initiating and terminating \( C \), and do not have any bearing on any actions that neither initiate nor terminate \( C \). None of these formulations tells me anything about bringing about unrelated fluent \( D \) acting on another substrate.

Only in the case of directives to not initiate or terminate \( C \) we can determine directly whether a possible violation has taken place: it takes place if we do initiate/terminate \( C \), i.e. when a change occurs. Directives to initiate or terminate \( C \) are however inevitably deadlines, and only the passing of the deadline gives rise to violation: the obligation to initiate \( C \) may well be irreconcilable with a course of action that consists of first initiating \( D \) and only then \( C \), but this has little to do with the position of the initiation of \( D \) relative to the initiation of \( C \) in time.
The implicit deadline leads to a bit of a problem if we consider this in the context of things allowed and disallowed by a norm. Take as as example the obligation that \( i \) terminates \( C \); This means the following in terms of normative qualifications:

1. \( i \) terminating \( C \) is allowed;
2. the continuation of \( C \) by \( i \) (beyond some unspecified deadline) is disallowed.

The representation of case 2 is obviously a bit problematic, because it addresses the non-action of ‘doing nothing about \( C \)’. Exact identification of the intended logical complement of terminating \( C \) is very hard.

Let it for instance be prohibited to drive with a vehicle with broken headlights. As usual, I discover a headlight is broken while on the road when dusk sets in. So I am obliged to leave the road to repair my headlights first, which is going to involve some more driving on the road. Since no deadline conditions are given, intuitively, only action betraying the intention to continue the violation, would clearly constitute a violation. Whether I get fined for it depends on enforcement policy and common sense.

**Proposition 4.5.3.** It is not possible to determine whether an obligation to bring about a change has been violated, unless one sets a deadline.

In a logic of agency which treats actions as instantaneous changes of state occurring at discrete time steps (i.e. Hoare logic, dynamic modal logic), it is natural to consider the next state as the implicit deadline. Combined with the presumption that, given full knowledge of the initial state and a known menu of possible actions, the state following execution of an action should be predictable with absolute certainty, we can axiomatize obligation completely in terms of agency, like the proposals in d’Altan et al. (1996). We should keep in mind, however, that the verity of these axioms depends on the conceptualization of the planning domain, and not on the properties of the concept obligation: it doesn’t shed any light on the nature of obligation.

Representing a normative rule involves some common sense judgment on the part of the knowledge engineer. How complicated a representation needs to be depends i.a. on the following:

1. Does the normative rule demand that one displays the intent to comply, i.e. does it take into account the possibility of failure of attempts?
2. Does the rule require continuation or change?
3. If it describes a change, is it phrased as an initiation or a termination?
4. If it requires a change, is there an indication of a deadline or a before condition?
We can identify a number of ingredients that should be covered, although we cannot solve the deadline problem. For the representation of normative rules, apply the following principles:

1. Normative rules apply to action;

2. Action causes changes;

3. Action, whether conceived of from the perspective of the actor or recognized by a bystander, is situated, i.e. takes place in a situation; and

4. A situation consists of a limited number of participants, it is not a (state of the) world in the logical sense (cf. section ??), but a conceptualization of the context in which the action takes place.

The direct connection of situation to action is inspired by Suchman (1987); Barwise and Perry (1983), and the representation of discourse context in general in computational linguistics (for instance Piwek (1998)). It deviates significantly from AI planning literature, which is largely based on the assumption that 1) world and situation are the same thing, and 2) that situation + action results in a new “situation”. This works well for planning, but is not terribly realistic.

Instead an agent reconceptualizes the situation in the context of planning the next action: since this is done in the knowledge of one’s previous action, and one may be acting still in the same role, it is not surprising if the new conceptualization borrows participants from the old one. Since we have no particular application in mind, it is wise to take the conceptualization in the source of law at face value.

Actions are situated, performed by an actor, and cause changes. As always, this is the simplest possible conceptualization, ignoring issues like decomposition of both actions and situations.

**Example** Consider the obligation to repair a broken headlight. Disallowed is any action performed in a situation which involves a broken headlight and does not involve fixing it. To find an acceptable resolution to deadline issues we would have to address decomposition, for instance:

1. Disallowed is any action performed in a situation which involves a broken headlight and does not involve fixing it, and is not part of an action that involves fixing it.

2. Disallowed is any action performed in a situation which involves a broken headlight and does not involve fixing it, and is not an execution of a task that involves fixing it, or of a subtask of a task that involves fixing it.

These kinds of refinement are either based in common sense or in jurisprudence. The scenario of negative causation, although superficially similar, is different. In this case a *future* change is implied by one’s conceptualization of the situation.

It may seem strange to include expectations about the future as part of the situation one acts in, certainly if one conceives of situations as is usually done in AI planning, but it is no stranger than including expectations about the past: if we see someone coming towards us with a blood-covered chainsaw, our
explanation of this event will cover both the past (to explain where the blood comes from) and the immediate future\textsuperscript{12}. The notion of normative position, which can also be conceived of as being part of the situation in which one acts, is also often essentially about future possible consequences of one’s actions.

4.5.3 Positions and Power

Interesting are the positional uses of obligation, called normative positions. The most obvious of these is the state violation possibly initiated by an action that is disallowed.

Other familiar ones are systematically related to corresponding normative rules, but are secondary to the reading of normative rules as being constitutive of the qualifications \{allowed, disallowed\}. Given the existence of a norm “given \(\alpha, \beta\) it ought to be that \(\gamma\) is initiated” and me being in a context that entails \(\alpha\), I have a contextualized obligation to initiate \(\gamma\) if \(\beta\).

The relation between this type of concept and the original norm is similar to “being in a position in which I have to get ingredient \(a\)” when I am preparing a dish that requires ingredients \(a, b, c\) and I have only \(b\) and \(c\). If I believe I cannot obtain ingredient \(a\), my position would be “not being able to make the dish”. This is a planning concept (and a fluent that exists only in my mind, or that of others) that has no direct bearing on the recipe.

The same applies to norms: the contextualized obligation is not the original norm, but a state relative to the norm that only acquires special meaning by adding a theory of what I am able to bring about and what I am trying to do. It is possible but not very useful to automatically derive positional obligations. In scripts, as institutional statives, they have their uses as a shorthand for a compound action.

While normative rules apply to action, positional concepts also often follow the state constitutes state pattern.

Positional concepts play a role mainly as a description of the result of the application of a constitutive rule. They also occur as conditions to constitutive and institutional rules. A special type of condition is the (legal) power or competence. In common law jurisdictions the term power appears to be preferred (cf. Bentham, Hohfeld, and Hart), while civil law jurisdictions commonly call it competence (cf. Bulygin (1992)).

The power to bring certain things about refers to a certain legal qualification in the constituting base that applies to the agent that gives the ability to bring about a legal fact. “Only parliament declares war” for instance means that being the parliament is a necessary condition for the initiation of the state of war. It is distinguished from other similar potentialities (“Only people who have money can buy stuff”) by the fact that the qualification is a fact within the institution.

\textsuperscript{12}The explanatory power of a scientific theory of the past, like evolution theory, is for instance in its prediction of future observations about the past, i.e. missing links yet to be found.
The agent is able, competent, or it is within his power, or he has the potential, to bring certain things about. Almost every agent is for instance able to open and close an unlocked door, but most agents are not able to open a locked one. This is ability, and was already indirectly addressed in section 4.3.2: to have the ability to do something in a certain situation is to meet the conditions for filling an agent role. Some agents are able to marry a pair of other agents, but most are not. Since marriage is an institutional status (nowadays), this is (legal) power or competence.

The abilities and powers of a person are limited to the agent roles for which he meets the necessary conditions in the situation. Agent roles are stative, and in that sense part of the situation in which an action is performed: it is however common to distinguish states that specifically pertain to the agent (e.g. police officer) from those that do not (e.g. dark), which are rather conceived of as conditions for executing the task, and relatively shortlived ones (e.g. unarmed) from longlived ones (e.g. police officer).

Some authors have tried to explain power-conferring rules as a variety of normative rules (notably Von Wright, Bentham, Cornides, Kanger, Lindahl, and Kelsen), while others explain them, as we do here, in terms of constitutive rules constraining institutional action (Hart, Ross, Searle, and Bulygin in Bulygin (1992), which is also the source used here for the classification of authors by the position they take on this subject).

As pointed out in 4.2.2, we usually speak of a power if the legislator intended to create a way for a certain class of agents to bring about the legal fact. Attribution of power is captured by standard constitutive rules, if these have legal conditions that pertain to the agent role.

### 4.6 Interaction Between Agents

The mere publication of a formal representation of set of constitutive rules does not make a functioning legal system. Implicit in the act of legislating is the implied threat to react if violations take place. The Functional Ontology of Law in Valente (1995) therefore posits a special category of legal knowledge that covers this function: reactive knowledge. The institution however does not depend on an ability to “punish” itself: it reacts to violations by either obliging, permitting, or empowering others to react. Essential to the functioning of the institution is that there is a critical mass of agents who are willing to follow these directives or use these means provided to punish others.

Tort law for instance depends on payment of damages to function as a sanction of the injuring party and an incentive to report the violation for the injured party, and police officers follow directives because they are paid to do so with taxes collected (using the law) i.a. for the benefit of having police officers who react to violation of criminal law directives.

**Example** The following is for instance a somewhat simplistic constraint on the power to arrest: An arrest involves a violation and a police officer.
The institution works if it is possible to arrange normative order in such a way that reacting to one agent’s violation of the law is in the interest of another. Very often a role is given to the “victim” of the violation, but it is important to note that no appeal to restorative justice (i.e. the notion of norm violation as an act against another individual, who should receive some kind of restitution) is needed to explain why the victim usually takes on this role. Very successful and efficient (but morally unappealing) enforcement systems have for instance been based on the principle that the agent who brings the case (and often also the suspect) in for adjudication receives property of the agent violating the law after he is executed or enslaved. The legislator counts on self-interest and commitment to the public interest.

Reaction of course subsumes not just punishment, but also the mere reporting of the case, gathering evidence, optionally apprehending suspect(s)\textsuperscript{13}, adjudication, and any other supporting activities. It is customary, for obvious reasons, to separate the role of adjudicator (who authoritively interprets the law and applies it to the case) from the reaction per se.

As pointed out before the whole system depends on nothing but constitutive rules, but these rules are arranged in such a way that the addressed population of agents makes enforcement work by organizing punishment of each other, and collecting the necessary funds for employing professionals to make the system run smoothly.

A simple example of this mechanism can also be seen in international paralegal frameworks like ship classification societies: ship owners voluntarily pay a subscription fee to a classification society, which in turn uses the collected funds to draft rules and send surveyors to ships to enforce compliance with these rules. The rarely used sanction consists of suspension or loss of “class”, which is a problem for the ship owner because port authorities generally demand classification with a reputable classification society. Authoritative interpretation of the rules in final instance is left to some court.

Valente (1995) is obviously right in noting that there is a functional distinction between norms and other rules that regulate normal interactions and those that regulate reaction sanctioned by the legal system. Contrary to some of the other distinctions made there (for instance between world knowledge and normative knowledge), reactive rules are however in no way distinguishable from other categories of rules by their content. It is their intended function in decision making which identifies them. Another functional category in Valente (1995) to which this observation applies is metalegal knowledge, which contains the rules of adjudication.

**Proposition 4.6.1.** _The judge, police officer, parliament, etc. just follow the rules of the institution. There is no distinguishing formal criterium that sets apart reactive rules from other rules. The distinction is purely functional, and often not easy to make._

\textsuperscript{13}This step only becomes important if the foreseeable punishment justifies running away.
4.6.1 Transactions and Interests

The previous section strongly suggests that the interests of others play a large role in deciding what you can do, and that the law is usually set up in such a way that your violation of an obligation may give rise to another person’s permission or power to do something to you which is in their, and contrary to your, interest. An example: your obligation as a driver to yield at intersections for drivers from the right, is correlative to their right of way. This may mean two things:

1. You violating against the obligation may be a precondition for some action the other driver can take against you. Typically, the violation of a traffic rule would make you for instance liable to paying for the damage if an accident happens. This is a specific position which is associated to him being the victim of your action, and it is in his interests to use this position.

2. More important in daily life, the other driver may proceed with the legitimate expectation that you will yield. The other driver will typically prefer driving on over stopping, and will expect you to stop in order to comply with your obligation.

These two options need not be correlated although they often are. A police officer can react to the traffic violation by fining you, by virtue of being a police officer, without being the victim of your action. The victim of your action is not in a legal position to react to you if no accident happens.

Most normative positions are two- or more-sided in the sense that there is or are possible victims of your noncompliance. But how do we decide that some bystander is a victim of your noncompliance, or a beneficiary of your compliance?

A number of implicit interests are involved in the example, that can be represented in analogy with the subjunctive betterness interpretation of norms:

1. The other driver usually prefers not stopping over stopping, and therefore you behaving normally over you not behaving normally.

2. Both drivers prefer no accident over an accident.

3. The other driver usually prefers you paying for the damage over you not paying for the damage in the case of an accident.

4. You prefer not paying for damage over paying for damage.

The legislator is making certain assumptions about the preferences of typical bystanders depending on their agent role. In legislation these usually remain implicit.

Imagine an agent who is solving decision problems all the time in a situation involving other agents who might do unpredictable things based on their own decisions. In his decisions he has to take into account his legal position. In the first place, this means that there are certain things that he cannot efficiently
bring about, because they are disallowed. But since these norms also apply to the other agents, there are also certain things he can efficiently bring about by taking into account their obligations, and assuming they behave normally. Crossing intersections while somebody else is coming from the left for instance, in the expectation that they will yield, or sending someone the product they ordered, in the expectation that they will make a payment upon receiving it.

**Proposition 4.6.2.** *The law does not only constrain action: It also creates abilities.*

The true real life beneficiary of an obligation is the one who has a preference for someone else meeting their obligations. The other driver on the intersection is therefore typically the beneficiary of the obligation to yield. This is to be distinguished from another function of explicit assignment of the role of beneficiary by the legislator: In traffic rules the other driver will not be explicitly recognized as the holder of right correlative to obligation, and in criminal law somebody may be explicitly recognized as beneficiary without actually having an interest in it: think for instance of euthanasia as murder. In many cases this fiction is used to attribute potestative right to the beneficiary to take some action following noncompliance, while in other cases – like violent crime – the notion of a victim is purely fictional, and intended only for the justification of the law itself.

**Proposition 4.6.3.** *The legal institution ascribes preferences to persons depending on the agent role they adopt.*

As pointed out in section 4.2.2 and Sartor (2006a) in general terms, and explained in detail in Sartor (2002), there is firstly a difference between things one does in one’s own interest, and things one does in the public or communal interest, and secondly a difference between powers one can exercise in one’s own interest and powers one can only exercise in the public interest.

This discussion of obligation and correlative right extends to Hohfeld’s jural relationships, of which duty-right is just one form. Hohfeld’s jural relationships can only be understood be appealing to the preferences and intentions that typically come with adopting a certain agent role.

### 4.6.2 Hohfeld’s Conception of Rights and Powers

The principal aim of Hohfeld’s work (in Hohfeld (1919)) was to clarify *juridical relationships* between parties. Hohfeld presents us with an analytical scheme which distinguishes four different categories of jural relationships between parties and makes a number of analytical distinctions between various legal positions. 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. In addition, each fundamental conception is a jural opposite to another: privilege, right, power, and immunity are the respective jural opposites of duty, no-right, liability, and disability.
Hohfeld’s system allows for a more fine-grained distinction between patterns we observe in legislation, and generally turns out to be more useful, in particular in private law, for recognizing and classifying patterns in legislative language than the basic deontic categories and constitutive rules in general. It is useful in this context because it covers nearly the whole terrain covered in this chapter, and has been used before in the legal knowledge engineering field (for instance Allen (1997)).

Hohfeld conceptualization is not without its critics, however. Hohfeld’s names are sometimes slightly confusing. The name privilege for instance suggests more than it actually means. The privilege can also be called plainly permission, right, or liberty. Also right has at times been renamed to claim or claim-right to distinguish it from other uses of the term right. Since section 4.5.3 distinguishes powers or competences from abilities, disability is also an unfortunate name. Ability/disability for “brute” ability, and competence/in-competence (or power/impotence) for institutional ability make more sense in the context of this book.

Also liability is likely to confuse people and has been criticized. It is called subjection in Sartor (2006a). Hohfeld’s liability means that you are exposed to the exercise of a power: you (have to) accept exercise of the power to change legal reality. In private law we encounter a similar concept of the same name, but with a more specific meaning. But rejection of a name like liability for this reason makes it impossible to develop a jurisdiction-independent vocabulary for legal reasoning.

Another problem with Hohfeld’s categories is according to some (cf. Halpin (2003)) that they fail Occam’s test and are therefore not fundamental: there are multiple ways in which his 8 fundamental conceptions can be reduced to eachother. This however also applies to the three deontic categories. Knowledge engineers should certainly be willing to take advantage of attempts to interrelate a number of wellknown concepts from the field of law. There is no need for them to be fundamental.

Another problem is with the notion of opposite. The opposite pairs seem to have subtly different meanings, which are not quite captured by simply negating the opposite. This is yet another indication that these patterns are hardly fundamental legal patterns.

Hohfeld’s relationships distinguish between normative positions and other non-normative legal positions, between the competence and incompetence to play a certain agent role, and therefore to cause a certain change of position, and between the obligation to cause a certain change of position or the absence of such an obligation, and most importantly, between the one who acts and the one who predicts the actions of another.

Although we can in principle apply this set of distinctions to any pair of agents, Hohfeld’s presumption is that the agent observing and predicting the actions of the other has an interest in what the other does, and ascribes interests to the other. In essence we are dealing with the ability of one agent to infer 1) that another agent has the ability or inability to change his situation in relevant ways and 2) that the other agent has an interest in changing or not changing it.
Hohfeld’s concepts of right, duty, privilege, and no-right relate to the interests someone has in changing the situation, presuming they behave normally, and the concepts of power, liability, disability, and immunity relate to the ability to change the situation.

The explanation again has to appeal to scripts (cf. section 4.3): an earlier example given of a “primitive plan” is buying a quart of milk. We realize that the execution of this script depends on someone else’s willingness to execute “selling a quart of milk” part, but we still tend to believe that this will be achievable without problems, even for a known reference price. The most obvious reason that we are actively monitoring and predicting someone else’s actions that change our situation is because we are executing a script that involves that other person as an actor.

Proposition 4.6.4. Jural relationships only exist between pairs of agents involved in the execution of the same script.

Example This has to be made concrete with an example. The initiative to act has to come from both parties, and both parties recognize the same relevant participants of the situation. As an example we will use a sales transaction initiated by an irrevocable offer to sell by y. Person y offers for sale to x a painting for the price of $500, to be accepted before next tuesday. Person y now has a duty to sell, under the stated conditions. Person x therefore has a right to buy under the stated conditions, by accepting the offer. Concluding the sale leads to a duty to pay of the seller towards the buyer, and a duty to deliver on the part of the seller towards the buyer. We can abstract away the amount (for the price of $500), the object to be sold, and the timeframe since these do not add to the legal complexity of the case. A not time-limited offer to sell for an unspecified price would however be more difficult to handle.

From the perspective of the buyer we are dealing with a buy action, and from the perspective of the seller with a sell action.

With this scenario in mind we run through Hohfeld’s relations.

Proposition 4.6.5. A right of x towards y wrt. z is equivalent to a duty of y towards x wrt. z.

The duty is obviously very similar to the obligation of deontic logic. Party Seller sets the script into motion by an OfferForSale which is recognized by Buyer, creating an institutional state, or position, OfferedForSale.

If Buyer proceeds with AcceptOffer, which requires situation OfferedForSale, he creates for himself, besides the institutional state OfferAccepted, an obligation to pay represented as in section 4.5.3, which is also Seller’s right to receive payment from Buyer. Moreover situation OfferedForSale and OfferAccepted also creates Seller’s duty to supply the sold item, which is also Buyer’s right to the same.

Informally we could also say that the acceptance creates Seller’s duty to “sell the painting”, and it is not uncommon to hear it phrased in this way. This obligation is violated if Seller does not sell the painting. Note however that this duty is also violated if Buyer does not complete his part of the transaction, which is giving the promised amount of money. Buyer may for instance find out that he has less money than he thought and try to back out of the sale. Seller can however also refuse to take money, and Buyer can refuse to take the
painting: the obligations do only apply to the part of the transaction which the party is able to perform.

This kind of statement therefore means in effect that $x$ has a duty to $y$ to perform $z$ with the assistance of $y$ only if and as long as $y$ prefers $z$ to happen. In practice this rarely leads to problems if we are dealing with the kind of obligations that only confer a potestative right (usually to go to civil court) on the other party. If a third party (e.g. a police officer) would have such a potestative right, the result would be bizarre.

If you have been offered the painting, and the offer is open to Tuesday, you can say now that you will not buy the painting and by saying so relinquish your right to buy it. In general this is however not the case. This power cannot be presumed.

**Proposition 4.6.6.** A privilege of $x$ towards $y$ wrt. $z$ implies no right of $y$ towards $x$ wrt. $z$.

Privilege suggests an exceptional situation. Privilege suggests permission since its jural opposite is the duty. If, for instance, the *Buyer* in the script is a minor, he has no duty to complete the sales transaction. As already noted in section 4.4, the explicit permission functions as an exception to a more general obligation.

Hohfeld considered the no-right as opposite of the right. We see no benefit in introducing a category for the mere purpose of completing the square. A no-right is simply the denial, the negation, of a right. Person $y$ does not have a right towards $x$ with respect to $z$, either because of the presence of a privilege to that effect or simply because of the absence of a duty to that effect. The Seller for instance has no right to sell the painting before his offer for sale is accepted by anyone.

Readers might object that the sale is a defined legal act, and that it also takes place if *Buyer* or *Seller* fails to complete his part of the bargain. In other words, as soon as *Buyer* has accepted the sale takes place. There is no way to get out of the sale. Running away with the painting does not change the fact that the sale did take place: it simply becomes disappropriation of the painting.

The second square deals with this distinction. It is analogous to the first one, but now we distinguish an institutional law:Sale that took place.

**Proposition 4.6.7.** A power of $x$ towards $y$ wrt. $z$ is equivalent to a liability of $y$ towards $x$ wrt. $z$.

In short, a power is one’s ability to alter legal relations. *Buyer* has the power to enter into a law:Sale with *Seller* of the painting when he has offered it for sale. Thus, *Buyer* has the power to bind *Seller, and* himself, to the legal consequences of the law:Sale. *Seller*, thus, has a liability, which is correlative to power, in that he is liable to having his legal relations altered.

**Proposition 4.6.8.** An immunity of $x$ towards $y$ wrt. $z$ implies a disability of $y$ towards $x$ wrt. $z$.
If $x$ has an immunity against $y$ with respect to the sale of the painting, it means that $y$ has no power to change $x$'s legal position with respect to the sale of the painting. Contrary to the permission the immunity is not generally understood as coming in a weak – implicit – and strong – explicit – variety. Immunity is always stated explicitly. The legislator may have stated that minors are immune to sales transactions. If Buyer is a minor, then accepting the offer does not change the legal position to law:Sale. A disability to bring about a legal effect may be because the counterparty is immune to it, but more commonly because some other necessary ingredient is missing.

While liability and right are simply other perspectives on power and duty, privilege and immunity are simple one of the possible reasons for no right or no ability.

Power and immunity are central to regulating the relations between state and citizen. For instance, if the constitution states that the state has no power to place me under a duty to ask prior permission for expressing my opinions in writing, then I have an immunity and the state a disability. In common parlance this is also a right. So right in common parlance may translate, depending on context, to a:

1. simple permission;
2. immunity;
3. potestative right, or;
4. right correlative to a duty.

4.6.3 Everyone and Delegation

Sofar we dealt with cases between a clear party $x$ and $y$. As a general rule, the parties will have a right to go to a civil court in the illegal situations and the court will search a remedy or reparation to be made by the party responsible for the violation to the party wronged by the violation.

Most norms do not clearly identify both sides of a transaction, however. When $x$ runs a red light while having a duty not to do so, for instance, $x$ wrongs everyone else present in that traffic situation. Other traffic participants have an abstract right towards $x$ with respect to stopping for the red light, but will generally only be in the position to go to court to find a remedy or reparation if there is a concrete wrong. The right correlative to duty does not imply any potestative right. Still $x$ wronged “everyone” in the abstract, and therefore the state exercises the reaction to the violation of a duty on behalf of the community as a whole. This “delegated” power is usually exercised in criminal law. Person $x$ will be fined by the state if caught running a red light.

Hohfeld’s analysis has often been criticized for not dealing with these problems in identifying the involved parties, and sometimes reference is made to “universally quantified” others (cf. Hohfeld (1919)). A categorical distinction between natural persons, private and public legal personalities, the abstract
state in general, and "everyone" or the community seems a more promising approach. The problem is one of identifying the various implicit delegations of the right to search remedy or reparation from (for instance deceased) natural persons and from "everyone" to the state and its constituent public legal personalities. Since modern democracies are based on a conceptual model of delegation of power by "everyone" to parliament, we can safely assume that government has the power to delegate these rights to its own parts.

Proposition 4.6.9. There are collective entities that can play the role of agent in the delegation of powers, but have no other ability besides that. Entities of this type for instance occur in constitutional law.

Delegation of powers is quite central to how administrative law conceives of the legal position of the state and its parts in many legal systems, and therefore also to the legislative process itself. It is considered a central feature of the conceptual organization of legislation.
Chapter 5

Managing a Corpus of Legislation

5.1 Introduction

Managing the corpus of legislation is an important supportive task in the Agile methodology, even if it is a side-issue for agility. Managing the corpus means to come to an interpretation of which rules it contains, after one has resolved all factors determining their applicability, and to deal with changes to the corpus. Most information in this chapter is based on Boer (2009a); Boer et al. (2010).

To legislate is to perform formal, legal actions: the legislator, by legislating, represents an institutional fact with the intent of creating that institutional fact. The source of law represents the rules the legislator creates, but is separate from them: the source of law will still represent the rule when the intention of the legislator to have the rule exist in institutional reality has already disappeared.

The institutional facts that the legislator creates are institutional rules, constitutive rules, and institutional facts required for the correct functioning of these rules. The ulterior function of these rules is to formalize a normative order intended by the legislator.

The legal rules represented by the source of law appeal to two separate realities – institutional reality and brute reality – and 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 recognition of the rules, because people act as if the postulated effects exist.

Through institutional rules, and indirectly through the things it assumes in the constitutive ones, the source of law represents an institutional ontology that describes an institutional reality: it maps out a logical space of possible models of the institution. These rules can be interpreted as terminological axioms.

The legislator makes assumptions about the structure of brute reality, in
particular concerning the behaviour of agents. Plans, intentions, choices, preferences, and abilities play a central role in the way legislators structure reality. The source of law does not, normally, posit terminological axioms about the relevant brute reality, as section 4.2.3 convincingly showed: it merely points to some legally relevant terms, and assumes that the users will be able to make sense of them.

Since the quality of the mapping between brute and institutional realities is likely to be imperfect, we should assume that institutional facts only exist as long as it is consistent to believe they exist. This is however based on an assumption, being that our beliefs about brute reality are generally speaking deeper entrenched than our beliefs about institutional reality: if they are in conflict, it is institutional reality that should give way.

5.1.1 Auxiliary Rules and Facts

Legislators and users of legislation have developed a number of strategies over the ages to deal with the ever increasing complexity of the mapping into institutional reality. These strategies can be classified as design strategies or interpretation strategies (cf. generally Suber (1990)).

Design strategies help resolve potential problems in the use of rules that were already anticipated by the legislator. The *lex superior derogat legi inferiori* (higher law overrides lower law) principle is an important example of a strategy that must be taken into account during the design of legal institutions. When the user of a source of law is confronted with an apparent non sequitur, he can use solutions designed into the law to resolve it. The *lex superior* principle will only be applicable if a hierarchy is built into the system.

Design strategies typically deal with the existence of multiple legislators, even ones who derive their legislative power from delegation by another legislator, that may contradict each other. In addition they deal with the fact that legislators regularly change the rules. In essence, these strategies help distinguish legal institutions and their respective institutional realities, and prioritize between the actions of different legislators operating within the same institutional reality.

Interpretation strategies on the contrary can also used to resolve arising ambiguities that were clearly not anticipated by the legislator. The *lex specialis derogat legi generali* (specific law overrides general law) and *lex posterior derogat legi priori* (newer law overrides older law) are typical examples of strategies that appear to spontaneously arise from use of sources of law.

The legislator can choose to organize his rules in accordance with the intended application of these principles if potential ambiguity is anticipated, but these strategies are applied even if we do not believe the result reflects the intentions of the legislator. As such their status as “legal principles” is questionable: one could take the point of view that these strategies merely reflect deeply entrenched mental habits in understanding messages that are generally respected by the legislator for pragmatic reasons.
Design considerations motivate the legislator to add various *auxiliary clauses* to legislation\(^1\). Some of these are rules, others are simple formal constitutive acts postulating an institutional fact or the occurrence of an institutional event that takes its significance from the rules of the institution, for instance *this statute is repealed on January 1st 2009*.

Auxiliary clauses shape and regulate the domain of legislating itself. In the work on MetaLex, discussed in section 5.3 (cf. Boer et al. (2007b,a)), they are of central importance: relevant metadata is usually found in auxiliary clauses.

Knowledge representation of the source of law can be considered a simple matter of translation of one language into the other, but the relation between the source of law and institutional reality is more tenuous: the clauses in the source of law may be defeated by others, and the sources of law reflect part of the history of an institution, from which its current state can be reconstructed, rather than an institution as such. The relation between sources of law and their knowledge representations is introduced in section 5.2, and sources of law themselves are discussed in more detail in section 5.3.

### 5.1.2 MetaLex

Over the last decade, legislators have begun to adopt XML standards for the formal sources of law they manage, and there is even some activity to standardize on a supranational level. Since these legislator’s standards however generally speaking have an institutional status, coordination between countries requires cooperation between governments, and this process moves too slowly from a consumers point of view, and for reasons largely irrelevant to the consumer.

In this report we use concepts from two XML standard proposals dealing with two complementary aspects of electronic legislation – the documents themselves as a carrier, and an institutional reality they represent: MetaLex XML and the Legal Knowledge Interchange format (LKIF), which standardized legal knowledge representation.

MetaLex XML Boer et al. (2010) is well on its way to becoming formal and de facto standard for legislation in XML. An exposition of what MetaLex is and does in relation to legal knowledge representation therefore suffices for managing sources of law. A description of MetaLex in found in section 5.3. MetaLex metadata is available for the Netherlands corpus (basiswettenbestand).

### 5.2 Sources of Law and Legal Rules

The legal knowledge source par excellence is the written source of law. The source of law is a writing that can be, is, was, or presumably will be used to back an argument concerning the existence of a rule in a certain legal system. It is the result of a legislative act performed with the intent of creating that rule, and functions as evidence of that legislative act.

\(^1\)Compare Dutch *hulpbepaling*.  

94
Here, again, it is important to stress that the message, the representation of the intent of the legal act, cannot be the same as the product of the legal act. The main function of the source of law is to function as evidence that certain institutional events really happened. Suber (1990) notes the problem:

Note a peculiar feature of any reflexive sunset clause, for example, clause C in statute S, saying “statute S will expire at time T”. If one were to ask, some time after T, whether S were valid, the answer would clearly be no. But how do we know that? We know by reference to the wording of C. These words bind us even after they expire. Those who act as if S were still valid may be opposed in court on the ground that present law recognizes that S has expired. The sunset clause itself may be cited as authoritative on the past effectiveness of the repeal of S. But to do so means that we believe the sunset clause did not completely swallow itself. Statute S, including sunset clause C, did expire, but somehow a meta-statement about this expiration did not expire. But there was no such meta-statement in S. If the meta-statement is law, then it appears that it expired; and if did not expire it appears that it is not law. But if we insist after time T that valid legal authority can be cited for the invalidity of S, then we are appealing to such a metaphysical mystery.

The expiration of S, as an occurrence, remains part of institutional reality, or more accurately its history, even if C is no longer (part of) an active source of law. The legislator directly caused the expiration of S in institutional reality: the occurrence is not supervenient on the message.

The written source of law is therefore only the physical evidence for the creation of the rule, not the rule itself. As observed in section 4.2.2, the legislator represents an institutional fact with the intent of creating that fact, of gaining social recognition (collective acceptance) for it.

Knowledge representation also attempts to represent the institutional fact; The knowledge representation is therefore not a representation of the written source of law as such: they are rather both representations of an institutional fact, in the case of sources of law usually rules (see figure 5.1). Rules are ideally phrased as statements about the rule instead of their reifications being the representation of the rule. By doing this we both avoid the complications involved with reifying, and commit to the idea of metadata as statements about something.

The document that is considered the source of law, and the legal rules and other legal facts it represents, are subject to different ontological criteria for their existence. In the case of documents a number of different levels of existence is distinguished, as explained in section 5.3. On the item level, the physical example of the document, the document is a concrete physical object that is created in some time interval and destroyed in some time interval, and exists in between. On more abstract levels – Saur (1998) distinguishes the work, expression, and manifestation level – the document comes into existence but it
is not entirely clear whether it ever stops existing if it ever does so. As long as one physical example of the document exist it can be brought in as evidence of the past of the legal institution. In rare cases even a document that claims a certain source of law once existed functions as evidence of the existence of a rule.

What is clear is that the existence of the document in no way corresponds with the legal reality it creates. Legislation is usually delivered a considerable time before it becomes applicable, and stops being applicable at some definite point in time.

It is good practice to separate the legislative fact or rule, as an occurrence with a definite starting point and end point in time, from the medium that first – and authoritatively – represented it, without however committing to the equivocation of the institutional rule and its representation as a logical rule: the logical rule that represents a legal rule is also an ontologically distinct entity, created at a different time than the legal rule, and it can be modified during its lifetime even though the legal rule remains the same, for instance because it contains an error or turns out to be a misrepresentation of the legal rule.

Experiences with MetaLex (cf. Boer et al. (2008a), in particular section 6.4.2) have shown that care should be taken to distinguish two senses of applicability constraints on rules, in particular with respect to time and location, as errors are most likely to be made in this respect:

The context of application: Concerns constraints on the time, location, person or agent role, etc., in the action of applying the rule to create an institutional fact.

The matter to which it is applied: Concerns constraints on the time, location, involved person or agent role, etc., in the behaviour to which the rule is applied.

When applying legal rules, one should ask oneself 1) whether the rule can be applicable in the decision one is about to make, and 2) whether it applies to the case, matter, behaviour at hand.

---

2 Speculation about the laws of ancestors is actually a common theme in medieval sources. The issue became considerably less pressing after book printing became common.
In MetaLex the main issue is time and versioning of sources of law. Usually the time interval in which the rule can be applied and the time interval in which the occurrences must have happened to which it is applied are the same, but they may diverge (cf. generally Marin et al. (2005); Palmirani (2006, 2005); Palmirani and Brighi (2006)).

Applicability statements are usually of the second type. The first type has a purely auxiliary function, and is usually of lesser importance in agent reasoner. Since the context of use of the agent reasoner is more or less fixed, its representation can often be ignored (for instance the involved agents, for instance a civil administration and its client, remain the same, and the agent reasoner is deployed only in the time interval in which it can be applied).

The distinction is most obvious in normative rules; If a rule is applicable in a certain time frame, one can apply it in this time frame to determine whether some behaviour you know of is allowed or disallowed. The question whether the behaviour must also have happened during this time frame is however a separate one.

Retroactive applicability for instance normally means that the rule may be applied to occurrences that happened before the rule became applicable. It does not however mean that the rule may be applied before it exists. This means that it is possible that a behaviour is at a later point in time judged disallowed by a court, backed by the existence of a legal rule, even though the legal rule could not possibly have been taken into account when the intention to engage in that behaviour came about.

Ex post facto legislation is regarded as subversive to justice and open to abuse, as remarked by David Hume on the infamous trial of Strafford in 1641:

Better to live under no law at all, and conform ourselves the best we can, to the arbitrary will of a master, than fancy we have a law on which we can rely, and find at last, that this law shall inflict a punishment precedent to the promulgation, and try us by maxims unheard of till the very moment of prosecution.

Prohibition of punishment based on ex post facto law is implicit in nulla poena sine lege clauses all over the world\(^3\). As a general rule, retroactive application of rules that cause harm is considered taboo. On the other hand, one can argue that it is sometimes just to apply a new regulation retroactively because the legislator has become more enlightened in time. It does happen quite regularly in any case.

Similarly, and less contentiously, the law may decide later that events that happened in the past constitute some legal fact now. It is however problematic to ascribe the intent to perform some legal act to behaviour in the past before the existence of the rule that makes the legal act possible, so one does not expect this type of retroaction in relation to legal powers and potestative rights. In

\(^3\)for instance in the Constitution of the United States, art. 1, §9, cl. 3 and §10, cl. 1, or German and Dutch penal codes, both Title 1, art. 1, §1
the decision to perform a legal act the context of application of the rule and the thing to which it applies are the same.

The notion of delayed application is often seen as the mirror image of retroactive application, but is in reality something very different: In this case the maxim is that the rules to be applied are the rules as they were when the occurrence happened. This is a routine technique in law, commonly encountered in routine administrative processes. An operative principle for tax deductibility of premium paid for capital insurance products may for instance be that they are tax deductible if they were tax deductible when the taxpayer entered into the contract.

The result is that there is no single clearcut time line of valid sets of legal rules and facts that can be used to test cases against: whether a rule is applicable depends both on the events to which it is applied and to the context of application.

The world of documents initially seems simpler. Sources of law come in two types: the ones which are not modified during their lifetime (for instance court decisions) and the ones that are (consolidated legislation). The ones which are modified by the legislator usually form a neat sequence of versions, only one of which exists at each specific point in time after the initial creation of the work. There are however some exceptions (errata corrige, modification ex tunc) which create more than one version of a source of law at the same point in time. These will be discussed in section 5.3.

It is in any case clear that a document management system capable of telling us which version of a document was the one in existence at a specific point in time, does not necessarily help us answer the question which set of rules should be applied to come to a correct decision.

Existent internet portals that make the law available for citizens are usually based on the idea that there is a single applicable version of the document for each date; Moreover they may fail to distinguish between the existence of the rule and the existence of the text. It is acknowledged that this timeline may be retroactively changed by the legislator, but the timeline is always clearcut when one takes a vantage point: If the date of viewing (sichttag) is today, then the valid version of a document at some date of interest (the stichtag) can be determined.

For the casual user who has a specific context of application in mind this is perhaps the best one can do in a simple user interface. For the knowledge engineer this is not helpful, since the knowledge representation should not reflect a sichttag: section 5.3 introduces an event-based method for analysing the history of a document.

\[4\] A side note: Dutch Guidelines for Legislative Drafting (Aanwijzingen voor de Regelgeving, 168.1) specifically instructs legislative drafters not to realize retroactive applicability with a fictional enactment of the document in the past, to avoid confusion.
5.2.1 Representation of Legal Rules

An interesting question is whether the special requirements of legislating or legal reasoning tell us something about the relation between the logical form of representation and the function of the rule as we find it (represented) in the source of law. There are indeed some useful recurring patterns revolving around the applicable and constitutes properties.

A central thesis from chapter 4 is that the distinction of brute reality – the outside not under control of the legislator – and the legal institution – in which the legislator can create his own reality and decide what interfaces for interacting with it exist for others – is paramount. All rules mapping into institutional reality are default rules, because brute reality must take logical priority over institutional reality, and the rest are not, since institutional reality is not intended to be completely arbitrary.

For a functional classification we can find some inspiration in Clancey’s well-known horseshoe diagram in figure 5.2 (cf. Clancey (1985)), which describes a simple problem solving method originally used for medical diagnosis: We for instance generalize from neutral input like for instance ‘has low white blood count’ to abstractions like ‘immunosuppressed’, and from the abstracted data generate hypotheses like ‘gram-negative infection’, and then flesh this out to a concrete solution by using additional data to distinguish between the subtypes of gram-negative infection arriving eventually at ‘E-coli infection’.

Abstraction and refinement are not really inverse directions of inference: on the left side of the diagram we abstract and aggregate. In law we find analogous classification and reasoning processes. First there is a problem or conflict, which is generalized into legally relevant terms, which constitute legal facts, and these help to generate hypotheses about the possible legal instruments available for acting on the problem, and then one checks the details, the necessary conditions: you get hit by a car, which makes you ‘disabled’ and ‘unqualified for work’, which means that you generally speaking are entitled to some social security-related payment, and only then you check out the specific details. Constitutiveness is

![Figure 5.2: Heuristic Classification.](image-url)
clearly a form of abstraction, a shift of granularity level of description.

One of the traditional problems with public administration is that it traditionally leaves the responsibility for the first three steps with the citizen, by organizing administration physically in such a way that the citizen has to make the choice to which desk to go. At each desk you can get only one type of ‘solution’, ‘product’, or ‘service’. Without going into the organizational aspects of this phenomenon, this organization does tend to lead to blind spots that affect design, life expectancy, and reusability of knowledge bases and knowledge-based systems.

In van Engers et al. (2004) we describe a system we developed for the _Juridisch Loket_ in the Netherlands, a semi-public organization that gives legal first aid to people with low incomes. The system we developed for this organization handles not only the checklists for determining eligibility for some type of service, but also assists in determining the nature of the legal problem. As we described in van Engers et al. (2004) one of the more complicated problems is dealing with the fact that many of the clients have multiple problems at the same time (they were unlawfully fired, then divorced, have a dispute concerning the children and the dog, don’t have a regular source of income, and are about to be thrown out of their house) and are not capable of separating them in the interview and handling them one at the time.

Besides the special demands this puts on interaction, it also directly suggests a knowledge representation, and a style of legislating. One does not systematically try out all possible checklists. Instead one takes some elements of the story the customers have to tell as *indicators*, and based on the indicators one decides to check which list of *requirements*.

The thesis is that a lot of legislation is usually organized in this fashion: there are defeasible *indicative rules* that map from a set of indicative conditions to a relevant legal fact. Very often the intent of the customer to create this legal fact is assumed by default in agent reasoner. This is however not always the case: people who have just been fired on the spot are for instance often reluctant to start an appeal to get their job back if the layoff was unlawful. The next step is to systematically check *requirements*, which are necessary conditions.

Interestingly, the set of requirements applied by the _Juridisch Loket_, which assists people in applying for unemployment benefits, is intentionally less strict than the set of requirements used by the system of the _CWI_, which decides on actual eligibility. The two organizations disagree about the burden of proof for certain requirements: the _CWI_ is of the opinion that it is always the applicant’s burden, and also has its own policies for accepting and denying evidence, while the _Juridisch Loket_ takes the point of view that the _CWI_ legally speaking has considerable freedom in these matters, and the applicant in any case has a right to appeal against unfavourable decisions. The agent reasoner not only implements the *legal rules* the interface shows, it also represents a policy decision.

---

5 previously _Bureau voor Rechtshulp_  
6 I.e. “customer wants to create this fact” is not part of the explicit conditions on the advicer’s screen.  
7 My own observation.
of the organization on burden of proof: what information must be supplied before we accept the application? The administrative organization usually even invents its own rules for evidence: the requirement that the applicant must be a subject of the Netherlands is for instance habitually translated into a requirement that the applicant adds a copy of a valid passport by some types of organizations, while others will happily accept verification by oath (i.e. they simply ask).

Similar structures are found in the argumentation literature (cf. Gordon et al. (2007)): indicative rules are used to construct pro arguments, while requirements are used to attack them on the grounds that the conclusion only follows if the necessary condition is met. In this case the burden of proof determines whether it is sufficient to merely question whether the requirement is met, creating a duty for the other side to produce evidence that it is, or whether actual evidence needs to be presented that it is not. In essence the same type of dialog occurs in administrative settings, except that the degrees of freedom are considerably reduced (viz. Boer et al. (2007b)).

The source of law contains rules of the following logical types:

**Institutional rules:** Terminological axioms are of the form \( r : \alpha \subseteq \beta \) or \( r : \alpha \equiv \beta \), where \( \alpha \) and \( \beta \) are concepts wholly belonging to the institutional ontology of the institution the legislator intended to create the rule for, and \( r \) is the identifier of the rule. Institutional rules are intended to constrain the use of terminology, and are fairly rare in sources of law. Terminological axioms are not intended to be defeasible.

**Requirement rules:** The purpose of constitutive rules is to specify how institutional reality depends on brute reality. As shown in section 4.2.3, constitutive rules sometimes specify necessary conditions on the mapping between institutional reality and brute reality. Necessary rules are of the form \( r : \alpha \subsetneq \beta \), where \( \alpha \) and \( \beta \) are concepts, and \( \alpha \) wholly belongs to the institutional ontology of the institution the legislator intended to create the rule for. \( r \) is the identifier of the rule. Necessary conditions are not intended to be defeasible.

**Indicative rules:** Indicative conditions function as the user interface of the institution: they provide the mechanisms by which institutional reality can be changed through action in brute reality. Since brute reality provides the independent variables, indicative rules are treated as defeasible rules, even if they are phrased as definitions. Indicative rules are of the form \( \alpha \text{ then } \beta \) where \( \alpha \) and \( \beta \) are concepts, and \( \beta \) wholly belongs to the institutional ontology of the institution the legislator intended to create the rule for. Indicative rules are prevalent. Besides their prevalence in sources of law, they are also the category of rules that is easily delegated to lower legislators. The freedom to make contracts also comes down to the freedom to define one’s own indicators for certain (already existing) categories of legal facts.
**Burden of proof rules:** While the previous categories only constrain the valid models of institutional reality, decision making about institutional reality is normally based on a specific mapping from evidence to institutional fact that must be settled before decision can be taken. In some cases such requirements are explicitly found in legislation. The burden of proof assignment is always a rule, and usually a constraint, although burden of proof rules may for instance also “relax” an indicative rule by turning a proposition that must be known into one which may be assumed.

Normative rules are from the point of view of knowledge representation complexes of rules, often consisting of both requirements and indicators. This is the reason not to add them to this list. The normative rules are essential for creating normative order: in some fields of law they may be rare, but their presence is essential.

### 5.2.2 Application of Rules

The autoepistemic burden of proof rule requires judicious use. It by itself does not automatically solve two important burden of proof problems for agent reasoner:

1. The burden of proof for necessary conditions usually means that there is an *independent* argument for why the condition is met, that is if the necessary condition is $N$ for $C$ $(C \subseteq N)$ there is another argument for $N$ besides the trivial and tautological $([C, C \subseteq N], N)$.

2. If in the current interface of the legal system there are only a few given rules to create legal fact $C$, the argument for $C$ must be based on one of these rules.

The reconstruction of arguments is not really a direct knowledge representation issue: tracing the reasoning method will produce the available ingredients for construction arguments for propositions. The second problem is in principle easily solved by explicitly enumerating alternative sets of conditions if you already know all rules that can generate $C$.

These problems cannot be solved inside a modular knowledge representation of legislation itself: these are exactly the type of assumptions we try to avoid in a context-neutral, reusable, and durable knowledge representation. Consolidated legislation is also usually sketchy on the issue of burden of proof: valid evidence rules are for instance most often found in case law. These issues relate to the epistemic competence of the agent reasoner (does it know all relevant rules for determining $C$, or is there reason to believe that some are unknown to the designers of the agent reasoner?) and policy decisions to be made by the agent reasoner user (what evidence do we routinely accept for $C$, and are we open to unforeseen categories of evidence in the main production process?).

---

8The administrative organization may decide, and it often does, that although there may be unforeseen but valid arguments, the appeals process will sort this out.
agent reasoner developers will however have a modest requirement for these purposes for a knowledge representation for legislation: it is very useful to know which rules are applied. It is better to state that \( C \) must have been the result of the application of \( r_1 \) or \( r_2 \), from the point of view of maintenance and compactness, than to repeat the conditions in these rules in a complex constraint rule.

Rules need to be explicitly identified and referred to for several reasons:

1. Firstly, if an important function in legal reasoning is to back arguments concerning their existence, they must be explicitly referred to;

2. Secondly, indicative rules (including normative ones) are sometimes subject to choice, and must be compared to determine which one to apply. Moreover, if it is not apparent which rule has been chosen we would also not know which one has been applied if no bookkeeping is done; and

3. Lastly, which rules have been applied can be a burden of proof issue: generally speaking institutional facts are only recognized if we know the rule by which they came into existence.\(^9\)

These concerns however apply to the legal rules that have been applied and not to the logical rules that represent them. An obvious solution is to do exactly what this explanation suggests: use an explicit representation of argument, and keep track of these during the inference process. If the logical rules are identified by a logical constant \( r \) and the represented legal rule by \( r' \), we can state explicitly \( \text{represents}(r, r') \), and keep track of the fact that \( (\{r, \ldots\}, C) \).

If we however commit to the principle that the logical rules are statements about the legal rules, we can avoid the reification. Central to this better solution is an explicit applicability predicate, always attached to the object the logical rule is about (i.e. the condition).

### 5.2.3 Representation of Applicability Rules

Applicability statements (cf. Prakken and Sartor (1995); Prakken and Schrickx (1991)), statements of the form *article \( n \) is [not] applicable to \( C \)*, are similar to the reading of requirements as necessary conditions.

Slightly more complicated are applicability statements of the form *the rules of act \( R \) are [not] applicable to \( C \), ministerial directives are only applicable to \( C \), or the rules of act \( R \), and rules dependent on it, are [not] applicable to \( C \).* These involve the grouping of rules into sets of rules.

When stating applicability restrictions on groups of rules it is however important to keep in mind that the applicability restriction only applies to things of the correct ontological category. If a statute \( R \) for instance only “applies to civil servants”, the knowledge engineer has to add himself the restriction that

\(^9\)This is however obviously not a terminological truth: we often accept institutional facts simply because we trust the source of information.
this means “applies to actions of civil servants only where it concerns actions and civil servants only where it concerns actor roles”.

In reality such broadscoped applicability restrictions often require reading the entire statute, as it is not inconceivable that such a statute that applies to civil servants has some rule that addresses the powers of others towards civil servants.

5.2.4 Also-applicability and Legal Fiction

A problem still exists with a certain formulation of applicability rules: the also applicable rule. When there is a rule \( n \) is applicable to \( C_1 \) and a rule \( n \) is also applicable to \( C_2 \) (which is not a \( C_1 \)), or alternatively to be disallowed by \( n \) it must be \( C_1 \) and \( C_2 \) is also disallowed by \( n \), we have a problem with our reading of the rule as a necessary condition.

In this case we really do need an edit of the first rule to accommodate the second (as suggested in section 4.2.3 as a less preferred solution), or a bizarre legal fiction.

The legal fiction can be characterized in two ways. Functionally speaking it is generally a rule intended to extend applicability of an existing rule to matters that were clearly not covered by the original rule. If theft is the taking of a good, but another rule extends theft to selling something that belongs to someone else, then this rule creates a fiction if according to the original rules theft is necessarily a taking: selling something also becomes a taking, even though this is contrary to common sense and existing requirements.

Since producing evidence to the contrary does not defeat such a rule, the conclusion of a legal fiction can be considered an undefeatable legal presumption. This is contrary to a fundamental principle stated several times: that brute reality takes precedence over institutional reality. As Jeremy Bentham pointed out:

Fictions are to law what fraud is to trade.

To accommodate it within this framework, one needs to clone all rules about Theft that do apply to the sale as theft, but applied to the new concept SaleAsTheft, and closely monitor future changes to Theft to see whether they are also compatible with SaleAsTheft\(^{10}\).

This representation is no less elegant than the original. Also-applicability is a pathological legislator’s instrument, particularly when used by a lower legislator to whom the legislative authority to create additional, usually indicative, rules for creating a certain institutional fact has been delegated. The applicability constraint limits the scope of this delegation, and the also-applicable rule bluntly ignores it to expand the scope of applicability. It is a deliberate misclassification. It is occasionally used by courts, and as a legislative technique\(^{11}\).

\(^{10}\)The legislator may change legislation with the intent of destroying such an established fiction. This may lead to the introduction of new clauses that are idiotic from a common sense point of view, explicitly denying things that make no sense in the first place.

\(^{11}\)Legal fiction is prohibited by AR’ 90, art. 61, in the Netherlands, but this is no guarantee
5.2.5 Purposes of Rules

The identified recurring patterns using the *applicable* and *constitutes* properties can be used with a large variety of different rules. In this section we run by some examples, categorized by the purpose the legislator had in mind when creating them.

The legislator believes that by shaping the institution in certain ways certain beneficial effects are brought about in brute reality. The intention of the legislator is generally to improve the normative order: the legislator expresses his own preferences through rules, creating social mechanisms, or at least proposing them, that will have certain effects on behaviour if certain assumptions of the legislator about the preferences and abilities of other agents are met.

The most obvious category of legal rules that has such an effect on normative order is the normative rule. What normative rules do is nothing but attaching the institutional qualifications allowed and disallowed to certain behaviours; The legislator intends us to choose the allowed alternative over the disallowed one. The legislator believes we will do so, because not doing so involves a risk. The allowed and disallowed qualifications perform their function because they are embedded in a larger system of rules that create an incentive for other people to take actions that work out negatively for the agent who violates a norm. This whole system obviously depends on a correct judgment on the preferences of agents addressed by the rules.

The normative rule brings about a legal fact that is evaluated negatively by those who bring it about.

Normatives rules are however not the only mechanisms through which the legislator manipulates the normative order in a society. Section 4.2.2 proposed a classification, based on Sartor (2006a), that distinguishes a number of components that also suggest a function of the rule.

For instance the intention to perform a legal act/bring about a legal consequence may be a condition for application of the rule: This gives people the freedom to choose whether or not they bring about this legal fact.

The legislator may however also restrict the intention with which the legal act may be performed. If $T$ is this inteneted task, then there is an applicability rule that says that the rule only applies to action performed to execute $T$.

This means that if one tries to perform the legal act, but without the right intention, then the legal act does not happen. This is not the same as a normative rule which states that such a thing is disallowed. Notions of abuse of power by for instance civil servants depend on normative rules.

Some rules are clearly intended to be used in one’s own interest, whenever one likes, but these cannot be distinguished by form. The absence of applicability rules applying to intention is weak evidence for such freedom. One could consider this a power-based version of the distinction between strong and weak permission, which will be discussed in greater detail in section ??: the power

that it isn’t done. Strangely, the official explanation of the article then uses also-applicability as an example method of working around the prohibition.
is always weak, and its systematic protection can only be organized through non-infringement on it by legislators.

The last recognized category from Sartor (2006a) is the legislator’s intent to enable people to bring certain things about that could not be brought about in the absence of the rules.

In some cases the legal rule only confers the benefit of legal recognition to actions that would also take place without it (for instance buying and selling), while in other cases the legislator creates a rule solely for the purpose of creating a recognizable way to achieve a certain legal effect (for instance a permit application procedure for gaining permission for constructing a shed in your garden). The legislator sometimes explicitly has to create a recognizable way of achieving certain novel legal effects.

In these cases the rules are created in combination with certain other administrative actions: an administrative body or function is set up, a business process is created that implements the intended function, a budget is made available, etc.

The legislator for instance prohibits theft, making theft, which presupposes a victim to whom the good taken belongs, disallowed. The disallowed qualification creates the power, which may be exercised in one’s own interest, to report the crime to a body set up for accepting those reports. The body has the power to charge the suspect of the theft with theft before a criminal court, which is to be exercised only in the public interest. The court then has the obligation to determine whether the crime happened, and to penalize the one responsible for the crime, which is supposedly in the interest of the suspect, because it is a foreseeable consequence of reporting the theft. The one who committed the theft generally prefers not to be penalized, and it was foreseeable that committing the theft entailed that risk.

As pointed out by Valente (1995), we might think of a creative function of some rules: some rules in a way create the courts, public prosecutor, etc, on the institutional level, and there is a quasi-obligation on the legislator that there is also something that constitutes it. These are simple clauses that postulate the existence of certain legal facts, with or without a constituting base in brute reality. Postulating the concept, which is by definition non-empty, creates the (possibility of a) court; Declaring instances of it populates the concept. And a judge in a court may for instance be appointed by royal decree, creating an institutional fact.

The class of auxiliary rules is not identified by its form, but by the regulated domain. Auxiliary rules are identified by the fact that they do not apply to the primary domain regulated by the source of law, but to the context of application of the source of law. To this category legal facts relevant to the legislative domain belong. Auxiliary rules are all those rules that exist because legal institutions have become so complex that simply publishing a simple list of rules creating the intended normative order no longer works.

Some of these are trivial. In the Netherlands, for instance, formal law declares how it is cited: This act will be cited as the Act on X and abbreviated to
Other countries simply have a custom that the title is cited, or the date of delivery plus an optional issue number etc.

Choice rules are applicability rules that make the applicability of a legal rule conditional on the applicability of another legal rule. They in other words require that one make a choice between applying two rules, and very often prescribe which one. This makes them normative, but on another level. Representation of choice rules is covered by section ??.

The source of law may also state auxiliary facts that allow one to derive applicability conditions. Instead of stating that this regulation is applicable from \( t_1 \) to \( t_2 \), and may be presumed to apply to occurrences that were initiated between \( t_1 \) to \( t_2 \) a regulation will typically state this regulation enters into force on \( t_1 \) is repealed on \( t_2 \), making the meaning of these statements dependent on other law or customs to the effect that a regulation is active and therefore applicable from enactment to repeal. A list of rules would presumably be applicable from assertion to retraction: legislators have however added another layer of complexity by distinguishing the existence of a rule and whether it is active or inactive if it exists. Some legislators allow rules to go into the active state repeatedly.

Explicit delineation of institutional realities may be considered an important auxiliary function, although absence of such rules does not prove that such a delineation does not exist – making all clauses part of the same institutional reality – or alternatively that all sources of law are completely unrelated.

Sources of law may for instance note explicitly that certain concepts are the same concept as the one in other sources of law (e.g. wages as defined by the act on income taxes and dependent regulations). Sharing concepts should be distinguished from mere confluence of terminology because of a shared abstraction (every jurisdiction has its own version of allowed and disallowed, similar in meaning but belonging to different institutional ontologies) or the deep copying that we for instance see a lot in insurance policy contracts: every different product has the same definition section copying the same definitions of the same terms verbatim, but they are still different concepts because they can be changed separately if the need arises.

More important for delineation of institutional realities are mandate, submandate, delegation, and subdelegation (cf. Boer et al. (2002b); Zijlstra (2001))\textsuperscript{13}:

\footnotesize
\textsuperscript{12}AR '90, art. \textsuperscript{—}
\textsuperscript{13}This conceptualization is borrowed from constitutional and administrative doctrine in the Netherlands.
\textsuperscript{14}I have once proposed to rename this to imperium, and to limit jurisdiction to the courts.
\normalsize

\textsuperscript{107}
Other legislators, subnational and supranational ones, receive their legislative competence from these initial legislators either by mandate or by delegation; The latter means that the original legislator cannot exercise it anymore (i.e. the delegate receives exclusive jurisdiction as opposed to shared jurisdiction). Furthermore the legislator who receives legislative competence by mandate or delegation may or may not further subdelegate or submandate it.

This structure may be explicit in legislation, which is more likely in civil law jurisdictions, or simply a result of recognition of unwritten customs.\(^{15}\)

Some of the previous patterns for legal acts also apply here: the legislative competence is often constrained by pre-existing applicability rules, and may only be used with a certain intention (for a certain purpose). When the competence is delegated to make rules that create legal facts in the same institutional reality, the rules must obviously also conform to the structure of that institutional reality, as defined by its institutional ontology and other necessary rules: this is why also-applicability (previous section) is a legislative pathology.

For agent reasoner development auxiliary rules rarely have to be part of the knowledge base explicitly: they are only relevant for determining which sources of law to search for to account for the rules regulating a certain administrative process. When a knowledge representation must however be reusable, jurisdiction, also on matters of substance, must be clearly delineated.

Typically this can be done – without explanation where such constraints came from – with applicability rules. The important part is the form of the condition: the legislator causes a change that initiates the rule. The properties of the agent that initiated the rule determines the jurisdiction.

The jurisdiction concept, and all complications associated with it, arises because legislators recognize each other’s existence. Both the tendency of delegating legislative competence to independent administrative agencies and complex supranational bodies like the European Union tend to complicate the determination of jurisdiction, and lead to an expansion of the role of auxiliary clauses in legislation.

5.3 Sources of Law and the Semantic Web

MetaLex XML positions itself as an interchange format, a lowest common denominator for other standards, intended not to necessarily replace jurisdiction-specific standards in the publications process but to impose a standardized view on documents that function as source of law for the purposes of software development. The MetaLex schema is based on best practices from amongst others the previous versions of the MetaLex schema, the Akoma Ntoso schema, and the Norme in Rete schema. Other important sources of inspiration are i.a. LexDania, CHLexML, FORMEX, R4eGov, etc. In addition to these government or

\(^{15}\)A good guide is the constitutional history of the country: bodies that factually existed when the constitutional arrangements of the country took their shape, are typically not created by constitutional law in that country and their legislative competence may be a factual customary one which is recognized.
open standards there are many XML languages for publishing legislation in use by publishers. Standards like PRISM, in which major publishers are involved, are also a source of inspiration.

The MetaLex XML standard recently moved forward significantly, with the adoption of part of it as a CEN\textsuperscript{16} prenorm, and its adoption by several industry projects. Many of the participants of the CEN workshop have also been involved in the Legislative XML workshops (see for instance the archive of the frontpage of the MetaLex website\textsuperscript{17} for previous calls for participation and online proceedings and presentations). In the process of standardization MetaLex changed significantly compared to its previous incarnations (versions up to 1.3.1) for which the author of this book was mainly responsible.

MetaLex is the subject of earlier publications, e.g. Boer et al. (2007b,a); Boer and van Engers (2005); Boer et al. (2002b,a, 2003a); Winkels et al. (2003); Boer and Winkels (2005); Boer et al. (2004b,c, 2003c). MetaLex is a generic and extensible 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, as well as emerging Semantic Web standards like RDF and OWL.

MetaLex, whose first version dates from 2002 (cf. Boer et al. (2002a)), has been redesigned from scratch in the CEN standardization workshop, taking into account lessons learned from Norme in Rete\textsuperscript{18} – the Italian standard for legislation – and Akoma Ntoso\textsuperscript{19} - the Pan-African standard for parliamentary information, and has been submitted as a norm proposal to the CEN.

A partial CEN Workshop Agreement (CWA) exists. This partial agreement contains agreements about the abstract content models supported by the standard, the way metadata is added to a document, and a generic model for organizing metadata in RDF. The description in this section reflects the features of the standard of interest to and mostly designed by the authors of this report.

The CEN Workshop on an Open XML Interchange Format for Legal and Legislative Resources (MetaLex), declares, by way of its title, an interest in legal and legislative resources, but the scope statement of the first workshop agreement limits the applicability of the proposed XML standard to sources of law and references to sources of law. As understood by the workshop, the source of law is a writing that can be, is, was, or presumably will be used to back an argument concerning the existence of a constitutive or institutional rule in a certain legal system, or, alternatively, a writing used by a competent legislator to communicate the existence of a constitutive or institutional rule to a certain group of addressees.

\textsuperscript{16}Comité Européenne de Normalisation; European Committee for Standardization; Europäisches Komitee für Normung.
\textsuperscript{17}http://www.metalex.eu
\textsuperscript{18}http://www.normeinrete.it/
\textsuperscript{19}http://www.akomantoso.org
An important aspect shared by this book and the MetaLex XML standard is the central role of actions and events in interpreting the law. MetaLex however uses another conceptualization of action. Section 5.3.1 explains the relevance of events and actions for a standard for marking up documents.

Identification (section 5.3.3) and reference (section 5.3.4) are also central topics: for the purposes of legal knowledge engineering no functionality is more important that the capacity to make stable and precise links to the things one is representing. The alignment of knowledge bases and sources of law, in particular when it comes to coordinating versioning of both, is the topic of section 5.4.

### 5.3.1 Metadata and Legislative Occurrences

MetaLex consists of a standard for document structure and identification, and a metadata standard. MetaLex organizes metadata around occurrences – events and actions. For metadata this is uncommon, but it is in effect the same thing as we have done so far with the interpretation of the subject matter of legal rules: these apply to actions, or to the result of actions. There are several good arguments for organizing metadata about legal documents around events and acts, besides a general descriptive fidelity argument for law.

Arguments can be characterized as arguments from knowledge engineering strategy, and arguments from legal theory and practice. All are relatively straightforward and unsurprising in this context, but they are nevertheless largely ignored in metadata vocabularies for legislation. A single attribute-value pair, with the document as implicit subject, is often used for such information items as the date of promulgation, instead of reifying the publication/promulgation event and treating the date as an attribute of the event.

A particular metadata description is usually about (a snapshot of) some information entity (taken) in a particular state – a perceived stability of the entity over a particular time interval that does not take account of changes that are outside the domain of interest. The granularity of that snapshot varies across metadata vocabularies, depending on the targeted community.

This is apparent in the IFLA FRBR conceptualization of bibliographic records (cf. Saur (1998)): it groups hierarchically the products of different types of events in the categories work, expression, manifestation, and item. When you make a copy, the item identity changes, but descriptive metadata stays the same. When you add or change metadata statements attached the document, which apply to manifestation, expression, or work, the manifestation changes, but the expression stays the same, when you edit the text, the expression changes, but the work usually stays the same, etc. When you plagiarize someone else’s text, you hardly change the expression, but you do create a whole new work.

To a community that works with certain legislation daily, the insertion of a new provision is for instance an important event to be noted, and even to prepare for; For the casual reader it happens to be just one of the many constituting parts of that document’s state at the moment of consulting.

There are legal theoretic arguments to be made for the importance of event and act descriptions, and the central one is found in the institutional interpre-
tation of the role of legislation (or contracts, or driver’s licenses, tax statement forms): One undertakes a legal act on the institutional level by producing a written statement in accordance with a certain procedure. In this reading the document is the mere physical residue of the intentional act that is really important: it functions as physical evidence that a constitutive act that modified institutional reality happened, and it declares the intent of the act. Evidence is not only found in the central position of legal action and declaration of intent (or will) in legal doctrine, but also in terminology like “Act of Parliament” when one is referring in actuality to the physical result of that act of Parliament. It is the act that matters.

In the MetaLex CEN workshop (see e.g. Boer et al. (2007b)) the widely used classification of event participants by Judith Dick (cf. Dick (1991)) is used: it is interesting to note that Dick developed this vocabulary to describe legal text, even though the vocabulary itself is very clearly generic and used in different domains. In this book a more generic format is proposed: Dick’s vocabulary has a very functional flavour, and the distinction between types of participants is too arbitrary.

There are several good reasons from the point of view of knowledge engineering, to explicitly reify the events.

One is supplied by Lagoze (see Lagoze et al. (2000)): for establishing semantic interoperability between different metadata vocabularies and for developing mechanisms to translate between them it is only natural to exploit the fact that some types of entities – people, organizations, places, dates, and events – are so frequently encountered that they do not fall clearly into the domain of any particular metadata vocabulary but apply across all of them.

It is very clearly the event, or more specifically act, that plays the mediating role between these entities and the resource the metadata description is about. The natural coherence between for instance between author, publication date, and publication channel information (e.g. state gazette bibliographic information) is apparent to all: all are participants in the publication (promulgation) event.

Some other reasons were noted by i.a. the author of this book (cf. generally Boer et al. (2004a)). Relevant events often transform “input” resources into “output” resources, at the expression or manifestation level, and the respective metadata descriptions for those input and output resources are often the data about the event, i.e. they are shared by the input and output resource: only the perspective is different.

In formal legislation, there is for instance a natural coherence between the old consolidation, the new consolidation, the modifying legislation, the modifying authority, and the modification date. The modification event, if identified explicitly, links together three different but related resources, and interesting metadata about them.

Different perspectives on this exact same event, because its identity was not made explicit, may yield incompatible metadata descriptions, result in unnecessary duplication of metadata, and several separate occasions in which to make mistakes, therefore unnecessary maintenance, and, lastly, the loss of rele-
vant references between documents. Explicitly identifying events increases the reliability of the metadating process.

As noted in Boer and Winkels (2005), keeping track of changes is especially relevant to law because we have to presume that the law does not become better over time. For most written resources, whether fiction or non-fiction, the last version dominates all others because it is the best: only rarely are we interested in anything other than the current state of the work. We trust that if there is a notable difference at all between today’s edition of a book and the first one, today’s version will be better. In law we are interested. A tax administration will for instance routinely work at any point in time with at least three different versions: the running tax year, the previous tax year, which is processed now, and the next tax year, which is being prepared. Things like retroactive and delayed applicability, and ex tunc and prospective versions of legislation, complicate the determination of the current information state about a document; Information on events that happened remains true, and is more easily codified if information is missing.

A version ex tunc of a bibliographic work is the product of a fictional legal event that is a temporal fiction of some other event – the constitutive event – that happened after the fictional event. The fictional expression creation allows rewriting institutional history, for instance through errata corrigae, or at the occasion of an annulment of a modification made in the past by a constitutional court. The fictional event is treated as if it were a true event after the constitutive event happened. Institutional history therefore appears different depending on which vantage point in time one takes, as already explained (the sichttag/stichtag distinction in section 5.2. What one wants to store is however not the snapshot and a description of the vantage point from which one took it, but simply the fact that some brute act caused a legal one in the past.

A prospective version also depends on the vantage point, but is conceptually something completely different. It is a manifestation of what is to be expected to be a future version of a work, made when that version doesn’t yet exist. Modifying acts are generally published well before the actual modification takes place, in order to give society time to prepare for the changes. It is only natural that early renderings of the future version are made, although there is a risk that the modifying act is retracted or changed in the meantime. Prospective versions may never become law, although they may on occasion still function as evidence that there was at a certain point in time an expectation that it would.

In Boer et al. (2004a) the point is made that the expectation of certain events also functions as a conceptual coat rack for missing information, which was nevertheless essential to the involved organisation, a tax administration, in its preparation for future legislation. I refer to that paper for details. We may know that certain events have happened or expect them to happen, but for instance cannot put a date on it, although we can infer some constraints on it. Essential was in this case that the Uniform Resource Identifier (URI) used in RDF metadata is not a unique bijective identifier: multiple identifiers can refer to the same event (but not vice versa obviously), and what are initially believed to be separate events can – by just stating their equality – be unified.
without changing the metadata. If we believe that two changes must be made, we can believe in two events, and later merge them into one if it turns out they are made as one action.

In Dick’s conceptualization, used in MetaLex, each *occurrence* has one or more participants: Figure 5.3 shows the classification of participants. The *patient* is for instance immanent and product of the action, and undergoes some structural change as a result of the action: at the level of bibliographic entities this applies to the work, while the expression usually takes the role of result or instrument. The instrument is immanent and source of the action, and is not changed during the action: this is for instance the *modifying* expression in a modification of a work, which results in a new consolidation. One of the greater qualities of thematic classification of participants is that it is largely impervious to differences in legal theory.

Consider for instance a Minister of Finance with the competence to index amounts in taxation for the purpose of dealing with inflation. At date $t_1$ he publishes a directive $s_1$ to modify income tax law $s_2$ at date $t_2$ to compensate for inflation, resulting in $s_3$. He uses a specific legislative competence for this purpose delegated to him by $s_{comp}$. In MetaLex terms we are dealing with the following participants:

The first action:

1. agent: the minister of finance;
2. date: $t_1$;
3. result: $s_1$;
4. instrument: the competence based in $s_{comp}$.

A background action:

1. agent: some legislator;
2. date: some time before $t_1$;
3. result: $s_{comp}$.

The second action:

1. agent: the minister of finance;
2. date: $t_2$;
3. result: $s_3$;
4. instrument: $s_2$;
5. instrument: $s_1$;
6. instrument: the competence based in $s_1$. 

113
Figure 5.3: Each MetaLex occurrence has one or more participants. The figure shows a taxonomy of participants.
Figure 5.4: Legislative events in MetaLex.

Figure 5.4 shows the results relating documents $s_1$, $s_2$, $s_3$ to events, dates, and persons. It is possible to replace the second action by an event. The difference is that the minister of finance (as an office) no longer has to exist at $t_2$, which is in this case immaterial. The directive to act in a certain way at a certain time can be violated, while an event of this type is a purely institutional fact that occurs by definition.

There appears to be a dislike of event descriptions on esthetic grounds in the XML community: they are perceived as in some way less real or objective than dates, persons, and places, and there are obviously no established methods for identifying them, comparable to those for times, places, and persons. This results in reluctance in attaching URIs to these events.

In law, however, it is important and therefore generally clear whether legislative acts (signature, promulgation, enactment, modification, repeal, etc.) happened, and their determination obviously cannot be less objective than for instance the determination of the dates at which they happened.

Organizing information in this way also makes it clear that this legislative domain works in exactly the same fashion as buying and selling, or traffic, or other domains of law. Documents are manipulated in brute reality, while legal rules and facts are manipulated in institutional reality.
5.3.2 MetaLex XML

The use of bibliographic terminology in the MetaLex standard is inspired by the *IFLA Functional Requirements for Bibliographic Records* (cf. Saur (1998)):

- A bibliographic object is a bounded representation of a body of information, designed with the intent to communicate, preserved in a form independent of a sender or receiver. A bibliographic work, expression, manifestation, and item are bibliographic objects.

- A bibliographic citation is a representation of a bibliographic identifier of a bibliographic object, with the intent of referring to that bibliographic object. *Article 1, the first article* and *the previous article* are examples of citation, and *the Minister, the President of the Republic, the accused*, and *We, Beatrix* are examples of references to other, interesting but non-bibliographic, things.

- A unique bibliographic identifier identifies a bibliographic object uniquely. The *uniform resource identifier* is used as a unique bibliographic identifier in Metalex.

- A bibliographic work is a bibliographic object, realized by one or more expressions, and created by one or more persons in a single creative process ending in a publication event. A work has an author or authors, and is the result of a publication event. We recognize the work through individual expressions of the work, but the work itself exists only in the commonality of *content* between and among the various expressions of the work: it is an intentional object.\(^{20}\)

- An bibliographic expression is a realization of one bibliographic work in the form of signs, words, sentences, paragraphs, etc. by the author of that work. Physical form aspects, as typeface or page-layout, are generally speaking excluded from the expression level. Any change in *content* constitutes a gives rise to a new expression. If an expression is revised or modified, the resulting expression is considered to be a new expression, no matter how minor the modification may be. Expression is an intention object.

- A bibliographic manifestation embodies one expression of one bibliographic work. The boundaries between one manifestation and another are drawn on the basis of both content and physical form. When the production process involves changes in physical form the resulting product is considered a new manifestation. Thus, a specific XML representation, a PDF file (as generated by printing into PDF a specific Word file with a specific PDF distiller), a printed booklet, all represent different manifestations of the same expression of a work. Manifestation is an intention object. A MetaLex XML element is a bibliographic manifestation.

\(^{20}\)I.e. it exists only as the object of one’s thoughts and communication acts, and not as a physical object.
• A bibliographic item exemplifies one manifestation of one expression of one work: a specific copy of a book on a specific shelf in a library, a file stored on a computer in a specific location, etc. Items stored on a computer can be easily copied to another location, resulting in another item, but the same manifestation. This makes adding metadata about the item to the item in principle impossible. On the Internet generally speaking only the uniform resource locator (URL) is an item-specific datum. An item is a physical object.

Figure 5.5 shows the relationships between the four levels of ontological stratification for bibliographic objects. A Metalex XML document is a standard manifestation of a bibliographic expression of a source of law. Editing the Metalex XML markup and metadata of the XML document changes the manifestation of an expression. Changing the marked up text changes the expression embodied by the manifestation. Copying an example of the Metalex XML document creates a new item.

Work, expression, and manifestation are intentional objects, i.e. they exist only as the object of one’s thoughts and communication acts, and not as a physical object. An item is a physical object. Note however that items stored on a computer can be easily copied to another location, resulting in another item, but still an instance of the same manifestation. This makes adding metadata about the item to the item in principle impossible. On the Internet generally speaking only the uniform resource locator (URL) is an item-specific datum. The item level is therefore not very relevant to XML standards.

The proposed standard is primarily concerned with identification of legal bibliographic entities on the basis of literal content, i.e. on the expression level, and prescribes a single standard manifestation of an expression in XML. Different expressions can be versions or variants of the same work. In addition there is the aspect of role, that relates the bibliographic entity to specific contexts of use: this is consistently treated as metadata.

A MetaLex XML element is characterized by a name, a content model, and zero or more attributes. These are the fundamental content models of MetaLex:

container a container of a sequence of other elements;
hcontainer a hierarchical container of nested elements with titles and numbers;
block the largest structure where text and inline elements mix freely, e.g., paragraphs and other (usually vertically-organized) containers of both text and smaller structures;
inline an inline container of text and other inline elements (e.g., bold); and
milestone an empty element that can be found in the text (as opposed to meta).

The philosophy behind content models is explained in Vitali et al. (2005), and i.a. Boer et al. (2007b) in the context of MetaLex.
Figure 5.5: A taxonomy of bibliographic entities in MetaLex.
Figure 5.6: **articleType** is a restriction of **hcontainer**.

Conformance in the strict sense means 1) validation of XML documents against a schema that includes the MetaLex XML schema, 2) the theoretical possibility of obtaining an XML document that uses solely MetaLex generic elements and validates against the MetaLex XML schema by way of simple substitution, and 3) conformance to the MetaLex CWA written guidelines. Any XML encoding is *transformation conformant* if instances can be transformed automatically into conformant MetaLex XML instances.

The process of declaring a concrete element conforming to the MetaLex norm works as follows:

1. You must use one of the abstract content models for the element;
2. You may define a restriction of the corresponding concrete type;
3. You may not define an extension to the content model of a concrete type;
4. You may define an extension of a concrete type for the purpose of adding attributes;
5. You must define the elements as a substitution group of one of the abstract elements and you must identify a type which is either one of the provided concrete types, or the restriction of the content model or extension of attributes of a concrete type that you have defined.

To easily define an element conforming to the standard that can be used in XML manifestations of sources of law, define a non-abstract complex type, for instance a restriction **articleType** of **hcontainerType** (see figure 5.6), and create an element belonging to the substitution group of one of the abstract elements according to the subtype specified, for instance:

```xml
<xsd:element name="article" substitutionGroup="e:abs-hcontainer"
             type="articleType" />
```

Existing vocabularies can usually be redefined in terms of MetaLex content types. It is not sensible to give an example of a MetaLex XML instance here because no such notion exists; MetaLex is intended as a metaschema for other schemas that define concrete XML vocabulary.

Existing vocabularies can usually be redefined in terms of MetaLex content types. It is not sensible to give an example of a MetaLex XML instance here.
because no such notion exists: MetaLex is intended as a metaschema for other schemas that define concrete XML vocabulary.

MetaLex prescribes what counts as a MetaLex metadata statement, how it is stored inside a MetaLex document, and what classes of entities and which predicates (properties) MetaLex distinguishes: its ontology. The RDF ontology is of course extensible. The ontology classifies:

- **bibliographic entities**: the work, expression, manifestation, and item level, and content models;

- **reference**: type of reference between bibliographic entities;

- **activities**: actions and thematic links, and thematic roles of bibliographic entities in at least the actions creation, enactment, repeal;

- **agent and competence**: the agents and institutional instruments (legislative power, etc.) used in legislative activity.

MetaLex uses the conventions of RDF/A processing for embedding RDF metadata statements inside MetaLex XML.

The purpose of MetaLex embedded metadata is nothing more than storage of RDF formatted metadata in MetaLex XML. An RDF description of a resource consists of a set of statements. The MetaLex standard includes an OWL schema that specifies commonly required properties and classes in RDF statements about legal and legislative resources. This schema may be used with RDF stored outside the document in question, and the embedded metadata processing mechanism may be used with other metadata schemas like Dublin Core or PRISM.

The main difference between storage inside and outside the standard XML manifestation is the identification of the metadata author: the metadata inside the document is associated to the editor of the manifestation, who may be presumed to be the author of the metadata.

A MetaLex document must declare what it is a manifestation of, as follows:

```xml
<meta id="m1" about="" rel="metaex-owl:exemplifies" href="/tv/act/2004-02-13/2/tv"/>
```

Other metadata may be embedded.

RDF/A statements may be added to any MetaLex element if the content model allows it. Elements derived from the `metalex:urMetaType` type must contain RDF/A attributes expressing an RDF statement. Relative URI references in RDF/A attributes are relative to the `xml base` of the containing element.

An RDF/A element is any XML element that contains either the attribute `property`, `rel`, or `rev`. Exactly one RDF statement is generated per `rel` (relation), `property`, or `rev` (reverse) attribute by an RDF/A processor: the attribute indicates a new statement whose `predicate` is the URI value of that attribute. In the case of `rel` and `property`, the subject of the statement is decided by `subject resolution`.

---

21http://www.w3.org/TR/xhtml-rdfa-primer/
In the case of rel, the object is decided by *URI reference object resolution*. In the case of property, the object is decided by *literal object resolution*. In the case of rev, the subject of the triple is decided by *URI reference object resolution* and the object of the triple is decided by *subject resolution*. If both rel and rev attributes are used within the same element, two RDF statements are generated.

Literal object resolution yields either the value of the content attribute or, if it is absent, the element content. The value of the content attribute is by default interpreted as a plain literal. The element content is by default interpreted as an XML literal.

The datatype attribute is used to specify a specific XML Schema datatype (cf. [http://www.w3.org/TR/xmlschema11-2/](http://www.w3.org/TR/xmlschema11-2/)). If it is present, the value yielded by literal object resolution is interpreted as an instance of the XML Schema datatype.

URI reference object resolution yields either the URI value of the resource attribute or, if absent, the href or src attribute. The resource attribute is only used to specifically communicate that the URI is not intended to be “clickable”, or if a href or src attribute is already present on the element for other reasons and does not refer to the intended object. It is strongly advised to use the href attribute whenever reasonable.

Subject resolution usually yields the URI value of the about attribute, or, if the RDFa element that includes the predicate attribute does not have an about attribute, the about attribute of the first ancestor element that has an about attribute. In the absence of an about attribute within scope, it yields the xml base of the element.

The OWL schema can be found at [http://www.metalex.eu/metalex/YYYY-MM-DD](http://www.metalex.eu/metalex/YYYY-MM-DD), where YYYY-MM-DD is the date of the agreement, provided that one correctly sets the mime type, i.e:

```
GET http://www.metalex.eu/metalex/YYYY-MM-DD
Accept: application/rdf+xml
```

A description logic syntax rendering is found in appendix ??). The OWL schema specifies the concepts mentioned in section 5.3.3, and the citation metadata specified in section 5.3.4, as well as some related auxiliary concepts.

### 5.3.3 Individuation and Identification of Sources of Law

In Metalex, bibliographic entities are identified with URIs or URI references. Identification of bibliographic entities plays a role in:

1. Self-identification of documents;
2. Citation of other documents;
3. Inclusion of document components.

A fourth important purpose for knowledge engineering is that these URIs function as stable target for represents links between source and the institutional objects – legal rules, facts – we assume to be represented by them.

Note that URI references that are relative can resolve to different URIs dependent on XML base processing context.

Each bibliographic item encoded in the MetaLex CEN standard must have at least one URI. Manifestations must refer to the item URI by the URI reference "" (i.e. the
empty string URI reference\(^{22}\)). It is in principle not possible to encode item level information in the manifestation. It must be possible to establish the xml base of an item, in conformance with the XML Base specification\(^{23}\) and IETF RFC 3986 (or 2396). The concatenation of the established xml base and the id attribute of an element must result in a valid URI reference for the element, conformant to the addressing recommendations of W3C, which counts as a bibliographic identifier of the element as a bibliographic item. The are no further restrictions on item identification.

Note that if one uses an explicit xml:base attribute it remains the same after copying the document, which means that it also behaves as a manifestation level identifier. There are legitimate use cases of the xml:base, where it is inserted as a temporary identifier to an XML subtree in an XML processing pipeline.

The manifestation, expression, and work must also have at least one URI, which counts as their manifestation, expression, work level base, respectively.

Every MetaLex element must have an id attribute, not necessarily the metalex:id attribute. The id value of an element is a manifestation fragment identifier. Concatenated to the manifestation level document identifier it globally identifies the element at the manifestation level, concatenated to the expression level document identifier it globally identifies the expression embodied by the content of the element, concatenated to the work level document identifier it presumably globally identifies a structural element common to various expressions of the work.

Embedded metadata explicitly indicates its subject: if it is the document self-identifier (the empty string) "" or a fragment identifier "#f" composed of the document self-identifier and the value \(f\) of the id attribute of the intended target element, then the metadatum refers to the item. The only item-level metadata property is however the metalex-owl:exemplifies property, which takes the manifestation-level identifier as value (cf. section ??). The required type of subject of a metadata property is explicitly constrained in the OWL schema by a domain restriction.

XML attribute values by default pertain to the expression embodied by the manifestation, unless explicitly specified otherwise in this document.

Each bibliographic item exemplifies exactly one manifestation that embodies exactly one expression that realizes exactly one work. Because all these mappings are functional, i.e. unambiguously maps to one entity, item identity can be, and often is, used as an indirect identifier of the other objects, similar to how, for instance, email addresses usually have a functional mapping to persons and can be used as an indirect identifier of persons. One can for instance refer to a work by referring to its initial expression in a context where a reference to a work is expected.

The inverse of these relations is however often not a function. One can think of the work as an abstraction of 1+ expressions, the expression as an abstraction of 1+ manifestations, the manifestation as an abstraction of 1+ items. The manifestation, expression, and work are intentional objects whose existence is conditioned to the existence of at least one item, manifestation, expression, respectively. There is normally speaking no such thing as an expression that is not embodied, a work that is not realized, etc.

Besides the hierarchical constitutive relationships between the four levels, there are also horizontal relations between the objects within a level. The expressions of a work

\(^{22}\)Note that URI, which is absolute, and URI reference (cf. IETF 3986), which is absolute or relative, and can therefore be empty, are different. URI are globally unique, but URI references are not: only after resolution to a URI they are globally unique.

\(^{23}\)http://www.w3.org/TR/REC-xml/
in the legal field are usually either the initially published expression, or expressions
derived by content modification activity or translation activity. Manifestations of an
expression are either the initially created one(s), or manifestations derived by editing
activity. Items of a manifestation are either the initially created one(s), or copies of
them.

The Metalex OWL schema includes a number of event type definitions to make
these horizontal relations explicit for the expression level. At the item level they cannot
be embedded in a MetaLex item for obvious reasons.

Appropriate manifestation level version management methods and tools already
exist (CVS, SVN, etc.). These however, from the point of view of the source of law,
merely do manifestation-level version management, while most legislative events hap-
pen at the work and expression levels. Content-related events like markup, metadating,
and digital signature happen at the manifestation level.

The MetaLex standard aims to provide metadata for describing both the hierar-
chical and relational way of positioning and identifying bibliographic objects, at least
at the work and expression level, at the relevant levels of granularity.

Some additional terminology captures some bibliographic phenomena relevant to
law, as well as operational criteria for recognizing these in a MetaLex XML document.
Their specifications are found in the Metalex OWL schema:

- A bibliographic source of law is a bibliographic object that can be, is, was,
  or presumably will be referred to, by way of bibliographic citation, to back an
  argument claiming the existence of a legal rule in a certain legal system, or, alter-
  natively, a bibliographic object published or realized by a competent legislator
  to communicate a legal rule to a certain group of addressees. Both the legislator
  and the user of the bibliographic source of law understand it as a medium used
  for communicating the existence of legal rules, including auxiliary declarations
  required for the proper understanding of legal rules, between legislator and user.

- An initial version of a bibliographic work is the expression that realizes the work
  at the time of its official release in the public domain as a bibliographic object. It
  is the metalex-owl:result of an metalex-owl:InitialBibliographicCreation.

- A version of a bibliographic work is either the initial version of the work, or an
  expression realized by modification of a version. It is the metalex-owl:result
  of a metalex-owl:BibliographicModification of the (metalex-owl:matter)
  previous version.

- A version in force of a bibliographic work is a version that is, was, or will be in
  force during a specific time interval. The in force time intervals of versions in
  force of the same work do not overlap.

- A consolidation of a bibliographic work is a version realized by the execution of
  legal rules found in another bibliographic source of law to the previous version.
  It is the metalex-owl:result of a metalex-owl:BibliographicModification of the
  (metalex-owl:matter), which is the previous version, by the (metalex-
  owl:instrument), which is the other bibliographic source of law stipulating the
  modification, usually when it enters into force.

- A variant of a version in force of a bibliographic work is an expression that shares
  its in force time interval. Although the concept variant is often nominalized, it
  is a symmetric relationship (metalex-owl:variant) between two expressions.

24See the appendix on versioning for details.
• A language variant of an expression of a bibliographic work is an expression that shares its in force time interval, and differs in nothing but language. For instance, the English, Dutch, Italian, and German versions of a European directive are different language variants. It is also a symmetric relationship (derived on metalex-owl:variant). In a MetaLex manifestation of the expression this is expressed in different values of the xml:lang attribute.

• A translation of an expression of a bibliographic work is an expression that shares its in force time interval, differs in nothing but language, and has been realized by way of translation of one expression into another expression. Translation is an asymmetric relation between bibliographic expressions, expressed by a metalex-owl:Translation event, which has a metalex-owl:translator, taking the initial expression as a metalex-owl:matter, and the translated document as metalex-owl:result. The translation should not be confused with language variant: while language variants can be realized concurrently by the legislator, and are equally authoritative if they are, the translation of an expression is generally speaking less authoritative than the expression it is a translation of, even if officially translated.

The list above provides a number of interesting relationships between sources of law mediated by events. For the purposes of explanation of both the use of metadata in MetaLex, and the minimal identifying set of information, we focus on the identifying set as determined by the naming convention. This set is obviously normative, and not descriptive, and aims at URI references that:

1. can uniquely identify sources of law, regardless of jurisdiction and legislative technique;
2. can be reconstructed with a high degree of intercoder reliability; and
3. do not depend on a vantage point in time.

The expression and the work must be declared in case of non-conformance to the naming convention which is not discussed here. Noting that the URI reference about="" refers to the document itself, the following declares a standard manifestation, expression, and work base (using the naming convention):

```
<meta id="m1" about="" rel="metalex-owl:exemplifies"
    href="/tv/act/2004-02-13/2/tv"/>
<meta id="m2" about="/tv/act/2004-02-13/2/tv" rel="metalex-owl:embodies"
    href="/tv/act/2004-02-13/2"/>
<meta id="m3" about="/tv/act/2004-02-13/2" rel="metalex-owl:realizes"
    href="/tv/act/2004-02-13/2"/>
```

The RDF reading of m1 is as follows: m1 is a statement that states that the (referent of) metalex:exemplifies of (the referent of) (empty string) is (the referent of) /tv/act/2004-02-13/2/tv.

Read for meta in the examples above any appropriate element that permits metadata attributes. The URIs are relative, in this case conforming to the naming convention: the base is set by the processing environment. This means that the mURI of the naming convention describes a URI reference that potentially resolves to large set of URIs for each bibliographic object: one for each processing environment that sets its own base.
Note that if one wants to conform to the naming convention but also want to embed another identifier, this is trivially achieved with a metadata statement of that impact, for instance:

```
<meta id="m" about="/tv/act/2004-02-13/2/tv" rel="owl:sameAs"
    href="http://foo.tv/123456" />
```

This simply states in OWL that \( c_1 = c_1 \), where \( c_1 \) is /tv/act/2004-02-13/2/tv and \( c_2 \) is http://foo.tv/123456.

If the naming convention is not used, a set of metadata must be available, either in the form of RDF/A statements, or in the form of RDF. Let \( \langle\text{ManifURI}\rangle \) be the identifier of the manifestation, \( \langle\text{ExpURI}\rangle \) be the identifier of the Expression, \( \langle\text{WorkURI}\rangle \) be the identifier of the work, all declared as in the previous section. Datatyped values \(<\text{xsd:date}>\), etc. are values conforming to the XML Schema datatype \(<\text{xsd:date}>\), etc. The relevant set of metadata for the work is the following:

```
<meta id="w1" about="\langle\text{WorkURI}\rangle" rel="metalex-owl:resultOf"
    href="\langle\text{WorkCreationEventURI}\rangle"/>
<meta id="w2" about="\langle\text{WorkURI}\rangle" property="metalex-owl:workClassID"
    content="\langle\text{xsd:strings}\rangle"/>
<meta id="w3" about="\langle\text{WorkCreationEventURI}\rangle" rel="metalex-owl:date"
    href="\langle\text{CreationEventDateURI}\rangle"/>
<meta id="w4" about="\langle\text{WorkCreationEventDateURI}\rangle" property="metalex-owl:xsdDate"
    content="\langle\text{xsd:date}\rangle"/>
<meta id="w5" about="\langle\text{WorkCreationEventURI}\rangle" rel="metalex-owl:country"
    href="\langle\text{CountryURI}\rangle"/>
<meta id="w6" about="\langle\text{CountryURI}\rangle" property="metalex-owl:countryCode"
    content="\langle\text{xsd:strings}\rangle"/>
<meta id="w7" about="\langle\text{WorkCreationEventURI}\rangle" rel="metalex-owl:issue"
    href="\langle\text{IssueURI}\rangle"/>
<meta id="w8" about="\langle\text{IssueURI}\rangle" property="metalex-owl:issueID"
    content="\langle\text{xsd:strings}\rangle"/>
```

The main purpose of the identifying set is to create a non-ambiguous sequence of work creation events for each work class. Work class is identified by a descriptive string (e.g. "act" for act of parliament, or "municipal;foobartown;decree"). It is recommended to identify work classes with specific publication channels that can be monitored. Work classes are specific to countries, identified by the two or three characters from the ISO 3166-1 standard. The combination of the date of the event and the issue number (some string) uniquely positions the work in the temporal sequence of works belonging to that work class.

The relevant set of metadata for the expression is the following:

```
<meta id="e1" about="\langle\text{ExpURI}\rangle" rel="metalex-owl:resultOf"
    href="\langle\text{ExpressionCreationEventURI}\rangle"/>
<meta id="e2" about="\langle\text{ExpURI}\rangle" property="metalex-owl:lang"
    content="\langle\text{xsd:language}\rangle"/>
<meta id="e3" about="\langle\text{ExpressionCreationEventURI}\rangle" rel="metalex-owl:date"
    href="\langle\text{CreationEventDateURI}\rangle"/>
<meta id="e4" about="\langle\text{CreationEventDateURI}\rangle" property="metalex-owl:xsdDate"
    content="\langle\text{xsd:date}\rangle"/>
```
The <<CreationEventDateURI>> may and will often coincide with the creation of the work in unversioned documents. In case of an ex tunc modification the creation event is a FictionalExpressionCreation, and it is a temporal fiction of some other event, to be filled in, which happened at another relevant date:

The temporalFictionOf property is for us simply a kind of constitutedBy. The relevant metadatum for the manifestation is the following:

References in MetaLex documents are usually made with the href or src attribute. Because the href and src attributes are also RDF/A attributes the referring element also encodes a metadatum if an about and rel attribute are present. Using this form – the external reference as a metadatum – reflects the point of view that resolving citations etc are acts of interpretation of the source.

The MetaLex standard distinguishes three forms of addressing:

1. Reference to external objects (agents, events, etc): these are always encoded as metadata, and if embedded, always using the href or resource attribute.

2. Citation of other bibliographic objects: this can be achieved through the citations attribute group, or through metadata, always using the href attribute. The target should either be identified by use of the naming convention, or by identifying metadata. In some cases the citation can be embedded in both attribute form and embedded metadata, because of the double interpretation of href.

3. Component inclusion: this can be achieved through the src attribute and either conformance to the naming convention, or additional metadata. The inclusion can also be made into a metadatum.
The distinguishing property of references is that the (rel) value is a subproperty of `metalex-owl:refersTo`. Citation uses a property value that is a subproperty of `metalex:cites`, which is itself a subproperty of `metalex-owl:refersTo`.

This means that the fact that some MetaLex XML element refers to another entity can also be stored in RDF, external to the MetaLex XML document.

A reference is something that refers to or designates something else, or acts as a standin for a relation between two things: the referrer and the referent. Since a relation can also be identified, the generic form of a reference is \((\text{referrer, predicate, referent})\), where predicate is the name of the relation, and are represented as RDF or RDF/A.

In the sense intended here a reference is an XML element (directly or indirectly) containing text, and the text is deemed to refer to something else. The XML element will typically be of the inline content model type. A citation is an expression that refers to something intralinguistic, i.e. to another XML element (directly or indirectly) containing text, or to the bibliographic (work or expression) objects directly or indirectly embodied by it. Other references refer to something extralinguistic, i.e. something other than text, recoverable from the context in which the document was produced.

1. Article 1, the first article and the previous article are examples of citation, and the Minister, the President of the Republic, the accused, and We, Beatrix, etc. are examples of relevant references to other things. In a more general sense, any term refers to a corresponding concept. It is good practice to let references to individuals refer to a nominal concept (i.e. the singleton set).

A similar distinction, that should be distinguished from the previous one, is the distinction between exophora and endophora in linguistics. Take the following sentences:

1. Theft is the unlawful taking of a good wholly or partially belonging to another.
2. It (The theft) must have been done with the intent to appropriate.

\(\text{It (or The theft)}\) obviously refers back to Theft in the previous sentence. It is an endophoric proform expression, as opposed to exophoric (That must have been done intentionally, pointing to an act of taking in progress). Exophoric proform expression-sexophoric proform expression are obviously rare in legal resources.

There is however a difference between the reference \(\text{It}\) referring to theft as defined in the previous sentence and \(\text{the previous sentence}\) as referring to a sentence. Theft and it are coreferents of something other than text: it is not an intralinguistic reference. Expressions like it do stand in for another expression (Theft), but only to indirectly reference meaning recoverable from context.

They are used to avoid repetitive expressions and in quantification (i.e. carrying a variable from one sentence into the next one). It is positive evidence that:

- that the sentences refer to the same thing, and therefore share a concept in the same institutional reality; and

- maybe even that they may be taken together to represent a single rule.

This only applies to sentences close together. One can maintain the principle that a rule or fact is represented by a single fragment of text, i.e. structure isomorphism from text to representation, but the intended fragment would be the immediate container of both sentences. The implied thesis is that the immediate container is in this case the smallest fragment of text that could be retrieved as a justification of a rule: if
The following is an example of a reference metadatum in RDF/A:

```html
<meta about="#x" rel="metalex-owl:refersTo"
```

The use of references is optional. The src attribute may not be used.

A citation in legislation normally refers to a work, unless explicitly noted otherwise. A reference to legislation in a court verdict necessarily refers to an expression. Citations in other documents can refer to any level.

The reference to a text fragment is taken as evidence that the represented rule is about the rule represented by the other text fragment. Citations in legislation typically do one of the following things:

- restrict (or extend) applicability of another rule;
- when referring to something that is taken as a definition of a term; indicating that a term used here refers to the term as restricted in the other rule;
- justify the present rule; either by indicating where legislative competence comes from, or an obligation to implement something, or the permission to do so, etc.

A reference with a single target (e.g. article 1 of the Income Tax Law or article 15.3) is easy to mark up. The entire referring text can be marked, and it should be linked to the one concept that is being referred to. This can be either a work or an expression, depending on whether or not version information is specified or hinted at.

References with multiple targets are more complex to mark, as we now have several concepts to link to (i.e. article 5 and 6 of the Income Tax Law). An important problem is where to anchor these links. In addition there are references to ranges, such as articles 5-10 of the Income Tax Law. This reference will have several targets, and without studying the target document we cannot determine the exact target locations. It might be article 5, article 6, article 7, article 8, article 9, article 10 or article 5, article 6, article 7, article 7a, article 7b, article 8, article 10 or even article 5, article 10. The exact targets cannot be included in the document, as they cannot be derived from the information present. In addition, the targets may change depending on the version being referred to in case of references to works.

Ranges get even more complex when exceptions are involved, such as articles 5-10 with the exception of article 9. An interesting variation occurs when an exception is made to something other than a range: article 5 with the exception of the first member. This implicitly turns article 5 into a range of member 1, [..] member n.

A final format for references is the “each time”: articles 10, 12, 15 and 16, each time the first member. First member by itself cannot be a reference to something here. The links may not be to the articles but instead must be to the first member of each article.

The advantage of special markup for exception and each time constructs is that it is a hint to (less refined) parsers that discover and markup references that this one has already been discovered, interpreted, and discarded.

Each of these complications show the same pattern: there is a difference between the entities explicitly cited in the text, and the ones it refers to. In some cases an element must be explicitly cited to convey the information that it is not being referred to.
Complex references create two problems for the MetaLex standard:

1. There are multiple reasonable ways to mark up the same statement. Some XML standards for instance uses to start and end range attributes on an element enclosing both start and end of the range. Other standards have two separate elements marking the start and end, each using the same href (or similar) attribute.

2. The encoding of a metadata about a complex reference will necessarily require more than one metadata statement, and therefore need more than one carrier element if RDF/A is used. This is a problem if the aim is to declare existing XML documents as MetaLex compliant documents without adding new XML elements.

There are two methods for describing references. The following is an example of a citation metadatum in RDF/A:

```xml
<meta about="#x" rel="metalex-owl:cites" href="http://gov.tv/tv/act/2004-02-13/2/tv#x">
```

Other available properties (metalex-owl:citeFrom, metalex-owl:citeUpto, metalex-owl:excluding, metalex-owl:excludeFrom, metalex-owl:excludeUpto) can be used for complex references if the information is stored in RDF or if RDF/A is strongly preferred and modifying the XML document is no problem. The href attribute must be used.

Alternatively the metalex:citations attribute group can be used, if adding elements to the original XML file is a problem but embedding the references inside the document is nevertheless strongly preferred, with the attributes metalex:href, metalex:upto, metalex:exclude, metalex:excludeTo to specify two types of ranges from metalex:href to metalex:upto, and from metalex:exclude to metalex:excludeTo. These can be automatically translated into RDF statements.

If the naming convention is not used, the target of the citation must be identified in accordance with the principles set down in section 5.3.3. The use of citations is optional.

Some XML manifestations may include components by reference instead of by physical inclusion as a fragment, i.e. the inclusion reference replaces a part of the MetaLex manifestation stored in an external object identified by a URI.

Any part, except the top level container, of a standard metaLex XML manifestation can be implemented as an inclusion reference to an external object. On the manifestation level you make choices about, for instance, object names and media formats (tiff, jpeg, pdf, etc.). In some cases a text that is (or could be) embodied by a metalex manifestation (a chinese appendix of a treaty) is embodied alternatively by a media object.

For this the metalex:srcatt attribute group is provided, which defines the metalex:src attribute. Presence of the src attribute includes a component of the MetaLex manifestation stored in an external object identified by the URI that is the value of the src attribute.

While not technically necessary, the existence of components can also be easily describes in the form of RDF/A metadata simply by adding the property metalex-owl:component in the rel attribute. The following is an example of a component inclusion metadatum in RDF/A, superimposed on the metalex:src attribute:

```xml
<meta id="x" about="#x" rel="metalex-owl:component">
```
If the naming convention is not used, the target of the inclusion must be identified in accordance with the principles set down in section 5.3.3. The `metalex:src` attribute must be used if the `metalex-owl:component` property is used.

While reference pertains to non-bibliographic objects, and citation to works or expressions, components are typically manifestation-level objects. None of these mechanisms is however ever used to locate items, even though the used URI and URI references have the form of URLs and URL references. The reason for URLs is a pragmatic one: they are easy to obtain from registrars, cheap to lease, and many people have access to a domain they can consider their own.

### 5.4 Knowledge Components and Sources of Law

Central to reusability in legal knowledge engineering is the notion of a knowledge component, in particular the knowledge component representing a source of law. The knowledge component is stored in the form of a set of logical sentences, in no particular order. An OWL DL knowledge component contains only OWL axioms, while an autoepistemic knowledge component consists completely of autoepistemic sentences. Ideally, for the purposes of agent reasoner maintenance, one maintains a one-to-one mapping between knowledge components and the sources of law it represents, in order to:

1. facilitate maintaining isomorphism between source of law and knowledge representation over time, and to
2. be able to make a single change to all deployed agent reasoner that use the same source of law at a single point.

Single point maintenance is based on the assumption that the agent reasoner that use the component are able to import the new knowledge component whenever it changed, that one has reasonable assurances that the change, if minor, does not introduce logical inconsistency, and that the epistemic competence of the involved agent reasoner remains intact, or, in the case of major changes, that the effort involved in impact analysis is minimized.

This expectation of localized impact is however based on the assumption that the institution’s design rationale for sources of law and the knowledge engineer’s design rationale for knowledge components are aligned. One notable difference is that the legislator does not distinguish between an axiomatic part and an autoepistemic part of the source: most sources of law correspond with two knowledge components, and the autoepistemic one always uses the OWL DL one.

The source of law at the expression level is identified by a URI, refers to a set of URI-identified terms (`metalex:refersTo`), and it represents a set of URI-identified legal rules and legal facts (`metalex:represents`). This applies to the source of law as a whole, and to any URI-identifiable fragment of the expression. Since the expression does not change, one would expect that the set of legal rules and facts it represents and the set of terms it refers to also normally do not.

The referred terms are usually explicitly present, but in text fragments it is often necessary to first recover the targets of endophoric proform expressions from the context of discourse; This is clearly a representation step, which involves a choice between viable candidates from a linguistic point of view. In addition, a source of law
may use two terms, that are the same from a linguistic point of view, but turn out to be different from a conceptual point of view: ontologically speaking they cannot be the same. It is therefore possible that this interpretation changes over time while the source expression remains the same.

These terms can be categorized into those that are in the institutional reality created or modified by the legal rules, and those that are relevant to it but belong to brute reality from the point of view of the institution. These legally relevant terms may be institutional ones which belong to a separate institution whose existence is merely recognized by the institution the source of law applies to. Specific patterns involving the \textit{constitutes} property separate these categories of terms in the knowledge representation.

Obviously multiple sources of law may be about the same institutional reality, and an explicit marker that this is the case is a desirable feature of legal institutions. One may also be of the opinion that legally relevant terms used in the source of law are the same as those used in other sources of law.

Since the same entity can be identified by multiple URI, it is possible to separate these interpretations from the codification of the terminology itself. All terms identified in a source of law expression are identified by a URI, for instance \( n_1 : C_1 \) (where \( n_1 \) is the namespace part of the URI), that only identifies the term as it occurs in that source of law expression. All terms identified in a source of law are in the same namespace, and no other terms are in that namespace. The integration of terminologies from different source of law expressions is made explicit in the form of equivalence statements \( n_1 : C_1 \equiv n_2 : C_2 \).

The most obvious such scenario, and at the same time the most complicated one from a maintenance point of view, is localized change of a work: the set of terms in the new expression for a very large part overlap with the previous expression of the work. One might think of this as the shared set of terms at the work level, but the shared work level set only exists from a specific vantage point in time, or only once the source of law has become immutable after its repeal and the shared work level set – which can no longer change – has become irrelevant.

A pragmatic solution to reduce the number of such terms is to identify terms by the \textit{expression} that created (first referred to) them in that work, and only to renew it if a new expression changes it by changing the set of OWL axioms that use it. This however creates a potential problem in relation to equivocation with terms from other works. It may happen, in rare cases, that institutional realities are split up.

Recently in Belgium, certainly areas of legislation, including the existing works, have for instance be delegated to a lower level. What was one federal institutional reality, has been split into three: the terms that remain on the federal level and retain a single meaning, and those that are now managed by the Flemish and Walloon community, respectively, and whose meanings now diverge.

This is however solved with the presumption that, with a change of author, the work is republished. This however requires a temporal fiction: for the correct application of lex posterior one has to presume that the community government is now the original author of all changes in the original timeline before the delegation happened. From a MetaLex point of view, it is certainly preferable to consider the delegation only a delegation of the power to change the source of law in the future, and to only relabel terms as soon as a divergence happens.

The legal rules and facts, together legal clauses, follow very similar rules. All clauses identified in a source of law expression are identified by a URI, for instance \( n_1 : r_1 \), that identifies the clause as it occurs in that source of law expression. All
clauses created in a source of law are in the same namespace, and no other clauses are in that namespace. A new URI is assigned only if the clause changes in a new expression of a work.

Do note however that the source of law on the expression level cites other rules on the work level, while the legal rules we represent are necessarily identified by their first representation in an expression, and not on the work level. This creates a problem for legal rules about other legal rules: the logical representation changes not only when the rule is changed, but also if the rule it is about is changed. The maintenance problem this would create is easily addressed by using the notion of embodiment between expression and work in institutional reality. We may need to say that legal rule \( r \) for instance claims that some set of rules represented by expression level text fragments that embody work fragment \( w \) apply to \( C \).

The notion of embodiment applied in institutional reality is not very elegant, but it is the legislator itself that is the origin of confusion between institutional reality and the texts that prove its existence. It is this kind of observation that has led to the adoption of the FRBR distinction between works and expressions in MetaLex. The applicability patterns in section 5.2.3 are thus overly simplistic and have to be modified to accommodate embodiment, which could however only be introduced after the discussion of the source of law qua document and MetaLex.

Legal clauses are identified by OWL constants: it is absolutely essential to state that each clause is different from all other known constants in contexts where there is a risk of unification of constants. Both legal clauses and terms, which are OWL concepts, are always denoted by URI in OWL by default. Logical sentences are not. Note that it is not customary in OWL DL to explicitly assign a full URI to OWL axioms. They do however always have one, derived from the xml base of the document they are stored in, which will by default function as namespace, and they can be explicitly reified to assign one. The logical sentences are not found in the source of law: they are created by the knowledge engineer. The logic of grouping their identifying URI into namespaces, and the sentences themselves into files, and doing version management on those files, perhaps by making a version number or publication date part of the namespace identifier, remains solely a knowledge engineering concern, limited only by the agreed upon update mechanism.

5.4.1 Knowledge Engineering Considerations

The knowledge engineer must keep in mind why he is making a change in an OWL file that represents a source of law: is it because institutional reality changed, or because his interpretation of the sources of law changed? In the latter case, the change can be made without further ado. Representation of a source of law expression consists of two interpretative phases in which mistakes can be made:

1. Identifying the entities in institutional reality represented or created (first represented) by the source; and

2. formulating the logical rules that describe the constraints on institutional reality and the interfaces by which institutional reality can be changed, as evidenced by the sources and the way they are used.

Case law has an important role in our increased understanding of law over time, and at the same time creates ambiguity because it functions both as a source of information on the meaning of a source of law, and as a source of law itself. It
can change our interpretation of institutional reality, but it also changes institutional reality by representing new rules.

A specific goal of this book is liberating knowledge engineers from the self-imposed straightjacket of trying to represent a legal rule by a logical rule. The logical rules that can be impacted by a change of the source are simply the ones that refer to the clauses and terms that were changed. In the case of legal rules, this use is always mediated through specific patterns of use of the applicable property.

Note that legal “facts”, represented as proposed in this book, are always rephrased to rules because they only have effect if applicable. Clauses which only postulate the existence of something are simply the first reference to an entity in institutional reality, and create it merely by referring to it.

The knowledge representation captures the entities that exist in the institution, including their temporal features, and the entities themselves are linked to the sources as evidence of their existence through the inverse of the refersTo and represents properties.

The goal of a methodology like this one is not only to retain descriptive fidelity in representation, but also to increase intercoder reliability. The patterns in this chapter in this sense function as a limited menu of options for representation that increases the chance that knowledge engineers using the same representation language will represent the same sentence in a source of law in a similar way.

A second instrument for increasing intercoder reliability is the use of core ontologies imported by the knowledge representation of the source of law, and to define terms in the source of law as subtypes of the terms in the core ontology.

The primary distinction between knowledge sources for the agent reasoner remains the distinction between descriptive ones, and defeasible ones. The descriptive part of the knowledge base, ordered from more reusable to less reusable from the perspective of representation of sources of law, consists of:

1. the core ontology; MetaLex and LKIF play this role in this book, although we have liberally deviated from LKIF;
2. the ontological constraints and requirements found in the sources of law, which introduce the relevant terms, subsumed by terms in the core ontology, which is the only thing imported by ontologies of sources of law;
3. a judicious set of subsumption or equivalency statements that connect terms describing institutional reality in the respective sources of law;
4. a judicious set of subsumption or equivalency statements that connect terms describing brute reality in the respective sources of law; and
5. a judicious set of subsumption or equivalency statements that connect terms in the source of law to relevant external theories – legal, abstract, commonsense, etc. – that are imported at this point.

The similar, bit shorter, list can be made for autoepistemic sets of rules:

1. the rules found in the sources of law, which appeal only to terms in the sources of law;
2. rules found in other sources of law, or customary in the involved legal system, that one cannot do without;
3. the rules of the relevant external theories – legal, abstract, commonsense, etc;
4. a judicious set of common sense rules that apply to brute reality in the respective sources of law, in particular those required by useful external theories (frame assumptions, etc.); and

5. agent reasoner-specific knowledge availability rules and constraints.

Although it is good strategy to separate the representation of individual sources of law from the integration of respective institutional and brute realities (as advocated in Boer et al. (2003b)), judicious sets of subsumption or equivalency statements, in line with current legal practice and case law, are obviously very valuable reusable knowledge components.

A lot of valuable advice on integrating ontologies, versioning ontologies, and pruning them to throw out unnecessary ballast for a specific KBS exists (cf. for instance Pinto et al. (1999); Klein and Noy. (2003); Noy and Musen (2004)). One should however be cautious with the idea of merging concepts often found in this literature: merging concepts instead of declaring them equivalent in itself never gains any computational efficiency. The case is different when two ontologies follow different approaches to doing essentially the same thing, i.e. follow different core ontologies: in this case real gains can be made by merging.

5.4.2 Applicability and Knowledge Sources

The isomorphic approach to representing sources of law is simple: take a source of law and represent it. Making an agent reasoner that exhibits a certain epistemic competence, for instance to determine one’s taxable income and the amount of income taxes due, or whether a certain plan for a shed is eligible for a permit, is a lot more complicated. In this case we want to know which (fragments of) sources of law are applicable to the problem; We want to represent only these sources of law, or if a complete repository for such representations exists, we want to know which ones to import.

Wouldn’t it be convenient if there was some sort of metadata set for knowledge sources representing sources of law that helps us to find the relevant ones for our domain of law?

We don’t have to look very far for a candidate solution: applicability rules do this thing for the law. The exact same problem is of course faced by anyone who tries to find the rules relevant to some predicament. Applicability rules are used to:

1. avoid having to repeat the same requirement a lot of times;
2. determine from when to when a rule can be applied, and from when to when the events must have happened to which it is applied;
3. demarcate the extent of one’s claimed jurisdiction over people, territory, and substance;
4. restrict the applicability of rules made by a lower legislator, often prior of delegation of legislative competence to the lower legislator; and to
5. make the application of one rule conditional on the application of another rule, or in other words, to force a choice between two rules.

All uses except perhaps the last one have a direct relevance to the problem of deciding whether a source of law is relevant to us, and it is especially convenient if we find them in the first section of the source of law. Applicability statements are
however found within a source of law and in other sources of law about the one we are evaluating.

The suggested representation in section 5.2.3 makes both the carrier of the applicability statement and its target explicit, so it is not hard to find them if one keeps proper track of the identity of one’s sources and legal rules. As noted in section 5.3.1 it is often the case that relevant metadata of one source of law is found in another one: one of the rationales for the design of MetaLex is to make this connection explicit.

Very often applicability rules do not apply to a specific rule, or the set of rules found in a chapter, or source of law. Mandate and delegation (cf. section 5.2.5) may link it to specific offices and legislative competences. For instance the provision the agency for legal knowledge engineering creates additional guidelines for the representation of legal knowledge creates the possibility of creating legally recognized rules for a newly invented agency (the agency for legal knowledge engineering), and restricts the applicability of those rules to the act of representing legal knowledge. It means in other words that the set of rules that is the result (metalex:result) of an act of the agency for legal knowledge engineering is restricted by the substance to which it can apply.

Applicability as a guide for clustering sources of law into domains of application works very well. The Dutch government search engine for legislation is based on tracking legislative mandates, and is able to show for any formal act which lower regulations are based on it. There are also ways of indirectly doing the same, for instance grouping sources of law by formal author; the agencies and offices that compose a government are generally created to exercise specific public competences, often including legislative ones.

In section 5.2.4 it was noted that legal fiction and also-applicability have the unfortunate consequence of requiring us to add an extra condition to other logical rules that we prefer to be immutable. Besides that it was considered a form of fraud in law by Bentham. Of course one could defend the point of view that the consequences for the proposed logical formulas is simply evidence for the non-adequacy of this representation, but the fiction has wider implications for the problem of finding the rules that apply to one’s predicament: it expands the context of applicability of other rules in counter-intuitive ways.

Adding non-indicative rules – the rules that restrict the valid models of the institution instead of the interface by which it is changed – should always have the effect of restricting the set of valid models. The fiction does the opposite: it expands the set of valid models. The correct way to do that is to retract rules, and replace them with less restrictive ones. The fiction can expand jurisdiction, expand the applicability of a source of law, and expand the applicability of a knowledge source representing a source of law.

The ex tunc modification of a source of law, which exists in countries which have a constitutional court that can annul legislation after it went into effect has a similar effect of changing the past, and retroactive applicability also does potentially (cf. section 5.2). In this case – because these are predictably patterned fictions that occur a lot – they have been accommodated by three layers of timelines:

1. the interval in which the occurrences to which a legal rule can be applied must
have happened;

2. the interval in which the action of applying the legal rule can happen; and

3. the interval in which a source of law has been known to exist (ex tunc).

These examples show that fiction patterns can be accommodated without accepting defeasible reasoning in the institutional ontology, but we need to engineer a special solution for each such fiction pattern. Arbitrary use of fictions is inconvenient from a knowledge management point of view.
Chapter 6

Modeling techniques for legal knowledge

Whether we are considering relevant case law, or stories about critical incidents in the organization, as they are shared in a user requirements meeting, we analyze failure cases as narratives involving agents and their interactions, as prototypical diagnostic agent-roles describing the rationale of certain actions (for instance fraud) from the perspective of the acting agent, and as argumentation frameworks connecting evidence to arguments, and counterarguments, about the actions and intentions of agents, and to appropriate remedies and repairs.

While argumentation frameworks are well known as a technique for the analysis of case law, and narrative frameworks have been used by others, we have added a number of innovations in the Agile project by situating the events and causal connections in a narrative in an underlying system, a multi-agent system, and addressing the function of case law and critical incident narratives as knowledge resources in a model-based diagnosis problem. This is an innovation in Artificial Intelligence & Law, and, even more so, in SOA.

6.1 Narratives

6.1.1 Operational definitions

Proposition 6.1.1. A narrative is a composition of connected events.

The strength of the connection between events ranges from simple discourse contingency (i.e. being in the same composition, and in a certain relative position) to contextual dependency, often implicit, determined by syntactic, semantic and pragmatic factors.

Usually such composition is performed only at epistemic level: it does not concern the events in themselves, but reified representations of them, i.e. facts about events.¹

¹This holds for written and oral forms of narration. A theatre performance would be instead an example of direct composition of events.
Events  The use of the term event is commonly used to refer both to ongoing change and to settled change. An event of the first type, or ongoing event, corresponds to the execution of a process and can be called also transient (or activity/action if performed by an agent).

Proposition 6.1.2. A ongoing event or transient corresponds to the execution of a process, that will bring about a change.

An event of the second type is related to the outcome of such execution; it is a transition or impulse event (act if performed by an agent). In the rest of this work, unless specified, the term event will refer to transitions rather than transients, and to acts rather than activities.

Proposition 6.1.3. An event or transition corresponds to some settled change.

The start of an event of the first type is an event of the second type. With such definitions, we can scope what kind of change we are referring to:

Proposition 6.1.4. Changes associated to events are: creation or destruction of objects, start or conclusion of transients, modification of properties, attributions or qualifications of objects or transients, enactment or abrogation of rules.

6.1.2 Ontological layers

Proposition 6.1.5. A narrative presents three ontological layers, always simultaneously present: the discourse, the story and the conversation, which are respectively signals, meanings, and relevant components of the social context—like knowledge and intents of narrator and listener—that concur to the generation and interpretation of such act of communication.

The same story can be reported with different discourses (changing words, expressions, order, detail). The specific choices of transformation from story to discourse (in generation) and from discourse to story (in interpretation), depends on the conversation layer. Fig. 6.1 briefly illustrates the dependencies between these three layers.

In our framework, when acquiring a case from an expert, we neglect the discourse and conversation layers of such act of narration. We target only its meaning—the story layer—making explicit the components of the context relevant for its interpretation. However, narratives often contains characters telling something. In respect to
these nested narratives, the main story includes aspects of their discourse (as order of messages) and conversation (e.g. position, knowledge, intentions of the characters) layer.

6.1.3 Functional perspective

Analyzing a narrative with a functional or organicistic perspective, i.e. as a system having parts that function in relation to the whole, we recognize four families of functions. The first two are distributive and express the sequential distribution of the events; the other two categories are integrative: not necessary for the sequence, but they increase the detail of representation for the sake of the interpreter.

- **primary/secondary functions** index events (including acts), critical or not in respect to the causal structure of the story.

  The primary, secondary qualification of functions depends to their criticality in respect to the story. For instance, considering the story of “going to a certain place”, taking a train or a car is not as critical as going to that place. At a first glance, this seems to be a matter of hierarchical decomposition of events. However, if some accident happens in the story, this would be still a primary function, despite of not being at the highest hierarchical level of description.

- **clues** are related to (a) qualifications or attributions of objects (including agents, i.e. objects provided with mentality) or events, (b) rules belonging to a certain system (physical, mental, social, legal, ..).

  In narratology, characters or roles are the most important clues: they connects events within the story as proactive agents (in practice determining their sequence), or as patients to other roles' actions (to which they usually react).

- **informants** describe properties of objects or events; informants that trigger rules are clues as well.²

6.1.4 Narratives through critical realism

Applying the critical realist framework on this analysis, we observe that all narrative elements (functions, clues, informants) are observations (belonging to the empirical). However, narrative elements concerning rules intend to describe some of the mechanisms of the world in which the story occurs. Such passage from observation to mechanism (from empirical to real) is a question of belief of the interpreter. A story is plausible if the interpret is able to align the events reported in the story with dependencies/mechanisms he knows or he acquires (and assumes as holding) while reading the story. If there were no dependencies and the interpreter was not able to learn any pattern, events would be just “white noise”, a series of chaotic elements.

As the empirical is a partial (and potentially wrong) image of the actual, a story is a limited container of knowledge about the world. Many (actual) events and (real) mechanisms are neglected while reporting the story, mostly because they are irrelevant, but also because they are assumed to be part of common knowledge.

²In respect to properties, qualifications and attributions necessarily need an evaluative framework in order to interpret them. However, on closer inspection, also properties like “being 13 year old” refer to units of measurement too, defined by a shared standard. This discrimination can be seen rather as a matter of “objectivity” or collective consent on such evaluative framework. Therefore, “13 years old” is at least an informant, while “a long distance” is a clue.
6.1.5 Informational interest

A story can be interesting for two different reasons: from an emotional point of view and from an informational point of view. In the current research, we neglect the first component, more related to story-telling. About the second component, once the interpreter receives a narrative, several informational consequences occur:³

- acquisition of what the story explicitly contains,
- abstraction to more generic patterns,
- elaboration with previously acquired knowledge,
- elaboration of adequate social responses.

Let us consider an example. If Paul tells us about how he made a certain cake, and if we consider him trustworthy enough, we are acquiring not only the fact that he actually did the cake and all those related facts, but we also receive elements on how anybody could make that cake. However, if, comparing those facts with our experience, we acknowledge that the recipe is wrong, we may think that Paul is finally not so good in cooking. The related informational consequences, expressed in the four categories presented below, would be:

- Paul has made a cake in that and that way, (direct content)
- anybody could make the cake following that recipe, (abstraction)
- from my experience, that recipe is wrong, (content evaluation)
- Paul should not be considered for cooking matters. (social evaluation)

6.1.6 Legal narratives

Proposition 6.1.6. A legal narrative is a narrative intended to make the point about the implementation or the application of law in a social system.

6.2 Argumentation

Argumentation is generally perceived as operating at a meta-level, concerned with support and attack relationships between claims, rather than between messages and explanations. In fact, argumentation theory treats messages directly as claims. Investigating this step through the narrative concepts presented above, we observe that claims are constructed using only the story ontological level, pruning the discourse and the conversation levels, i.e. considering the content not depending on how (when) it is provided, or on its contextualization. Residuals of this cut are for instance found in standard critical questions⁴ about witness or expert statements, concerning motives and position to know. They are an ex-post solution to take into consideration the intentions and the knowledge of the agents producing the speech acts in the argumentation.

³Despite the fact that we write them one after the other, we cannot assume a specific chronological or functional sequence between these mental activities. Alternative dispositions of the elements are plausible as well.

⁴See for instance Walton et al. (2008), Walton and Zhang (2012).
6.2.1 Argumentation as a narrative act

From another perspective, the whole process of argumentation can be seen as the generation of a collective narrative. Competing parties produce and receive others’ messages, which are sometimes collected by a third-party adjudicator, entitled to interpret them from a neutral position. Messages are probably the most common events in narratives, and practically the most relevant type in legal narratives: the events of a case do not occur in front of the court, they are always reported.

**Proposition 6.2.1.** A message is a speech act. It is the outcome of an act of communication performed by a certain agent, with a certain intent, and towards a certain audience.

The explanations of a case correspond to the possible scenarios or stories covered by its narration (including its argumentation). They are partial representations of the world, usually intended—at the conversation level—to follow a word-to-world direction to fit. The ingredients taken into account in constructing explanations are given at the discourse level, which corresponds to the whole observation of the interpreter/adjudicator: an ordered sequence of the speech acts performed by parties, witnesses, experts, etc.

**Proposition 6.2.2.** An explanation is valid if it reproduces the messages provided by the narration.

In the case of legal cases, if the outcome acknowledged the existence of a failure in the socio-legal system, this would be a diagnostic explanation, otherwise it would be a non-diagnostic explanation.

6.2.2 Argumentation and inference to best explanation

Because this process of reconstruction is a form of abduction, there are usually several valid explanations associated to the same observation. However, when interpreting a narrative, and still more when adjudicating a case, we are interested in determining what is the case. Valid explanations compete, and such competition is a matter of justification.\(^5\)

Instead of being a static entity, the best, i.e. the most plausible explanation is refined along with the narrative flow (as foreground), integrated with common and

---

\(^5\)In a few pages, we will distinguish epistemic from practical justifications. In this context, because we are aiming to the truth, we are referring to the first type.
expert knowledge about the world (as background), by deletion of possible worlds rather than addition.\textsuperscript{6} If at the beginning all explanations have the same probability, after some relevant message, their relative weights will change, distinguishing which, at that moment, is the best explanation. Using a metaphor, instead of being drawn, the best explanation is sculpted.\textsuperscript{7} Conversely, when messages transport conflicting claims, they support conflicting explanations, whose relative weight is consequently equilibrated. The matter becomes raw again. Fig. 6.2 illustrates this point.

\subsection{Claims as manipulations of the explanatory space}

In a classic Toulmin’s scheme, data require some warrant, in order to generate the claim. But, generally speaking, data and warrants are claims too. When the dissent opinion in \textit{Pierson vs Post} raises the argument concerning the ferocity of the fox, a different allocation rule is presented, it is instantiated as new warrant, and so generates new claims from the data shared in the court. This example shows how claims are made by agents to manipulate the explanatory search space ascribed to the recipient of the message.

Claims attack or support other claims depending on the effect they have on that search space. A claim has utility if it increases the probability of explanations that agree with one’s goals, relative to the probability of explanations that disagree with one’s goals. If one’s goal is cooperatively finding the best explanation, the best message is the most informative one in terms of the imagined common hypothetical explanation space. This is only an idealized conception: a pragmatic image of the argumentative process between enforcer and potential suspect, or plaintiff and defendant, is as a competitive game between parties. This game may be modeled as a game-theoretical exercise, based on an classification of explanations into won and lost positions, as we see in some approaches to argumentation theory. This is not necessarily the distinction between diagnostic and non-diagnostic explanations: a defendant may not only argue for non-diagnostic explanations, but also for diagnostic explanations that minimize harm. Consequently, we should not use the assumption of prior indifference between explanations for the players, but only for the adjudicator.

\subsection{Reasons and rules}

Argumentation theory provides a bridge between deliberation and agency, and specifically in the concept of reason. Several authors distinguish two kind of reasons: practical reasons, i.e. reasons for action and epistemic reasons, i.e. reasons for conclusions. Reasons assume one of those qualifications depending on what they support and, consequently, to their potential consequences: an act, or a fact.\textsuperscript{8}

\textsuperscript{6}We are not considering here messages introducing new mechanisms (e.g. agent-roles, rules), which multiply the space of explanations.

\textsuperscript{7}This is a famous interpretation of the practice of sculpture given by Michelangelo: the artist let figures emerge from the raw stone.

\textsuperscript{8}However, an act usually becomes also a fact, as consequence of some mental or perceptual feedback process.
6.3.1 Representation

Practical and epistemic reasons can be both represented as:

\[ \text{reason} \rightarrow \text{action} \]

In this formula, \textit{action} refers both to \textit{practical actions}, e.g. physical actions, or to \textit{epistemic actions}, like adding or removing a fact (a belief in the case of an agent).\footnote{Someone may argue that \textit{epistemic actions} are not really intentional actions, because they belong to cognitive modules mostly not directed by intentions. For instance, we don’t forget at command. However, from the perspective of an organization, adding a “belief” means adding an element or more to its databases. We could say that the intentionality of keeping track of such facts is implemented by design, through procedural rules.}

For instance: “if it rains, then I take an umbrella”, “if my doctor says I’m in a good shape, then I believe I’m in a good shape”. From these examples we observe that the conditional operator seems to model a causal dependency rather then a generic relation: when applied, it produces an outcome. In expert systems, such figure is called \textit{production rule}. The previous form was in particular a \textit{condition-action} rule.

An alternative form of production rules is given by \textit{event-condition-action}. The reason triggering the rule is composed by an event, i.e. a certain change that has been captured (a transition) and by a certain context (whose components are static or transient):

\[ \text{immediate reason} \land \text{contextual reasons} \rightarrow \text{action} \]

Extending the previous example, we have: “if I’m about to go outside, it is raining, I want to avoid to become wet and I don’t have an umbrella on me, then I take and open an umbrella”.

6.3.2 Planning and projection

In the end of the previous paragraph, we provided an example integrating an \textit{agentic} perspective, i.e. based on an intentional stance. We recognize intents (e.g. “I’m about to go outside”), desires (e.g. “I don’t want to get wet”), and beliefs (“I don’t have an umbrella on me”). Is there a stronger relation between reasons and agents?

\textbf{Practical reasons and planning}

Intentions are reasons for action (“I do this, because I want that”), but also \textit{reasons for intention} (“I want this, because I want that”). We therefore add to the previous account a set of \textit{volitional actions}, concerning the commitment and the dropping of intents. For example, “if I’m about to go outside, it is raining, I want to avoid to become wet and I don’t have an umbrella on me, then I want to take an umbrella and open it”. This passage allows the reference to nested intents. For instance, “when I want to take an umbrella, and I don’t know where it is, I want to find it”; “when I want to find an umbrella, I look around my house for it”; “if I see an umbrella and I want to take it, I take it”. The last rules refer to practical actions as well. This is a general scheme: a reason for certain action can be translated as a chaining of reasons for \textit{intents}—i.e. for commitment to goals—and reasons for actions.
constructing that targeted action. This is basically the same idea used in planning via backward chaining.\(^{10}\)

As a consequence of this account, alternative actions (or courses of actions) may respond to the same event+condition configuration. For instance, “if I’m about to go outside, it is raining and I want to avoid to become wet, then I wear a raincoat”. In this case, it would be strange to trigger all the available rules (e.g. to take the umbrella and the raincoat): one would be sufficient.

**Projection and reaction**

Projection usually plays a role when something happens in the environment or there is a change of intents. It is a basic mechanism of the agent that supports and constrains the construction of new objectives, adequate to the occurred change. Projection is then just half of the story. There is also another step aiming to identify the response to such event: the reaction. For instance, if the seller receives an acceptance, he expects that the buyer will pay. His normal reaction would be to deliver the good to the buyer and possibly to check the payment. Unlike planning, projection and reaction have mainly a forward chaining flavour. The same fact could entail in the agent different independent consequences. Once the reaction sets the main objective, however, planning operates the usual derivation in a specific course of action.

6.3.3 Justification

**Justification for practical and epistemic reasons**

We observed in the paragraph about planning that one solution is sufficient to solve the problem. In a general case, however, we are not satisfied with the first available solution, but with the best solution within a set of possible solutions, according a certain evaluative function. In this respect, we distinguish optimal solutions—to be found analytically—from sub-optimal solutions, to be found relatively to a partial set of possible solutions. The selection of such elements is usually synthetic (via heuristics), and their evaluation can be analytical or synthetic.

There is an interesting connection between best solution in practical terms, and best explanation in epistemic terms. The model-based diagnostic process we present in this work consists of generating all possible explanations, and of assessing their relative weight, depending on certain grounds, which include observations and assumptions. It is easy to observe a strong correspondence with the previous description of planning. As a matter of fact, the quest for an explanation corresponds to an epistemic goal.

There is however an intrinsic difference. Given certain circumstances, practical reasons are usually associated to problems like “what is the best action to do, in respect to certain goals/values?”; while epistemic reasons refers to “what it the best explanation, in respect to what we know?”. The first presuppose some definition of good, or of value, while the second track truth. We distinguish therefore a practical justification, which depends on such definition of value, from an epistemic justification,

---

\(^{10}\) Given a target objective, we consider, up to our knowledge, which sub-objectives we should obtain in order to reach it from the actual circumstances, and recursively we do the same with those. The lowest hierarchical level of this recursion corresponds to objectives which are actions. The final result consists of a set of alternative courses of actions, i.e. plans, which bring about the initial goal.
that allocates a relative measure of strength to a certain explanation. In both cases, however, justification allows to settle a decision.

Justification for projections

In the previous example we overlooked an important aspect. “I want to avoid to become wet” is a declarative goal catalysed by some related temporal projection, e.g. “if I go out and it is raining, I expect I will get wet.” This line of thought is backed up by a causal dependency assumed to hold in the world.

In relation to the world of experience, however, projection is very rarely a deterministic process. Starting from an observation (always partial to a certain extent) of the current context, we know that many unpredictable events may occur in the future. Constructed by heuristics on direct and indirect experiences, causal dependencies are often defeasible. We are in the domain of ontological uncertainty, that can be modeled, for instance, with probabilistic methods. Interestingly, the internal structure of inferring a projection turns out to be not so different from that of giving an explanation.

In Fig. 6.3 we illustrate this point. On the left, starting from an observation of certain events, we infer possible explanations of their causes. Only one of them is selected via justification, and that counts as our interpretation of what occurred. Along with the interpretation, we have accepted certain underlying causal mechanisms; at this point, we could optionally test if, starting from our reconstruction, they actually reproduce the starting observation when executed, so as to validate our reconstruction. On the right, we start again with an observation, but in this case we aim to produce a picture of what may occur, starting from the observed facts and partially known mechanisms of the world. The causal dependencies we consider are non-deterministic, for instance defined in probabilistic terms. Several projections are therefore possible, some of them will be plausible, and few of these will be more probable. The analysis of the alternatives is constructed via some process of justification and selects the most plausible/probable projection, also called prediction.\footnote{For visual completeness, we close the cycle also in this part of picture, although this is less intuitive than in the previous one. Any projection goes with a specific causal dependency, which inverted gives an evidential dependency. Therefore, starting from a projection and such dependency, we can infer which context causes it: if this re-presents the initial observation, our projection is \textit{a priori} valid.}
Reconstruction (or justified explanation) and prediction (or justified projection) are basically two actions that humans perform in order to simplify the burden of information and information processing: *because* we accept that this fact occurred, we are able to understand other related facts, *because* we think that something will plausibly happen, we can choose a certain conduct, in order to pursue our intents.

### 6.3.4 Obligation and power as reasons

#### Obligation as practical reason

Let us take again the example of a sale. The reception of an acceptance by the seller can be modeled as a *reason for conclusions*—attributing to the buyer/seller the *obligation* of paying/delivering— whose secondary outcome is a a *reason for action*—delivering and monitoring/listening to the payment.\(^\text{12}\) Generalizing this example, we can draw the following principle:

**Proposition 6.3.1.** An obligation is fundamentally a prototypical reason for action for an agent.

The mediation role that obligations play between facts and actions explains their double nature of descriptive and prescriptive entities. Similar considerations hold for all legal concepts present in the normative square, i.e. the first Hohfeldian square. A peculiar case is the concept of *claim*, which usually require some monitoring of the demanded performance.

**Proposition 6.3.2.** A claim (or obligatory right) is a prototypical reason for “listening” to a specific outcome.

#### Power as epistemic reason

The second Hohfeldian square is about *power* (or *ability*) of changing jural relations. Differently from obligations, the concept of power plays a role at epistemic level. For instance, in the case of *practical power*, agents take a behavioural decision in respect to what they think they are able to do. If they act, they usually have some assumption about their power, backing their action. For physical actions, these assumptions are constructed through learning. In case of legal institutional actions, power is defined through law.

**Proposition 6.3.3.** The power-related concepts have an effect at epistemic level.

### 6.3.5 Subjective, objective perspectives

Another distinction is between *agent-neutral* and *agent-relative* reasons, i.e. between reasons that always apply, and others that depends on the individual disposition of the agent. Abstractions like mathematics are for instance based on *agent-neutral* reasons.

\(^\text{12}\)Interestingly, under a certain perspective, all direct actions can be mediated through obligations. We know that an action is performed when an intention has been formed in such respect. An intention is a commitment: we do not know where and how it has been generated, but it exists. When an intent is activated, however, it is like the individual has an “obligation”, towards himself, to fulfill it. As he can drop the intent, however, he has also the power to drop such an “obligation”.
A similar distinction separates motivating from normative reasons. The first are reasons related to the motivational context of the agent: from the individual perspective of the agent, they imply or deny his commitment to a certain action or fact. Conversely, normative reasons require or prohibit a certain action, or imply or deny a certain conclusion, according to general rules. The difference is in where such rules come from: internally, or externally.

Such conceptualization is however simplistic in one point. The implication of certainty or impossibility presupposes an analytical representation of reasons and of the rules that relate them. This is however an abstraction, in respect to human socio-legal systems: the lacunae of law, the constructive nature of such systems, and the consequent synthetic nature of many of the concepts used in law will always undermine any attempt to create a complete analytical representation of the legal system. Much of the debate between legal realists and legal positivists can be reduced to the position normative reasons occupy in their conceptualization. A radical positivist approach claims that they are agent-neutral and that legal truth is just a matter of substantial procedures. A more pragmatic approach would say that they should be as far as possible agent-neutral, but they are mostly based on justification. A radical realist position would claim they are always agent-relative.

6.4 Multi-agent systems

6.4.1 Agent-roles
Rationality is commonly dened as the ability of the agent to construct plans of actions to reach a goal, possibly referring to a hierarchical decomposition of tasks. When considering a narrative, the agents it describes do not deliberate, or better, the decisions they deliberate have already been taken. Consequently, their behaviour is fully deterministic. Instead of considering a full account of agency, we opted to base our framework on a more constrained concept: the agent-role, which integrates the narrative and institutional role concepts in a intentional entity. Agent-roles are self-other representations, i.e. used to interpret, plan or predict oneself s or others behaviour. They are indexed by: (a) a set of abilities, (b) a set of susceptibilities to actions of others. In a social scenario, both abilities and susceptibilities become manifest as messages exchanged with agents playing complementary agent-roles. Thus, the topology of a scenario (e.g. Fig. 8.4) provides a fast identications of these indexes. Dierently from objects (and actors), however, agent-roles are entities associated also to motivational and cognitive elements like desires, intents, beliefs and plans.

6.4.2 Services and work flows
[TO BE DONE]
Chapter 7

Model-based problem solving and the legal system

7.1 Introduction

Having introduced modeling techniques for public administration, we now turn to reasoning with scenarios and multi-agent systems. Because diagnostic ability is the key to organizational agility, we devote special attention to model-based diagnosis. The general principles are however shared among types of reasoning.

For starters, we discuss the three spheres of activity to which problem solving activity, and associated knowledge assets, may be linked. Sphere and generic problem together, for instance operational+diagnosis, or development+monitoring, characterize a problem in an organization.

7.1.1 Spheres of activity

Our big picture of public administration is based on two, what we feel are, big results of knowledge representation research over the last decades.

The first one of these is the very general observation that all things that really matter, can be described at different levels of abstraction, and that these levels of abstraction reduce to each other in an imperfect way. This is true generally in science: the physical, the chemical, the biological, and the social-economic exist side-by-side. We can switch between levels of abstraction to explain a feature on one level in terms of another one. We all intuitively know which abstractions are at the bottom and which are at the top, but a coherent general-purpose reduction is unimaginably complicated.

One may abstract roads to lines, to planes, and to three-dimensional volumes, and all these are useful for some problems, but cannot as truths co-exist for the same object. In knowledge representation we find this notion in the abstractions of Hobbs (1995), or the concept of ontological stratification of, for instance, Borgo et al. (1996). We choose a strategy for picking the right level of abstraction for describing a problem, rather than to adopt all of them at once and ending in a knowledge representation quagmire.

In public administration we distinguish three distinct spheres of activity in which legal knowledge plays different roles, and is described on different levels of abstraction:
1. operations,
2. development, and
3. policy.

In the operations sphere we conform to the law, or violate it, in our actions. In this sphere we expect others to do things because of the law, and we tend to think of it in a positional sense, structuring our interactions with others.

In the development sphere we design and build social structures that are in compliance with the law or not, or that are functionally required by the law. In this sphere, we usually read the law from a different perspective, finding different duties than we do when we read it from an operational perspective.

In the policy sphere we can change the law, and are mainly interested in what works, as an instrument to steer collective behaviour in some direction. By and large, these spheres stand side-by-side, and only occasionally mix in the same decision making context.

These spheres of activity operate on different conceptualizations of a domain, and different interpretations of the same legal rules, and usually involve different people in different departments in a public agency.

Example To illustrate the different conceptualizations of a domain, let us for instance consider traffic rules as a simple example. Important traffic rules regulate who yields to whom on intersections between roads. When we make such rules, we categorize traffic participants and we invent traffic signals. Pedestrians may for instance be required to yield to all other traffic, and non-pedestrian traffic yields to the right, unless there is a road signal that tells it otherwise. In the development sphere, we organize road intersections in accordance with these rules, following a logic implied but not explicitly articulated by the rules, to minimize dangerous situations of conflict between traffic participants and optimize traffic flow. We place road signals where we want traffic to yield in exception to the general rule. We may also change the layout of intersections themselves depending on specific rules. If we want bicyclists to yield to motorized traffic somewhere, but there is no rule or road signal to that effect, we can split of a bicycle road from the main road before an intersection, creating two separate intersections, to make it possible to regulate the bicycle intersection with a yield road sign independently from the main road. When we finally drive on those roads, in the operational sphere, we 1) keep to the rules and yield where we have to, 2) we expect certain other traffic to yield and don’t watch it very carefully, 3) we expect traffic that we yield to, to move forward and clear the intersection, 4) we righteous get angry if traffic participants don’t act in their role, and 5) we expect that the intersections we cross have been designed to avoid conflicts potentially leading to collisions.

7.1.2 Generic Problems in Public Administration

A general asymmetry appears between the three spheres of activity. Operations informs development, and development informs policy making, of problems. Policy making guides development, and development guides operations, in presenting solutions. This asymmetry brings us to the next concept.

The second concept from knowledge representation research that we use is based on work, mostly in the eighties, on the categorization of types of problems. Our typology of generic tasks in public administration is based on the suite of problem types of Breuker (1994), an analysis of generic problem and task decompositions proposed in knowledge-based system literature at the time, mainly by Chandrasekaran and Johnson (1993); Clancey (1985); Steels (1990). The suite of problem types was a part of the CommonKADS methodology for the development of knowledge-based
systems (Schreiber, 2000), which enjoyed some success in business and government in the Netherlands. Breuker (1994) makes the following claims:

1. the availability of structural and behavioural models in a domain determines which problems can be conceived of and solved, and

2. there are recurrent functional dependencies between problem types characterized by the roles that knowledge plays in the problem definition.

A complete library of problem-solving methods with associated knowledge assets, organized by generic problem type, has been developed, described by Breuker and de Velde (1994) and we find an application of this library of knowledge assets to law by Valente (1995).

The suite of problem types presents us with one generic problem solving cycle for an intelligent agent, and two different vocabularies for it, depending on the type of model of the domain that is available, representing two alternative perspectives on what is in essence the same generic model of problem solving behaviour:

1. Model → Design → Implement → Monitor → Diagnose

2. Interpret → Plan → Execute → Monitor → Diagnose/Assess

When we feel able to control a domain by encapsulating processes into fixed structures and resources we decide on a design for dealing with a type of problem and implement it. To the extent we feel this is not feasible, given the unpredictable characteristics of an environment, we make situation-specific plans to address a problem, using the fixed structures and processes available. A fully articulated problem solving cycle would in principle allow description in both terminologies, but real world knowledge-based systems generally address some problem as primarily planning problems and others as primarily design problems.

The general structure of problem solving behaviour is constant. Given an ill-defined problem, we first choose a problem definition, which supplies us with a definite search space for a (design or planning) solution. We execute, or implement, a chosen solution, and we monitor, driven by knowledge-based expectations, whether execution or implementation is succesful. In planning-dominated problems, we typically, but not always, stop monitoring when a plan is completed. In design-dominated problems, monitoring remains relevant as long as the design is implemented. If our expectations are not met, we diagnose, as an intermediate step to a new modeling problem, to localize the problem, and start again with a new, localized, but ill-defined, problem. The diagnosis may lead to a localized repair of an otherwise functional model, or it may lead to an entirely new conceptualization of the problem, even a redesign of a service, depending on the diagnostic explanation found. Generally, we have a preference for keeping problems small and localized.

Fig. 7.2 presents the problem solving cycle in a more traditional agent vs. environment representation. Both the modeling and monitoring problem are fed with observations. The major difference between them is context: monitoring is done in the context of a plan and expectations about the observations to be received. Unexpected observations signal plan failure. Modeling is more open-ended, trying to mak esense of observations in the context of the intentions and abilities/susceptibilities of the agent. We have chosen to consider diagnosis an endpoint of model-based reasoning rather than feeding back the results of diagnosis into modeling and/or planning. Awareness of a diagnostic problem (that may implicate the agent’s behaviour model itself) signals that the agent is involved in an interaction that is beyond its abilities and knowledge.
Figure 7.1: The operations, development, and policy making problem solving cycles.

Figure 7.2: The problem solving cycle in a more traditional agent vs. environment representation.
Figure 7.3: Distribution of work between an operational agent and a dedicated diagnosing agent.

It is a reason to shift to another perspective, and to forward the problem to another agent (cf. Fig. 7.3).

Fig. 7.1 reflects our interpretation of how the public agency absorbs law into its operations. In the policy sphere, it is the planning vocabulary rather than the design vocabulary that dominates, reflecting the messy and open-ended nature of the activity. The really interesting thing happens when law is implemented, however. We then see a clear distinction between service, process, and product developments projects in supporting departments, where some of the law, the part deemed suitable for this, is implemented into new or existing organizational structures: services, processes, units, software, training, and manuals. Although project management involves some planning, it is of passing interest. It is the design of the products that remains as knowledge assets. These products then get rolled out to the operational level, where the planning vocabulary dominates, and the services, processes, units, software, training, and manuals become instruments used to get work done.

**Example** In the IND, service delivery is literally interpreted as the execution of a plan. When a client contact is initiated, a plan is automatically composed for that client, based on skeletal plans for the delivery of specific services, and appointments are scheduled. The client, too, receives a plan, mainly outlining the various formal statements the client will need to acquire and bring to appointments as evidence. The skeletal plan components, and the scheduling software, are however obviously the result of design.
An important function of diagnosis on the operational level is to detect violations of the rules.

Example Granting an immigration permit based on marriage is for instance a basic service, and its delivery a simple, routine plan execution. If there is however a suspicion of a sham marriage, based on observations, this is a diagnostic problem, leading to a long term monitoring plan, looking for confirmation of this suspicion.

Fig. 7.1 reflects our interpretation of design and policy making as core activities of the organization, requiring development and management of knowledge assets, and the assumption that the deeper level has direct access to, and control over, the diagnosis and modeling problem definitions at shallower problem solving levels.

7.1.3 Cybernetic Control Models of Organizations

The model as presented in Fig. 7.1 is suggestive of cybernetic control models of organization like the influential Viable System Model of Beer (1985), familiar in management circles. These models have a different origin, and are based on a scientific claim of another kind: the claim that organizations that survive are inevitably structured in certain ways. It is a truism that this kind of cybernetic control model describes the actual structure of many organizations, designed with management science abstractions in mind, which is in itself a justification for claiming that the three spheres can be distinguished, regardless of the empirical inevitability of this organizational structure.

In viable systems we distinguish a sphere of tactical operations, strategic planning, and normative planning, distinguishing concerns with what an organization is doing now, what it could be doing with existing resources and constraints, and what is feasible if certain normative constraints can be removed. Strategic planning is similar to what we see as the development sphere, developing new and maintaining existing abilities on the operational level, and normative planning is the purpose of the policy making sphere.

This is however not a model of problem-solving behaviour, but of control systems, and the problems of measuring performance (monitoring) and providing good feedback information (diagnosis) to the strategic and normative levels are not addressed on the knowledge level, and not considered to be fundamentally problematic.

Our problem-based approach adds to this control model by pointing out two important aspects of monitoring and diagnosis that relevant to policy making, being

1. firstly, that our ability to distinguish between diagnostic hypotheses, the underlying deep model for diagnostic argumentation, is constrained by the current operational implementation of a regulatory framework; and

2. secondly, given the degrees of freedom in choosing, and following up on, diagnostic hypotheses, that there is an argumentative aspect to the solutions to a monitoring and diagnosis problem.

Qualitative feedback information contains interpretation, and is based on, what is in a essence, a policy field and policy effects theory. In cybernetic control models, the system to be controlled is a given. In social-legal reality, it is ill-defined; a problem. In the next section we first address the functions of knowledge in problem solving, before we take account of the structure of legal knowledge.
7.1.4 Functions of Knowledge in Problem Solving Activity

The function of the suite of problem types in section 7.1.2 is primarily to classify knowledge sources used in the organization as belonging to, and taking their requisite structure from, a generic problem definition.

Firstly, let us address the planning vocabulary. Interpreting one’s problem in this vocabulary is a matter of choosing one’s role towards one’s environment. The first problem to be solved is therefore one of understanding a situation, and choosing a self-other relationship, or a system of affordances, that allows one to have grip on the situation. In a social situation, this comes down to choosing who we are, what our position is, and what interests, resources, and abilities we have in that position, and who they are, and what abilities and interests they have to affect our position.

The knowledge assets we use in planning, are 1) knowledge of our resources, 2) a set of planning operators and/or skeletal plans that represent our ability to change the situation, if the right resources are there, and 3) one or more goals. In simple planning problems, the environment is assumed to be inert, every change not initiated by the planner being a surprise. More dynamic, social settings involve counterparties, and we ascribe to them goals, resources, planning operators, and skeletal plans that represent their ability to change our situation. In social environments, we proceed from the assumption that we can infer the plans of others from the knowledge assets we have ascribed to them. This is the more relevant case for the field of law.

Since the success of a plan depends on our understanding of the situation, and on the expectations we have developed of the future action of others, actual developments during execution may surprise us. This is the significance of monitoring: we actively compare the outcomes of planning with actively obtained observations. Monitoring must therefore be functionally dependent on planning: without expectations and a plan, no observation is surprising. Diagnosis works on the results of monitoring: if there is an observation that cannot be explained by the model and the plans we have constructed, there must be some element or aspect of these that is not working, and a diagnostic hypothesis states that some elements are faulty, i.e. are not turning out as was expected. We have argued that the natural element to look for in public administration contexts is an agent, and that the fault consists of ascribing the wrong agent model, or specifically the wrong bundle of goals, abilities, or resources, to that agent (Boer and van Engers, 2011a).

In the design vocabulary it is not our understanding of a situation or the presence of counterparties that attracts our attention. The focus of design modeling is to find a configuration of components that meets complex constraints. To explain why we have a design problem, we may point to our limited abilities in some kind of typical planning situation. Designs help us develop our abilities. In a modeling problem for design, we choose our components and the rules for assembling these into configurations, and we make the constraints explicit in a form that allows us to test (partial) configurations against them.

In Agile, we design social arrangements, with agent roles as components (Boer and van Engers, 2011a). A completed design is a specification for something that still has to be implemented: a product. Social arrangements obviously differ from products like smart phones. The first one is intangible, and typically implemented only once in an organization. The second product is tangible, and mass produced. Both may however at some point be faulty, and the problem of determining whether a product’s properties deviate from its normative specification is a monitoring problem. Once the fault is established, we have a design-level diagnosis problem.
In the Agile framework, the fault is that some agent role components have goals, resources, or abilities that were not foreseen, and that cause the system as a whole to fail to meet its normative specifications. The diagnostic problem feeds into the modeling problem, and is in that sense the primary input to theory construction about our domain: the policy field and effects theory we are looking for.

7.1.5 Model-based diagnosis

An influential characterization of model-based diagnosis has been given by Reiter (1987). Model-based diagnosis of a design starts with a system specification, and observations that deviate from it. The purpose of a diagnosis is to find an explanation, pointing out one or more components that do not behave as specified. For diagnosis, it does matter what one will do with a localization of the fault. One may simply replace or repair components, if their occasional failure is acceptable. Alternatively, one may just change one’s model of the implementation, and learn to work with faulty systems. Diagnosis may also point at more fundamental design flaws, prompting reconsideration of the design itself, and reimplementation.

It is claimed, by Valente (1995), specifically for the field of law, that one does not monitor and diagnose, but monitor and assess, in planning. The logic behind this notion is that one does not repair other agents if they behave unexpectedly, because we have no such control over them, but rather that one comes to a moral qualification of their behaviour based on an assessment against norms, and punishes them.

We argue in that the assessment problem type understates the complexity of finding the misbehaving agents in a network of such agents in the field of law (Boer and van Engers, 2011a). We for instance show that there are closely similar scenarios for real estate fraud that involve only a seller, only a buyer, both buyer and seller, or agents representing these (Boer and van Engers, 2011c). Even in a basic sales transaction it is obvious to investigate whether we failed delivering on our part, before we start accusing our counterparty. Only in simplistic problem conceptualizations can we immediately jump to conclusions about who’s behaviour should be assessed. We propose that the original diagnosis/assessment distinction is informed mainly by a bias to design of complex physical systems, and planning in simplistic models of social environments, in early Artificial Intelligence. We therefore reject this general distinction in our problem conceptualization, and do consider the diagnosis problem type, with its focus on fault localization, to be as much part of the planning vocabulary as the design vocabulary.

Problem solving activities in modern organizations will partially be automated, and partially allocated to human problem solvers. Often we see a distinction between a heavily automated happy flow, as it is sometimes called, that deals with the great bulk of operational service requests, and manually handled hard cases that drop out of the happy flow. Moreover, only the first half of the problem solving cycle dominates service modeling: automated planning and scheduling of service delivery is common in practice, and automated diagnosis is very rare. The happy flow tends to be better modeled by design departments, even if it is not where the organization actually spends most of its resources. The happy flow gives the best return on investment for design activity in the operations sphere, if one models it in isolation, without addressing the development and policy spheres.

The agility problem, as perceived by organizations, originates in this bias towards the happy flow. Agile enterprise definitions generally focus on flexible configuration and scheduling of business processes to deal with rare, or even unforeseen, transactions
Lack of agility originates in systematic undervaluation of feedback from diagnostic processes in the organization.

### 7.1.6 Multi-agent Systems and Coordination of Problem Solving Activity

Application of the problem-solving paradigm to the activities of a complex organization is obviously not without its problems. The paradigm is most convincing when applied to a single, intelligent agent. Organizations consist of many such agents, who coordinate their activities, in order to give the appearance of being a more or less intelligent collective agency. When service delivery depends on a complex network arrangement, this metaphor of intelligent collective agency is stretched to its limits.

Fig. 7.4 gives a more general representation of the relation between problem solving agents. It applies to the organization in relation to its clients and network partners in the environment, but also to the agents in the organization, and in the state as a whole. Generally, agents influence each other’s problems in two possible ways: 1) they do what is expected of them, advancing the plan of the other agent, or 2) they do something unexpected, leading to a diagnosis problem for the other agent, and a reconsideration of their role by the other agent.

Agents that act collectively also have this problem. If a buyer and a seller conclude a sale, the seller has a duty to deliver, and the buyer a duty to pay. From the seller’s perspective, he firstly has a monitoring object towards the buyer: does the buyer pay? But the seller has this problem towards delivery too, if he does not control the sold good himself, and does not deliver himself. He also has to monitor the delivery agent, and if the delivery agent does not deliver, he still has the duty to see to it that delivery of the good will take place. Whether delivery is part of the same organization makes little difference. The monitoring responsibility has to be laid down somewhere.

Fig. 7.1 idealizes the organization as one single cybernetically controlled system capable of learning new policies. Cybernetic control models like the viable system model present us with a normative model of organizations: they should behave like a collective agency, but the constituting parts may have interests contrary to the interests of the whole. The key problem that organizations have to solve is the organization of distributed multi-agent diagnosis, in an environment characterized by variety in modes of, and reasons for, coordination of activity. Diagnosis is however also performed in that same environment.

Boer and van Engers (2011e,c) proposed a formal interpretation of model-based diagnosis of role-based multi-agent systems, with an assignment of health and fault modes to agents, playing legally constrained agent roles, as diagnostic hypotheses. The most complex aspect of this problem is clearly the collection of relevant observations.
In a system consisting of agents, one depends on agents for observations, and the motives of agents play a key role in the reliability of the observation. Moreover, whether one wants to make an observation depends not on its diagnostic value – as it would in a standard model-based diagnosis problem in a physical system – but on the observation’s effects on other agents, who are informed of the diagnosis taking place.

7.1.7 Conclusion

The problem decomposition into three spheres of activity is based on a simplification of control: deeper levels have access to, and control over, the diagnosis problem definitions at shallower levels (the dashed lines in Fig. 7.1). Communication between levels is fundamentally asymmetric: the deeper levels directs, and the shallower level informs. Any intentional, argumentative aspect of diagnosis feedback from shallower levels is obscured by the control model.

Because a regulatory framework, and its implementation, have direct consequences for the distribution of costs and benefits, and information, in a network of agents, intentions are likely to play a role in the way diagnostic feedback is constructed as an argument.

In the reasoning model presented in the next few sections, assumptions of control and position in the system are abstracted away, resulting in a model of reasoning that is as generic as it can be.

Obviously, the risk of trying to cover all is that one ends up saying nothing in particular. This chapter presents neither an executable logic nor a syntactic knowledge representation. In the next section an operationalization is given of the knowledge structures manipulated by the various generic problem types. Next we discuss some ways of reasoning with these knowledge structures: a more traditional approach based on default rules, discussed only in very general terms, and a more innovative approach based on Bayesian concepts, that will permit integration of data mining and statistics.

The latter method improves on the state-of-the-art in legal reasoning, and addresses an issue that is of great relevance in public administration: how to detect suspicious cases, and how to argue the suspiciousness of these cases based on statistical evidence before a court.

In section 7.6 we informally discuss some examples of model-based diagnosis applied to law without the intention of developing these into complete demonstrations of our modeling techniques. For these demonstration we have developed some simpler examples in later chapters. The chapter ends with a discussion of related research.

7.2 A formal model of problem solving with agent roles

This section presents concepts and assumptions that underpin our approach to problem solving, starting with some preliminaries.

Diagnosis, modeling, and monitoring problems, turn on observations and explanations. Given an observation, the problem owner should be able to generate a set of explanations, i.e. interpretations related to that social context, and pick the right one.

Where the observation represents the empirical, the explanation fills in the actual, and represents the real, in terms of the critical realism conception of knowledge in figure 3.1.
7.2.1 Multi-agent system and agent-roles

A multi-agent system consists of a set of intentional agents. Agents are rational: in order to bring about certain intents, they perform actions, to the best of their knowledge, producing acts (with possible side-effects), including speech acts, i.e. messages directed toward other agents. Agents may communicate only with agents in their operating range, i.e. whose message boxes are reachable to them.

The message box decides the structure of the multi-agent system. It connects the agent that receives messages through that message box, with agents that send to it. Each agent may have different message boxes, and each of them can be epistemically associated to an identity. Identities may be used to deceive other agents. Configuration of message boxes is part of the specification of an agent role.

Example A seller knows a message box of a buyer. Without one, it cannot play its role.

Agent-roles are self-other representations. They are identified with a set of abilities, a set of susceptibilities to actions of others, and associated with goals, plans, and beliefs. In a social scenario, both abilities and susceptibilities become actual as messages exchanged with agents playing complementary agent-roles.

Each agent role class $R$ is associated with:

1. A set of normal behaviour specifications $N(R) = \{n_1, n_2, \ldots, n_n\}$
2. A set of abnormal behaviour specifications $A(R) = \{a_1, a_2, \ldots, a_n\}$

7.2.2 Scenario agent

A scenario agent is an agent embodying an agent-role. It is deterministic in its behaviour: all parameters are either set, or determined by messages from other agents. This determinism, following from the internal description of the agent, constrains the temporal order of some messages between agents, but not the behaviour of an entire multi-agent system. Thus, a multi-agent system composed of scenario agents, playing certain agent-roles, will still represent a set of possible scenarios.

Proposition 7.2.1. A scenario agent is an allocation $(i, b)$ of an instance of an agent role $i \in R$ to a behaviour script $b \in N(R)$ or $s \in A(R)$.

7.2.3 Observation

One observes events, but the process of observation moreover determines temporal constraints between events, and in the case of messages in a multi-agent system, the apparent senders and receivers of messages.

Proposition 7.2.2. An observation $O$ consists of three elements: 1) a set of agent role instances, including the observer, 2) a set of messages between the observer and other agents, and 3) a temporal ordering relationship on messages (e.g. indexed by the observer on reception time).

Some observations are problematic, but not every observation qualifies as a problem asking for an explanation. Many observations are boring. Boring in the sense that an explanation of the observation adds nothing relevant to the problem context that was not already present in the observation. There is nothing more to the system than meets the eye. The explanation and observation may then take each other’s place: the
observation is symmetrically supervenient on the real (multi-agent) system posited by
the explanation.

**Example** A sale is boring, if, to explain the transaction between the buyer and the seller,
we only need to consider typical scenario agents for buyers and sellers. The story will not
mention the bank, or the tax administration, or prior social relationships between the buyer
and seller, etc. This does not mean that no bank or tax administration is involved in the
transaction, but that its role is too boring to be revealed in the observation.

Boring observations should generally be ignored as problems. In administrative
settings these are the routine cases handled in the *happy flow* of the organization.

The following quote, discussing the quest for fundamental particles of nature in
physics, exemplifies this use of boringness as a well-defined concept:

**Example** That the particles are postulated by a complete microphysics shows only that one
can tell a complete causal story with particles as protagonists. But one can tell a complete
causal story with divisible protagonists, provided that the divisions are boring, in that the
characteristic properties of all the parts supervene on the characteristic properties of their
wholes. We now have no evidence that there will be a final theory, no evidence that such a
theory will postulate anything that could serve as a mereological atom, and no evidence that
such a theoretical postulate will correspond to an ontological atom as opposed to a boringly

The complement of a boring observation is obviously an interesting one. An interesting
observation naturally requires a non-trivial explanation.

### 7.2.4 Explanation

The explanation allocates scenario agents to the agent roles in the observation.

**Proposition 7.2.3.** An explanation $E$ is a multi-agent system, and consists of three
elements: 1) a set of scenario agents, embodying agent-roles, 2) a set of messages
between the agents, and 3) a (partial) temporal ordering relationship on messages.

Given an observation, the problem owner should be able to generate or remember
a set of explanations, i.e. interpretations related to that social context. Where the
observation represents the empirical, the explanation represents the real.

A multi-agent system explanation may include 1) additional agents beyond the
observed ones, 2) the merging of multiple agents in the observation into one agent, or
3) the splitting of an observed agent into multiple agents.

**Remark** Note that the problem owner may inform another agent of his observation, or his
(partial) explanation of the observation, in the form of a narrative, and that it is not trivial
to distinguish between getting observations and getting explanations of observations.

### 7.2.5 Modeling problem

A modeling problem aims to find an explanations of observations received by the
problem owner from the environment with the purpose of designing an arrangement
of scenario agents to deal with those messages in an appropriate way. Its function
is specify a design/planning problem. The behaviour of scenario agents is a priori
considered normal: abnormal behaviours will be rarely be at the center of attention.
Result is a set of possible explanations in abstract form: the environment model.
7.2.6 Design or planning problem

Given an environment model an agent an finds an arrangement of scenario agents that meets certain goals. The process uses the same ingredients as explanation, but using practical knowledge aimed at action. Goals may be specified in the form of observations of the agents in the environment. The result is an expanded design model that extends the environment model to represent design/planning commitments of the agent, for execution, yielding observations.

7.2.7 Monitoring problem

A monitoring problem aims to find explanations of observations with the aim of comparing them with the design model of the environment. If the observation does not fit the design model, one has a diagnosis problem.

7.2.8 Diagnosis problem

An explanation is diagnostic if at least one of the participating scenario agents is deemed faulty (i.e. behaving abnormally in the agent-role) by the problem owner. To settle on a diagnostic explanation of observations, is to commit to qualifying certain scenario agents, implicating the environment or design model, as being at fault for the behaviour of the system.

Diagnosis leads to action, but outside the design/planning competence of the diagnosing agent. The diagnosis functionally feeds into a modeling or monitoring problem on a higher level of abstraction.

7.3 Allocation and rule-based reasoning methods

In the previous section we assume a relationship between observations and explanations with certain properties. A proper explanation undefeasibly determines observables: it accounts for the observation done sofar and moreover permits us to predict what will be observable in the near future. We like to be able to deduce events and observations from the explanation, and the explanation should give enough detail to permit that.

The other way around, an observation may permit many, possibly conflicting explanations, and denies many others. When we reason from an observation to an explanation, we use a kind of inverse deduction sometimes characterized as abduction, to find explanations capable of reproducing the observation.

We start of with a generic characterization of explanation, as an allocation problem. The allocation method approach to model-based diagnosis in Agile has been discussed in more detail in Boer and van Engers (2011c). In essence, one way to approach the problem is as an assignment or constraint satisfaction problem where the known set of senders and receivers of messages in the observation are the variables, the sets of normal and abnormal behaviours are the possible values for those variables, and the object is to find an assignment that reproduces the observation. Let the knowledge components be:

1. A set of agent role instances $IS = \{i_1, i_2, \ldots, i_n\}$
2. A set of agent role classes $RS = \{R_1, R_2, \ldots, R_n\}$
3. A complete 1:n relation $R(i)$ of agent role classes to instances (for all $i \in IS$ there is an $(i, R)$ where $R \in RS$)

4. A set of normal behaviours for each $N(R) = \{n_1, n_2, \ldots, n_n\}$

5. A set of abnormal behaviours for each $A(R) = \{a_1, a_2, \ldots, a_n\}$

6. An observation $O$, which includes a subset of $IS$.

7. Some arbitrarily structured background theory $BT$, if required, that permits consistency checking.

**Proposition 7.3.1.** A scenario agent allocation is an n:1 relation $SA$, where for all $(i, b) \in SA$, it is the case that $R(i)$ where $i \in I$ and $R \in RS$, and either $b \in N(R)$ or $b \in A(R)$. A complete allocation allocates one script to all $i \in I$.

A scenario agent allocation assigns a scenario agent to senders and receiver of messages.

**Proposition 7.3.2.** An allocation is an explanation of $O$, given $BT$, if $O$ follows from $SA \cup BT$, $SA$ does not already follow from $BT \cup O$, and $SA \cup BT \cup O$ is consistent.

The explanation as defined here simply follows the logic of inference to the best explanation\(^1\). Explanations need not be complete, and a partial agent role behaviour script allocation may form part of the problem definition.

Note that messages imply message boxes capable of sending, and susceptible to receiving, them. Agent roles are characterized by message boxes capable of sending, and susceptible to receiving, messages. Agent identity, from an external perspective associates with the message box. To model cases of deception we would have to separate these concepts in the allocation (which we do not do here).

The step towards diagnosis is simple:

**Proposition 7.3.3.** A diagnostic explanation of $O$ is an allocation $SA$, where, for at least one $(i, b), b \in A(R)$.

The above does not necessarily imply that no non-diagnostic explanation fits. That there is only a diagnosis problem if all non-diagnostic explanations have been rejected is a simplifying assumption in problem solving frameworks, not a reflection of reality, as the adversarial nature of court argumentation, where often one party offers diagnostic explanations, and the other non-diagnostic hypotheses, shows.

Finally, the agent role behaviour script allocation is a (parametric) design when used on the agent roles inside the organization with the purpose of deploying the right scenario agents for a (type of) scenario. The observation data structure no longer functions as a thing to be explained, but rather as a requirement on the design to be tested against. The requirements on explanation still apply. This is the attraction of the model: the model-based diagnosis model is a different perspective on a model that can also be used for design of services, and monitoring arrangements.

The great weakness of this allocation characterization is that it gives us no clue about which explanation in a set of hypothetical explanations to prefer.

The simple example below explains the allocation concept, and forms a bridge to the subject of rules.

**Example** Let observation $O_1$ be that $i_1$ sold something far under the commonly agreed market value to $i_2$. The variables to find an assignment for are $\{i_1, i_2\}$, where $\text{buyer}(i_1)$ and

---

\(^1\)(Boer and van Engers, 2011c; Bex et al., 2010a) mentioned elsewhere explain inference to the best explanation, and Lipton (1991) explores a whole space of possible accounts.
seller(i_2). For the buyer we know of three behaviours: the normal behaviour (N(buyer) = \{n_1\}) and two abnormal behaviours A(buyer) = \{a_1, a_2\}, where a_1 is being an idiot, and a_2 is evading taxes. For the seller we know of three behaviours: the normal behaviour (N(seller) = \{n_2\}) and two abnormal behaviours A(seller) = \{a_3, a_4\}, where a_3 is being an idiot, and a_4 is evading taxes. Three explanations exist: e_1 = \{(i_1, a_1), (i_2, n_2)\} and e_2 = \{(i_1, a_2), (i_2, a_4)\}. Any other assignment will not reproduce the observations. Tax evaders do nothing without coordinating their actions (i.e. only e_2 = \{(i_1, a_2), (i_2, a_4)\} produces a transaction), and idiots do not take advantage of each other (e_1 = \{(i_1, a_1), (i_2, a_3)\} does not produce a transaction).

7.3.1 Default rules

A scenario agent can be specified as a set of non-defeasible rules. No message is sent by a scenario agent that is not determined by its observations and initial beliefs. Knowing the scenario agent therefore in principle determines the events caused by that scenario agent.

In the example above, we can therefore simply make all assignments, execute them in a multi-agent system and determine which are consistent with our observation. In the process we are bound to discover some invariant rules, like we did in the example: Tax evaders do nothing without coordinating their actions (i.e. only e_2 = \{(i_1, a_2), (i_2, a_4)\} produces a transaction), and idiots do not take advantage of each other (e_1 = \{(i_1, a_1), (i_2, a_3)\} does not produce a transaction).

Since the scenario agent is an explanation tool, these non-defeasible rules describing their behaviour are vacuous as a tool for reasoning towards conclusions. They rather represent a set of ontological commitments on scenario agents, and function as constraints in testing explanations. Only using that knowledge for testing is bound to be inefficient.

Rules reasoning from observations to likely scenario agent allocations make great sense. These rules are defeasible (cf. for instance Pollock (1992, 1987)), because scenario allocations can be in conflict, and observations offer only partial access to the multi-agent system to be explained. Two such defeasible rules cannot be applied as part of the same explanation if the scenario agent allocations are in conflict. Priorities between such rules may be set by additional priority rules.

Proposition 7.3.4. An observation may be a reason to default to a (partial) explanation. We can write this down in the form of an (epistemic) \[\text{if reason then action}\] rule.

Example From the example above, we can take the default rules if O_1 then \{(i_1, a_1)\} \subseteq e and if O_1 then \{(i_1, a_2), (i_2, a_4)\} \subseteq e. The two rules are conflicted for O_1 because (i_1, a_1) and (i_1, a_2) cannot be part of the same explanation.

Rule-based approaches to explanation are common in computer science & law literature. The technical challenges to solve are conflicts in explanations, defeat and prioritization of rules, and the possibility of undercutting defeats. Horty for instance discusses all these concepts in Horty (2007). A final observation to make is that we have a choice, when using default reasoning to accept or reject the argument in Horty (2007) that the premises of \[\text{if reason then action}\] rules are the only proper reasons to try out a scenario agent assignment.

Proposition 7.3.5. Without a rule there is no reason to default to a (partial) explanation.
If we accept this proposition, the rules define the allocation search space. If we don’t, they just function as heuristics for trying out the most likely assignments first.

7.4 Probability-based reasoning methods

The second characterization considers the integration of probabilistic reasoning into model-based diagnosis. It mainly accounts for the question which explanation should be preferred given an observation, leaning heavily on the assumption that we can determine the space of possible explanatory hypothesis that can be generated from the available agent role behaviours, and that we are a priori indifferent to explanations.

A secondary purpose of this account is the possible integration of empirical data into model-based diagnosis. In law, and, because of that, in AI & Law, argumentation Bench-Capon and Dunne (2007), scenario-modeling, and the combination of both Bex and Verheij (2011); Bex et al. (2010b), are the traditional ways of theorizing about judicial reasoning and legal truth, while probabilistic reasoning has traditionally been treated with suspicion. In for instance Nulty & Ors v Milton Keynes Borough Council [2013] the court puts the point concisely and – to many indignant scientists – provocatively:

The chances of something happening in the future may be expressed in terms of percentage. Epidemiological evidence may enable doctors to say that on average smokers increase their risk of lung cancer by X%. But you cannot properly say that there is a 25 per cent chance that something has happened: Hotson v East Berkshire Health Authority [1987]. Either it has or it has not”.

Still, mainly because of the growing relevance of forensic scientific evidence, a proper integration of probabilistic reasoning into the argumentation process is increasingly a, hotly debated, problem. In the field of AI & Law we find recent attempts at such an integration by Fenton et al. (2012); Vlek et al. (2013). John Pollock, an authority on argumentation, presents in John L. Pollock (2007) an lucid philosophical critique on how probabilistic methods approach the problem of justification, in the form of some interesting legal puzzles that are hard to solve with probabilistic reasoning.

We believe we have successfully made this integration by distinguishing between observation an explanation, on the condition that explanations are independent alternatives.

7.4.1 Messages

Firstly, we assume that an explanation is specific enough to entail the occurrence of certain message events, and the temporal orderings of message events, given that these are described on the right level of abstraction. This is the most substantial informative assumption we have to make to perform ampliative inferences: an observed message or message ordering either fits an explanation or it doesn’t.

Given an explanation, all possible executions of the associated agent system result in certain messages. The likelihood of $E$ on $M$ (i.e. the conditional probability of message $M$ given explanatory hypothesis $E$) is one of $\{0, 1\}$ for any $E$ or $M$ known.
to be relevant to the problem.\(^2\) Obviously, this does not address the matter of (a) whether we can recover these messages from an imperfectly accessible social system, and (b) whether we can interpret their meaning correctly with certainty. We have a separate problem of abstracting our observations to the kinds of messages we are searching for. In the ideal case, the likelihood of \(E\) on \(O\) is equal to the likelihood of \(E\) on \(M\), and, for any \(E\) or \(O\) known to be relevant to the problem, is, again, one of \(\{0, 1\}\). If all likelihoods of explanations on a given observation are the same, then this observation is irrelevant.

7.4.2 Space of explanatory hypotheses

Secondly, we assume, in Bayesian terms, that an explanatory problem space is well-known, i.e. that the sum of probabilities of the set of explanatory hypotheses \(E_1, E_2, ..., E_n\) approaches \(1\) or, that, at least, the sum of probabilities of the set of explanatory hypotheses is much greater than the probability that there is no good explanation \(E_\ast\) of the events \(P(E_1) + P(E_2) + .. + P(E_n) \gg P(E_\ast)\), and \(P(E_1) + P(E_2) + .. + P(E_n) + P(E_\ast) = 1\).

**Proposition 7.4.1.** The problem is wellknown: As pointed out first by Van Fraassen, the problem of finding the best explanation assumes that we have good information about observations and explanations in the first place. Neither Lipton’s inference to the best explanation (IBE), nor Bayesian reasoning based on probabilities, tell us whether we should dare to find an explanation. If you know just one explanation, its prior probability would have to be unlikely high.

**Proposition 7.4.2.** Expected utility equals probability: if we have no motives for proposing explanations beyond fact-finding, the expected utility of entertaining an explanatory hypothesis would equal its probability.

7.4.3 Prior probability of explanations

Thirdly, as hard core explanationists, we may assume prior indifference between explanatory hypotheses \(E_1, E_2, ..., E_n\) if we are not willing to commit to prior probabilities. Thus, including \(E_\ast\),\(^3\) we have \(P(E) = \frac{1}{n+1}\).

Prior probabilities for explanations can be problematic, certainly in law. It is the space of possible explanations that matters, and not how often we settle on one of them, if we might be consistently wrong. In the Netherlands, for instance, 90% of the suspects of theft brought before a criminal court is convicted of theft. This does not justify the use of this statistic as a prior probability for the set of diagnostic explanations for judges. For a court this would be an unacceptably dangerous line of reasoning: because most people brought before the court are criminal, this person is likely a criminal. Other diagnostic agents may be entitled to jump to conclusions, in their own defense, however.

**Proposition 7.4.3.** Availability heuristic: The availability heuristic is a mental shortcut that occurs when people make judgments about the probability of events by how

\(^2\)If a buyer and seller are for instance colluding to evade taxes, the conditional probability of coordinating messages being exchanged between buyer and seller to mark a price up or down a price is 1. If, on the other hand, they are acting at arm’s length, the probability of such a coordination event is 0.

\(^3\)The no good explanation hypothesis \(E\) allows for discovery of learning experiences.
easy it is to think of examples. The availability heuristic operates on the notion that if something can be recalled, it must be important. One may think of the space of possible explanations in these terms. Although the availability assumption is often conceived of as a fallacy, it is in effect unavoidable. Since the selection of observations for diagnosis is not a random sampling method, and errors in explanations cannot be repaired post hoc, one typically cannot infer actual probabilities.

7.4.4 Probability of observations

Lastly, we expect to have empirical data about the prevalence of relevant observations \(O_1, \ldots, O_n\) that are habitually recorded in some decision making setting, which may lead us to have a theory on the prior probability of observations, and the conditional probability of observation, given other observations.

**Proposition 7.4.4.** Whether we obtain probabilities from data mining, or from carefully controlled experiments, prior probabilities are obviously always derived from conditional probabilities, and any empirical generalization is always open to additional qualifications of the conditions in which the probability holds, including the qualification that they no longer hold because something in the world has changed.

**Proposition 7.4.5.** Statistics-driven information gathering leads to reject inference issues. This means, in a nutshell, that an organization that accumulates knowledge about interesting statistical patterns in the suspicious part of a population that it accepted as a case for explanation, may not immediately generalize this knowledge to the larger part of the population that it rejected. The organization collects information that confirms its explanation biases. This is a serious theoretical limitation of the utility of statistical evidence from operational monitoring processes.

7.4.5 Surprisingness of observations

**Surprisingness** is in principle associated to a low probability of an observation given the normal state of the social system, i.e. \(P(O) \sim 0\). An observation \(O\) is a diagnostic problem if a diagnostic explanation \(P(E_d|O)\) is much more probable than a normal explanation \(P(E_n|O)\). If \(P(E_d|O) = 1\) then we are sure we have a diagnostic problem, but we usually cannot just sit around doing nothing until \(P(E_d)\) reaches 1. If we lack certainty as to whether observation necessitates a diagnostic explanation, the case is open to argumentation.

7.4.6 Evaluating explanations

To determine the relative value of an explanation \(E\), given \(O\), we may calculate the likelihood of \(O\) on \(E\), or the conditional probability of explanatory hypotheses \(E\) given observation \(O\). \(O\) confirms \(E\) if \(P(E|O)\) is higher than \(P(E)\), and disconfirms \(E\) if it is lower (the Bayesian confirmation constraint Tentori et al. (2007)). Unfortunately, this measure does not permit ordinal comparisons of explanations, given \(O\). Therefore, we calculate the confirmation value of \(O\) for explanation \(E\) with another measure, that permits ordinal judgments about explanations:

\[
c(O, E) = \frac{P(O|E) - P(O|\neg E)}{P(O|E) + P(O|\neg E)}
\]  

(7.1)
Experiments by Tentori et al. reported in Tentori et al. (2007) suggest that the above is the psychologically most plausible confirmation measure of those proposed in the literature. If $c(O, E)$ approaches 1 (-1), the observation $O$ confirms (disconfirms) the explanation $E$. If $c$ is equal to 0, the observation $O$ is irrelevant. Put in words, with this measure, an observation confirms an explanation if it is predicted by the explanation and discriminates the explanation from its alternatives. The more alternatives present that predict the message, the lower the degree of support the message gives to any hypothetical explanation, assuming prior indifference towards explanations. In other words: increasing or decreasing the set of possible stories affects the subjective likelihood of individual stories.

**Example** For instance, let $P(O) = \frac{1}{4}$, $P(O|E_1) = P(O|E_2) = 1$ and $P(O|E_3) = 0$, assuming indifference towards prior probabilities of explanations. The likelihoods of $O$ on $E_1$, $E_2$, $E_3$ are $\frac{1}{4}$, $\frac{1}{4}$, $0$ respectively. Confirmation value $c(O, E_1) = \frac{1}{3}$, and $c(O, E_3) = -1$ as one would expect.

### 7.5 Reasoning about argumentation structures

In law, we are used to look at justification as a game between players with opposed interests. The players posit claims, which attack or support other claims, and one player wins. Argumentation is generally perceived as operating at a meta-level, concerned with support and attack relationships between claims, rather than between messages and explanations. In fact, argumentation theory treats messages directly as claims. The problem solving philosophy advocated in this methodology may strike some readers as counter-intuitive compared to that one. A lot of knowledge acquisition activity in the field of law directly follows argumentation structure, focusing on cataloguing reasons to adopt legal conclusions, and dividing these into general reasons for and more specific reasons against conclusions. Even in legislation we see this form followed in the gradual buildup of exception, and exceptions to exceptions, to general rules. Argumentation structures are, besides narratives, also directly found in case law.

Argumentation structures are easily translated into default rules of the type briefly discussed in section 7.3. Complete spaces of relevant, truly independent, hypothetical explanations for observations are harder to conceptualize, let alone construct and use for reasoning.

In this section we briefly demonstrate the power of reasoning from observations to explanations using the statistical method.

#### 7.5.1 Disadvantages of argumentation structures

As a knowledge acquisition methodology, argumentation structures have serious disadvantages. Firstly, the rule or argumentation scheme is deceptive in its attractiveness. Semantically, default rules also manipulate spaces of possible explanations, but these remain opaque and ill-understood for most users, and experiment shows that people are in practice far less skilled at reasoning with sets of rules than they tend to think van Engers et al. (2002). This is in itself a reason to prefer managing knowledge of scenarios rather than rules.

Secondly, thinking of legal knowledge in terms of argumentation schemes or default rules allows for ontological sloppiness. The claim in argumentation is neither an observation nor an explanation: it is a message from a player that carries a proposition.
If we adopt the proposition, some explanations are in, and others are out. Confirmation and disconfirmation (of explanations) work both at the level of proposition (the events in a story) and at the level of the observed message (the events of the discourse). The Agile methodology does not distinguish what happens in court (or any practical decision making context in public administration) from what happened in the story.

In section 7.3 we proposed that default rules should take observations as conditions and partial explanations as conclusions. In prima facie argumentation structures as we may find them in a court decision, observations and partial explanations are inextricably tied together in claims.

While we do admittedly share intuitions about which claim supports or attacks which other claim, and intuitively place evidence in the bottom of an argumentation structure, the supports and attacks relationships do not easily let themselves be analyzed as pure relationships between propositions.

Formalizing burden of proof remains problematic, as does the influence that our opinions of the claimant on our evaluation of the claim (the argument from motive and position to know, for instance). The Agile methodology is an improvement in these respects.

### 7.5.2 An extended example

To demonstrate that the explanation model captures important intuitions about the strength of argument we reconstruct a complicated argument, one which is considered a hard case for the field of argumentation theory, in the vocabulary of the previous section.

In one of his recent papers John L. Pollock (2007), John Pollock, and authority on argumentation and defeasible reasoning (cf. for instance Pollock (1987)), outlines some relevant puzzles in defeasible reasoning and argumentation. He refers in particular to two properties that everybody acknowledges: (a) certain arguments are stronger than others, (b) when arguments exist for competing conclusions, their strengths are relevant to decide which is the conclusion to accept. Although some literature exists that attempts to integrate probabilistic reasoning with qualitative argumentation structures, some puzzles remain enigmatic.

He gives the following case:

**Example** Jones says that the gunman had a moustache. Paul says that Jones was looking the other way and did not see what happened. Jacob says that Jones was watching carefully and had a clear view of the gunman.

This is an example of collective defeat (Paul vs Jacob), which results in a zombie argument (Jones’s). From this story, Pollock targets some intuitive properties.

1. Given the conflict of witnesses, we should not believe Jones’ claim carelessly.
2. If we consider Paul more trustworthy than Jacob, Paul’s claim should be justified, but to a lesser degree.
3. Conversely, if Jacob had confirmed Paul’s claim, its degree of justification should have increased.

Pollock gives then an elaborated proposal for degrees of justification, based on probable probabilities. The complexity of the analysis in John L. Pollock (2007) is worth reading to understand the point of this example.

Working with a different perspective, we too find a solution to the problem. The characterization presented here has the merits of 1) being grounded in better objects
of reasoning than claims, 2) trivially solving Pollock’s puzzles, and 3) proposing a measure for the strength of support or attack derived from probabilities that might be used to evaluate argument structures. The characterization moreover highlights assumptions that underly all characterizations of the problem: we have to assume that the space of possible explanations is largely known, and that explanations are truly independent alternatives.

Fig. 7.5 shows a typical representation of this argumentation structure in the form of claims, and attacks and supports relationships. The essential first step is to separate observations from explanatory propositions:

Example Your observation A: Jones says to you that the gunman had a moustache. The claim, by Jones: the gunman had a moustache. This proposition is P. Your observation B: Paul says that a) Jones was looking the other way and did not see what happened, and b) that the gunman had no moustache. Claim a is about Jones, while claim b is about the story. Both independently attack Jones’ claim. For the example we care about the proposition that Jones was looking the other way, which is Q. Your observation C: Jacob denies Q. Q and Q attack each other symmetrically, while Q sort of attacks P.

The problem with the argumentation structure in Fig. 7.5 is that Q attacks the position to know of Jones, and not the propositional context of his speech act. Moreover, why should the conflict between Paul and Jacob decrease our trust in Jones?

We apply a method of chapter ?? to the story provided in the introduction. In that chapter a puzzle similar to this one is worked out in detail. In the example case used there we assign simple agent behaviours: the world is separated into liars (k) and truthtellers (k). This is the agent behaviour at stake in this case. In this case we have \(2^3 = 32\) possible scenarios for explanation. Dividing the world into liars and truthtellers is obviously not a terribly realistic approach to discovering the real causal mechanisms behind agent behaviour, but as long as we address only the narrative complications of this case, rather than the substance of the story (here reduced to the matter of the moustache), this is the best we can make of it. This is of course Pollock’s point: we have intuitions about the effects of the claims by Jones, Paul, and Jacob...
<table>
<thead>
<tr>
<th></th>
<th>(1) Jacob attacks Paul</th>
<th>(2) Jacob attacks Paul</th>
<th>(3) Jacob supports Paul</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(E_9)</td>
<td>(E_{24})</td>
<td>(E_{26})</td>
</tr>
<tr>
<td>Jones tells the truth</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Paul tells the truth</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>Jacob tells the truth</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>Jones saw the gunman</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>gunman had a moustache</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>(P(k(\text{Paul})))</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P(k(\text{Jacob})))</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c(O_2, E))</td>
<td>0.72</td>
<td>0.72</td>
<td>0.72</td>
</tr>
<tr>
<td>(c(O_3, E))</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Table 7.1: Confirmation factors in Pollock’s puzzles
even without knowing their roles in the story, and we would like to see our reasoning method reproduce those intuitions.

For simplification, we neglect $E_\ast$. We distinguish the observation $O_1$, where only Jones has spoken, from $O_2$, where Paul has spoken too, and $O_3$ where Jones has attacked (or supported) Paul’s claim.

7.5.3 Conclusions

The outcomes are on Table 7.1, reporting only explanations confirmed by the complete observation. Referring to the targeted properties, we have the following results. (1) Assuming indifference toward hypotheses, our approach confirms to the same degree hypotheses in which the gunman has a moustache, and not. (2) Using for instance $P(k(\text{Paul})) = 0.8 > P(k(\text{Jacob})) = 0.5$, the hypothesis in which Paul is telling the truth is the one confirmed to the greater degree. (3) Seeing that Jacob confirms what is said by Paul, we observe that the confirmation factor of the hypothesis they both support increases, but just as much as the hypotheses in which they are both lying. This is an important consequence of indifference towards prior probabilities.

What this reconstruction fails to do, however, is tell us why observations $O_1 - O_3$ came to us in that order: To solve that question, we need more detailed agent behaviour models that account for the functions of the claims. It also does not tell us why we believe in the supports and attacks relations of Fig. 7.5.

In law, and, consequently, in AI & Law, argumentation (Bench-Capon and Dunne (2007)), scenario-modeling, and the combination of both (Bex and Verheij (2011); Bex et al. (2010b)), are the traditional ways of theorizing about judicial reasoning and legal truth, while probabilistic reasoning has traditionally been treated with suspicion. Still, mainly because of the growing relevance of forensic scientific evidence, a proper integration of probabilistic reasoning into the argumentation process is increasingly a debated problem. In AI & Law we find recent attempts at such an integration by Keppens (2011); Fenton et al. (2012); Vlek et al. (2013). John L. Pollock (2007) gives a lucid philosophical critique on how probabilistic methods approach the problem of justification, in the form of some interesting legal puzzles, ending up with a ptolemaic solution based on probable probabilities.

This work proposes an entirely different perspective on Pollock’s puzzles. First, we consider the direct relation between a individual message and an explanation, and the space of hypothetical explanations. Support between claims is a derivative measure, and may even be based on a consilience effect: the claim merely supports the other claim because they both confirm the same explanation, while otherwise being uncorrelated. Second, we emphasize agents’ positions. We depart from the assumption that the physical, mental, social and legal positions an agent finds himself in determine his observation and actions. All agents, including the neutral third party adjudicator, are bound by this constraint. The concept of agent-role is central in this frame. Eliciting agent-roles, we encourage the modeller to consider scenarios from the perspective of the participants, which is uncommon in traditional argumentation theory. In conformity to social reality, the diagnostic agent is part of the multi-agent system, has limited access to the information exchanged, and manipulations to obtain it may have side-effects in the system.

Thus, the quest for a solution to a problem case requires not only an investigation into the structures and processes that made the occurrence of the case possible, but also into the subsequent process of elicitation of explanations of the case. Parties simply bring relevant observations that support their explanations, contrasting them
from the explanations given by the counterparty, and also judges make choices at an intentional meta-level, that may be interpreted using values, as proposed by Bench-Capon and Sartor (2003). Residuals of the latter decision-making processes are found in the existence of votes, and dissenting and concurring opinions, on the bench.

Obviously, our easy solutions do not solve Pollock’s argumentation puzzles within the rules of his game, but they clearly demonstrate the added value of our model-based diagnosis framework, proposed first in Boer and van Engers (2011d,b), in the field of law.

7.6 Explanation problem spaces illustrated by some examples

In the next few chapters additional examples of reasoning with multi-agent systems will be worked out. Because such models easily become rather complex as a tool of explanation, we include here two additional, informal, examples of the kind of diagnostic reasoning we envision, illustrating the factors that influence the quality of monitoring policies.

7.6.1 The pickpocket

The train conductor stands in the doorway of his train, watching a fight between two men erupt on the railway platform. While he routinely picks up the microphone to warn the passengers that pickpockets may be present in the train, a passenger walks up to him: “My laptop bag has just been stolen while I was watching those men from the window”.

This is a familiar storyline: the pickpocket’s diversion. The story is consistent with a number of very different hypotheses, and police officers will have a hard time producing evidence:

1. The pickpocket used a spontaneously arising diversion to steal the bag, making the pickpocket the sole offender.
2. An accomplice of the pickpocket picked a fight on the platform to create a diversion, allowing the pickpocket to steal the bag. The pickpocket and one of the two men involved in the fight are co-offenders.
3. Two accomplices of the pickpocket simulated a fight on the platform to create a diversion, allowing the pickpocket to steal the bag. The pickpocket and both men involved in the fight are co-offenders.
4. The passenger, guessing how the fight will be interpreted by the train conductor, uses the opportunity to make the train conductor a witness to the simulated theft of a non-existent laptop, in order to make an insurance claim. If the train conductor calls the railway police, the insurance claim of the passenger gains credibility. The passenger is the sole offender.
5. The passenger left it on the railway platform. No offense took place. Perhaps the laptop bag was brought to a lost and found desk.

Immediate collection of witness statements, in the train and on the railway platform, would help the police to pick the most likely scenario. But the train must go on, and a minute later most witnesses will have left the scene of the crime. Suppose that
you are a railway police officer hurrying to the scene before the train leaves. What do you do? Do you want to board the train or stay on the platform? Are the two men on the platform, the thief, or the passenger in the train on your mind?

What should a railway police protocol for this type of event look like? And, considering the paucity of evidence: Does this kind of scenario ask for cameras in trains and on platforms? Should the railway company store personalized ticket information? This is the kind of question that public administration faces in implementing compliance controls. The legal qualification of the story – theft, assault, insurance fraud – and the agents in the story that should be held legally responsible, depend on which scenario really happened. Both the qualification and the assignment of responsibility largely depend on the intentions attributed to agents, even if those intentions do not later play a formal evidential role. Note that the thief is not physically present in the scene, just presumed to exist. Note that timing is important for collecting evidence.

7.6.2 Tax evasion

Now consider this more relevant storyline, which is more representative of diagnostic reasoning in public administration:

The cadastral register claims \((e_1)\) that a project developer sold an apartment for €350,000 to a natural person \((e_2)\). The payroll tax register claims \((e_3)\) that the project developer paid wages to that person \((e_4)\). The cadastral register claims \((e_5)\) that this person sold the apartment three months later for €500,000 to a third person \((e_6)\), but NVM real estate brokers claims \((e_7)\) that median home price hardly changed in that period \((e_8)\).

This scenario is relevant to a tax administration. The story is consistent with a number of very different hypotheses, which are often hard to tell apart. In the field of real estate we find a variety of types of crime consistent with the observation of large deviations from a reasonable market price, or untypical quick depreciation or appreciation of a property. People may sell below or buy beyond a reasonable market price to avoid income taxes; in a foreclosure auction, a seller may unknowingly or unwillingly sell his property below a reasonable price to a buyer cartel, which distributes the profit among the cartel participants (bid rigging); the transaction may hide a bribe or theft; and finally the seller or buyer may be a victim of extortion.

The reader of this paper, having his imagination already activated by the previous example, will undoubtedly realize that the information presented permits for many detailed hypotheses of varying plausibility, for instance:

1. NVM median home price development is not representative for the specific property.
2. The project developer, in order to avoid immediate insolvency, sold a property in great haste to one of its employees.
3. The employee, acting as the agent of his employer, stole €150,000.
4. The employer paid the employee €150,000, avoiding payroll taxes.
5. The third person was forced at gunpoint to acquire a property worth €350,000 for €500,000.
6. The employee, acting as the agent of his employer, was forced at gunpoint to cash in €150,000 from his employer, and an extortionist made off with the money.
7. Another employee, who owed the first €150,000 euro, acting as agent of the employer, sold his property to the employer first, in order to extend the transaction chain and escape detection, and then to the other employee, who cashed in the difference, to evade income taxes.

8. The project developer was a victim of bid rigging in a foreclosure auction, coordinated by the employee.

We can’t decide, based on the story alone, whether something extraordinary happened, and, if so, who is the beneficiary of the extraordinary transaction. Different explanations of the story point to different offenders and victims, and to different legal qualifications of the story. Note also that the value of witness testimony depends on these qualifications: an offender for instance has a prima facie motive to misrepresented what happened.

The events in the story are the result of the plans of the agents involved. Rather than understanding the story directly on the level of plans, we propose to explain the story on the level of agent role behaviour descriptions for the buyer and seller. This allows us to group together in a coherent unit the goals and adaptive plan operators that characterize

1. plans for buying or selling, and
2. plans for dealing with information seeking by third parties.

This deals with the intuition that the answers to the classical critical questions about the reliability of testimony depend on the roles one could play in the story. These critical questions are whether the agent is in a position to know something, and whether the agent has a motive to falsely claim something. In addition, we have a way of dealing with track covering behaviours triggered by information seeking.

Let’s consider the example story in this context. If it is a simple case of tax evasion, the buyer and seller collude against the tax administration, and will coordinate testimony. They have reason to fabricate evidence for circumstances that would make the transaction normal, for instance by having the property retroactively appraised for €350,000 by a fraudulent professional appraiser, claiming great urgency, or claiming that the sale was agreed long ago when the reasonable market value of the property was €350,000. In the case of a theft, on the other hand, an agent representing the seller and the buyer colluded, but another agent representing the seller would likely offer reliable testimony. It is arguable that the agents were in a position to know that the suspicious transaction was going to be easily noticeable, and that therefore something more complicated, involving for instance extortion, could be going on. A thief or extortionist, becoming aware of having aroused suspicion, may react by threatening potential witnesses. Etcetera. These processes all influence the effectiveness of the monitoring process.

The following two behaviour descriptions for tax evasion satisfactorily explain event $e_2$. $e_4$ is only potentially relevant as a background to the motive for paying the employee. Plan operators follow the general pattern If event (given conditions) then plan, and adopted intentions in events and conditions start with to. First we present a script for the seller $s$, to pay an untaxed amount $a$ to $b$:

1. To pay an untaxed amount $a$ to $b$, propose to sell a property worth some $v$ for $v - a$ to $b$ and pretending the agreement was made long ago.

2. If $b$ accepts proposal to sell a property worth some $v$ for $v - a$ then propose a specific property worth $v$ to $b$.

173
3. If \( b \) accepts proposal of a specific property being worth \( v \) then secure property for \( v \), and sell property for \( v - a \) to \( b \).

4. To sell property for \( v - a \) to \( b \), offer to sell property for \( v - a \) to \( b \).

5. If \( b \) accepts offer to sell property for \( v - a \) then register property transfer to \( b \) in cadastral registration and monitor payment of \( v - a \) by \( b \).

6. If someone asks about the price \( v - a \) then claim the agreement was verbally long ago.

In the interest of simplicity the plan is rather static. Note (2, 7) that fabrication of evidence plays a role in the planning. Next we present a script for the buyer \( b \), to receive an untaxed amount \( a \) from \( s \):

1. If \( s \) proposes to sell a property worth some \( v \) for \( v - a \) and pretending the agreement was made long ago given the goal to receive an untaxed amount \( a \) from \( s \) then accept proposal from \( s \) to sell a property worth some \( v \) for \( v - a \).

2. If \( s \) proposes a specific property worth \( v \) to \( b \) then have property for \( v \) appraised and consider proposal of specific property for \( v \).

3. If property for \( v \) was appraised given the goal to consider proposal of specific property for \( v \) and the appraisal shows that the property for \( v \) is acceptable then accept proposal of the specific property being worth \( v \).

4. If \( s \) offers to sell property for \( v - a \) to \( b \) given accepted proposal of a specific property being worth \( v \) then accept offer to sell property for \( v - a \) then monitor registration of property transfer by \( s \) in cadastral registration then pay \( v - a \) to \( s \).

5. If someone asks about the price \( v - a \) then claim the agreement was made verbally long ago.

6. If someone asks about the appraisal then claim the agreement was an option to purchase, and the appraisal was made to decide about exercising the option then quickly inform \( s \) about the optional nature of the agreement.

Here again fabrication of evidence plays a role (2, 6, 7). In this plan, having the appraisal performed may attract attention: why have the property appraised before acceptance if the agreement already exists? Step 7 is a plausible, but – from a timing point of view – hairy improvisation in response to the monitor, that must be supported by \( s \).

Of interest are the strategic monitoring choices that this scenario suggests:

1. This specific pair of behaviour descriptions suggests that obtaining testimony from buyer and seller simultaneously is important to prevent them from filling in the blanks in the feigned verbal agreement.

2. Because appraisers work in a large area, only a central registry for appraisals is a viable option. In this scenario, the appraiser is an innocent bystander, meaning that 1) a duty to inform the tax administration about appraisals made is in this context an undesirable administrative burden, and 2) that the appraisal may be assumed to be reliable.

The associated monitoring script interacts with the cadastral registry, payroll data registry, online NVM database, appraisers, buyers, and sellers. It is omitted for reasons of space, considering its complexity even in a simplified domain model. Let us assume
the investigation completes the story with the following interpretation of events by the monitor:

The parties involved agree \((e_9, e_{10}, e_{11})\) that the reasonable market price is \(€500,000\) \((e_{12})\). The project developer \((e_{13})\) and its employee \((e_{14})\) both claim to have agreed previously to an option to purchase the property for a then reasonable price \((e_{15})\). The employee later also claimed \((e_{16})\) that the agreement was an option to purchase, and the appraisal was made to decide about exercising the option \((e_{17})\). The tax administration claims \((e_{18})\) that the verbal agreement claimed in \(e_{15}\) is an unverifiable result of a fabrication for the purpose of evading taxes \((e_{19})\). The project developer and employee countered with the claim \((e_{20}, e_{21})\) that the relationship of trust that exists between them made requiring a written agreement imprudent \((e_{22})\). The tax administration claims \((e_{23})\) that the difference between the reasonable price and the actual price paid \(€150,000\) is taxable income \((e_{24})\).

From an explanatory point of view, \(e_{24}\) is the conclusion, supported by \(e_{12} (e_9, e_{10}, e_{11})\) and \(e_2 (e_1)\). The argument for a contrary conclusion is supported by \(e_{15} (e_{13}, e_{14})\). If \(e_{19}\), the weak attack in the tax administration’s reconstruction on \(e_{15}\), is not accepted by the tax court, a new policy guideline points the way out: only written options to purchase real estate should be enforceable, and they should be duly registered somewhere to prevent antedating. More generally, the tax administration’s explanation of events is that the project developer and employee were following scripts \(s\) and \(b\), and that the tax administration’s specific monitoring policy in this case, script \(m\), and the opportunities created by the civil code’s rules on agreements, which require that real estate transactions must be provable to third parties, but options to purchase need not be, makes enforcement of tax evasion a problem in this case.\(^4\)

### 7.7 Related research

#### 7.7.1 CommonKADS problem solving methods in the legal field

Many in the field of intelligent systems implementing law have addressed aspects of the CommonKADS problem solving cycle in their work. In this section we categorize some approaches, and some relevant work in other fields.

Both rule-based (Binbasioglu and Zychowicz, 1998) and case-based approaches to decision support (Jung et al., 1999) may be considered diagnostic in a general sense. The notion of model-based diagnosis, as opposed to assessment, however, presupposes fault localization in a complex system, and that element is missing in, what Breuker (1994) would classify as, assessment tasks. In assessment problems, one comes to a judgment, a qualification. The next problem, however, is to localize responsibility, or fault.

As observed, intelligent systems in the field of law are usually based on either the concept of normative reasoning by agents following formalized legal rules, or the concept of legal argumentation about individual cases.

The formalization of legal rules prior to implementation of services is common practice in executive suborganizations of the state, including our research partners.

\(^4\)A note of caution: this plan is obviously not state-of-the-art in tax evasion.
A common academic approach to business process compliance formalizes legal rules as deontic statements, and simulates proposed implementations as a normative multi-agent system, cf. for instance (Governatori and Sadiq, 2009; Boella et al., 2008). This is however not the kind of environment that allows for the animation and classification of stories about real world incidents beyond the confines of the model. It is a valid design paradigm if the problem of filling it with the right knowledge about the world is addressed.

It is the occurrence of incidents, whose meaning is subject to argument, that leads to another important tradition in legal knowledge representation, more useful for knowledge acquisition: the formalization of arguments and accreting counter-arguments, and counter-arguments to counter-arguments. In practice, this evidentiary inference approach is the more relevant approach to the design of fielded decision support systems today. Decision support systems we see in the tax administration and immigration service simply collect common arguments for and against a decision. These are then ticked off by an employee, and supplemented with evidence, in support of service delivery. The weakness of this approach is that it does not provide us with classification criteria for isolated components of argument schemes, which lose the connection to the stories they came from. Besides that, it has no clear ontological connection to the normative agent point of view that is useful in design.

Statistical and Bayesian approaches to decision making, another form of evidentiary reasoning, suffer similar problems: statistically valid information about a domain can only be collected if one keeps the conceptualization of the domain constant. Even a tax administration cannot investigate enough suspect cases to collect valid statistics for any relevant piece of evidence, and certainly not if tax regimes changes nearly every year, inviting new forms of evasion. Statistical approaches address only some, isolated problems, well.

In the field of legal knowledge engineering, there are some accounts of incident stories that may be considered direct precursors to the conceptualization presented here. There is firstly, a tradition of story-based inference to the best explanation, where the object is to match the evidence presented in a case with the best story, constructed from causal generalizations, to explain it (e.g. (Lipton, 1991; Keppens and Schafer, 2006)). Bex et al. (2010a) present a hybrid proposal with both causal and evidentiary inference. The reusable design components in this conceptualization are rather small: defeasible evidentiary and causal generalizations that may be used to back up individual arguments. While this is a flexible theoretical model of the construction of hypothetical stories, it presents practical problems for knowledge management in a large organization that knows of many stories and has to organize the knowledge about them in some way.

We propose the allocation of hypothetical complete mental processes to agents as an account of common sense explanation of stories. Attractive in this proposal is that it applies the model-based reasoning paradigm to mental models of agents, and supplies a criterium for classification: health and fault modes of agent roles.

The problem of multi-agent model-based diagnosis has been addressed in various contexts, but our approach to it appears to be original. There are several interesting ways to combine multi-agent systems with model-based diagnosis that have been explored by others. One approach focuses on distributed diagnosis of (generally non-agent) systems (Kalech and Kaminka, 2007; Roos et al., 2003). This addresses the organization of network arrangements for monitoring and enforcement, but not the social arrangements being monitored. Another approach is to diagnose multi-agent plans. A diagnostic hypothesis from this point of view is an identification of failing
parts of a plan (Witteveen et al., 2005). Although this approach treats a multi-agent system as the subject of model-based diagnosis, the agents are not the components of the system of interest. This problem formulation has some similarity with the one addressed in this paper, but the setting to which it applies is a fully cooperative setting. Like (Kalech and Kaminka, 2007; Roos et al., 2003), it operates on the assumption that problems are at the root caused by different or false beliefs about the non-agent environment, and not by competing objectives.

Many of these approaches are complementary to our approach, although we are not aiming for integration with specific other frameworks at the moment, limited not only by our resources, but also by the fact that tax administration and immigration service, and other such organizations, already have methods, specifications, and commitments to representation languages.

7.7.2 Model-based diagnosis

The concept of model-based diagnosis is a generic problem definition, describing a problem structure repeatedly found in early knowledge-based systems. One has a diagnosis problem if one’s expectations deviate from one’s observations. Something is wrong. If one has a model of a system consisting of components, and one’s expectations are based on this model, the diagnosis problem is to localize the fault in one or more components. A diagnostic hypothesis is a hypothesis that some component(s) are broken, and others are not. The model-based diagnosis problem has resulted in various formal models, for instance by Reiter in Reiter (1987), and others, cf. for instance de Kleer et al. (1992). It is distinct from the assessment problem, which deals with judgments, but not with fault localization. Assessment judges a single person, or event, or case, or relationship in an environment, generally resulting in an accept-reject decision. Superficially, assessment may appear to be the problem type one addresses when judging violation of the law, but it is in our view a naive model of enforcement.

Legal argumentation is most neutrally represented by argumentation frameworks. The argumentation framework introduced by Dung in Dung (1995) presents a very general and abstract account of the structure and acceptability of arguments in an exchange of arguments. The concept has gained currency as a general theory of argumentation in legal knowledge engineering, mainly because logic programming and several major formalisms for defeasible and nonmonotonic reasoning, including Reiter’s default logic (Reiter, 1987) and Pollock’s inductive defeasible logic (Pollock, 1992), can be explained in terms of argumentation frameworks.

We extended the model-based diagnosis problem definition in Breuker (1994) to include new types of diagnosis problems, based on a functional similarity between diagnosis and assessment problems observed by Breuker in Breuker (1994), to bring it into the domain of law and social structures. Initially, in Boer and van Engers (2011c) we restricted our attention to Reiter’s formalism for model-based diagnosis problems, because Reiter reduces model-based diagnosis to default logic reasoning about a domain model with certain properties. This formalism is rather restrictive as a model of the acceptability of diagnostic hypotheses given observations, preferring for instance single fault hypotheses over more complex hypotheses. In Boer and van Engers (2011a) we therefore also formulated a more general concept of diagnostic hypotheses, without committing to a way of determining preferences over them. These preferences are in reality based on a variety of opportunistic criteria, often determined by available resources for monitoring and enforcement.
To bring model-based diagnosis into the legal domain, we firstly introduced agent roles as components of social systems, based on an analogy between function and social role. An agent may play multiple roles, but whether roles are played by the same agent is only of secondary importance in theory construction in law, as argued in Boer and van Engers (2011c). The same agent may act in conflicting roles, without taking advantage of opportunities offered by them, and whether unpermitted coordination takes place between two agent roles does not depend on whether they are the same agent, although it is certainly evidence of the possibility of coordination. For instance CEO a of company c may have financial dealings with private person a. Clearly they are not at arm’s length from each other, but as a coordination pattern it does not distinguish CEO a of company c having financial dealings with his wife a’. At issue is the motivation and evidence for not acting at arm’s length in financial transactions.

1. the diagnostic value of information,
2. the reliability of the information that can be obtained,
3. the costs of the action to obtain it, and
4. the effects of obtaining it on future action.

### 7.7.3 Reiter’s method

Reiter’s work, in Reiter (1987), is an important historical landmark in model-based diagnosis, in that it recognizes model-based diagnosis as a generic problem type and presents a convincing logical approach to it. The same approach can be applied to the problem of diagnosing compliance of multi-agent systems.

The default reasoning approach to model-based diagnosis in Agile has been developed in Boer and van Engers (2011c).

Following Reiter (1987), he characterizes a description of a system as a pair \((SD, COMP)\), where \(SD\), the system description, is a set of first order sentences, and \(COMP\) a finite set of constants identifying components.

Our components are the set of agent role instances in \(I\).

Typically, a system description describes how a system normally behaves, and it often distinguishes a description of structure from a description of function of the components. The functional model causally relates input and output terminals of components, and terminals of components are connected.

In a multi-agent system we find the same system topology, but with the big difference that the system topology is internalized in the scenario agents in the form of agent role beliefs associated with known message boxes (cf. section ??).

An observation of a system is a set of first order sentences \(OBS\) on the events happening at component terminals.

**Proposition 7.7.1.** If \(OBS\) and \(SD\) are inconsistent with the assumption that all components are normal, certain of the components behave abnormally. A diagnosis is a hypothesis that components \(AB \subset COMP\) in \((SD, COMP, OBS)\) are abnormal and the rest normal.

The diagnostic process usually involves making additional observations as evidence for ruling out hypotheses (measurement). Diagnostic reasoners may use the functional model for both causal and evidential reasoning.

Diagnoses can be generated on the basis of component fault modes (our abnormal behaviours). The diagnosis in this case conjectures alternative behaviour descriptions.
for the components that are behaving abnormally. The set of fault modes of a component may be complete: in this case the component must be behaving according to the health mode or one of the fault modes. Alternatively there may be generic unknown fault models.

**Proposition 7.7.2.** Default reasoning about normality of component behaviour can be modeled with a predication of (ab)normality, \( n_i \) for any \( i \in N \), and a normal default theory of the form \( DT = (\{ Mn(c)/n(c) | c \in COMP\}, SD \cup OBS) \) Reiter (1987).

For this default theory, Reiter’s default logic extensions are those of the generic diagnosis problem for \((SD, COMP, OBS)\) directed towards a minimal set of abnormal components Reiter (1987). The preference criterium for explanations is the minimal fault assumption: the diagnoser will prefer an explanation based on a single misbehaving agent.

In \( SD \) a complete set of fault modes \( a_1, a_2, \ldots, a_n \) can be expressed by the first order axiom \( r(i) \land \neg n(i) \supset a_1(i) \lor a_2(i) \lor \ldots \lor a_n(i) \) Reiter (1987), and alternative health modes \( n_1, n_2, \ldots, n_n \) by \( r(i) \land n(i) \supset n_1(i) \lor n_2(i) \lor \ldots \lor n_n(i) \), where \( r \) is the component type to which the partition into behaviour modes applies, which is in our problem the agent role.

Additional sources of knowledge may guide the exploration of the diagnosis search space as it is characterized here. Diagnoses may for instance be ruled out on grounds of impossibility, faults in components may imply faults in other components, and a probability distribution of fault modes may be known that guides selection of hypotheses.

Weaknesses of the approach for our purposes are 1) the minimal fault assumption, which often fails in social domains, and 2) the externalization of system structure does not fit well with our agent role concept.

We see social structure as an artifact of agent beliefs about their social environment. In our conceptualization social structures supervene on the beliefs about the social structure shared by the agents that participate in it.

### 7.7.4 Links with argumentation

Default logic extensions are stable extensions in terms of Dung’s argumentation frameworks Dung (1995). This makes the notion of a diagnosis concrete enough to implement it with logic programming Dung (1995). We used the multi-agent programming environment Jason, which has such an extension Bordini et al. (2005).

Another interesting use of the connection to Dung’s argumentation frameworks is that these have been used to account for legal argumentation, making this an interesting starting point for connecting model-based diagnosis to the evaluation of legal argumentation.

### 7.7.5 Statistics in argumentation

In the field of AI & Law we find recent attempts at an integration of statistical reasoning of the type performed by Bayesian networks with argumentation frameworks by Fenton et al. (2012); Vlek et al. (2013).

John Pollock, an authority on argumentation, presents in John L. Pollock (2007) an lucid philosophical critique on how probabilistic methods approach the problem of justification, in the form of some interesting legal puzzles that are hard to solve with probabilistic reasoning. The problem in our arises because information about
probabilities do not naturally attach to attacks/supports links between claims. Our approach is rooted in this observation.

Integration of probability-based decision making in an explanation framework turned out to be one of the most pressing concerns in the Agile project. Bayesian network reasoning depends on the availability of prior probabilities. These should be discovered through controlled experiment. The current popularity of evidence-based practice (EBP; cf. e.g. Cohen et al. (2004)) in policy making (see e.g. evidence-based governance in popular literature) more often works against experimentation than in favour of it. What happens in reality is that public administration is pushed to use statistical correlations discovered by data mining on data it collected in processes that are already targeted towards suspicious activity. The popularity of evidence-based practice underlines the importance of a methodologically sound justification of explorative monitoring policies, and we hope to provide methodological justification for both principled experimentation, and for using common sense rather than correlations to define suspicious behaviour.
Chapter 8

Knowledge acquisition from legal experiences: from narratives to agent-roles

Reliance on case studies and other narratives is definitively not strange to legal practitioners and scholars. Such legal narratives—e.g. court proceedings, or scenarios making a point about the implementation or application of the law—are just examples of the much wider domain of products of human narration. Legal narratives, however, are particularly interesting for our purposes because they are produced by components of a certain legal-social system, with the intent of transmitting—to its current and future components—relevant social behaviours and associated institutional interpretations. This phenomenon supports the fundamental importance attributed to narratives in any human social setting. But how can knowledge be acquired from narratives? Unfortunately, the narrative object in itself often consists of ill-defined and under-specified knowledge: many elements, for the economy of the interaction, are necessarily missing, there may be ambiguities, if not errors, etc. From this perspective, we would face many of the problems researchers in natural language processing are investigating. This is not in our scope. In this research we will neglect the “pure” narrative object, and concentrate rather to the meaning it transports.

Four assumptions provide a basis to our approach:

• a narrative synthesizes a subjective representation of the world, describing a story with elements that support its interpretation (the foreground).
• a story involves agents, i.e. components of a social system, provided with their own conceptualization (which may be wrong).
• the narrator holds a systemic representation of the story; if questioned, he can provide elements that are missing but necessary for its understanding (the background);
• when the narrator is not available, the interpreter/modeler can do this reconstruction, integrating common-sense or expert knowledge.

Analyzed under this frame, legal narratives are intended to bring part of the background (concerning and concerned-by legal institutions) in the foreground. When adjudicating,
a judge declares publicly how (and why) the legal system is responding to a certain case; considering a certain domain, legal experts provide past or hypothetical scenarios along with their institutional interpretation, and so forth.

The objective of this chapter can be resumed in translating such experiences of law in an adequate representational model. In 8.1 we define a conceptual framework. In 8.2 we describe in detail a scenario-modeling methodology which transform the interpretation of a given scenario in the correspondent agent-role scripts.

### 8.1 Conceptual framework

The contribution of this section aims to analytic clarification rather than automated reasoning. This is the reason why we preferred the term “conceptual framework” on “ontology”. Because we target complex and adaptive systems, we have chosen to take advantage of current practices and developments in multi-agent systems instead. Two main sources of requirements have been considered:

- **At conception level**, we need sufficient descriptive power and granularity to describe stories from a legal perspective: at least two epistemological domains of reference are naturally addressed: brute and legal reality;

- **At implementation level**, we implement a strong correlation between the components of the ontology and the components of the targeted multi-agent platform, so as to enable a relatively easy translation from stories to agent programs; the target platform chosen for this scope is Jason\(^1\), based on the language AgentSpeak(L), introduced by Rao (1996).

This section proceeds as follows. After presenting some foundational concepts, like objects and events (8.1.1), we define agents, i.e. objects provided with mentality, based on an intentional stance (8.1.2), and with social capabilities, constructed on top of an institutional stance (8.1.3), concerning also legal institutions. Finally, because characters can tell “stories” within a story, we integrate concept derived from narratology (8.1.4), also useful to identify the main components of a narrative. As final result, we construct the agent-role model as abstraction of individual characters/agents involved in a case.

#### 8.1.1 Foundational concepts

A perspective founded on common-sense, supported in cognitive science\(^2\) and also in legal core ontologies\(^3\), considers objects and events as the foundational basis of world representations. An **object** is an entity with certain properties, which may change in time: for instance, it occupies a certain space.\(^4\) An **event** is instead the manifestation of a process, i.e. a change occurring in a certain time and involving certain objects. Just like objects occupy space (in a certain moment), events “occupy” time (and a certain space, via the involved objects).\(^5\)

---

\(^1\)For further references, see Bordini et al. (2007)
\(^2\)See for instance Zacks and Tversky (2001)
\(^3\)E.g. Breuker and Hoekstra (2004); Hoekstra et al. (2007b)
\(^4\)Although, from an epistemic point of view, we do not always acquire and process all these properties.
\(^5\)An example: gravity is a process (described by the law of gravity), and results in an event with the fall of a stone. When there is no event (i.e. a sensible change), from a common-sense
Existence from knowledge: facts. Because of its systemic validity, the story model should reproduce a world during its animation, i.e., its execution. When it is instantiated in a specific computational reality, it has to show an objective and ontological dimension. Nevertheless, from an epistemic perspective, we do not handle directly objects and events. What we have are facts, i.e., reified statements about the world. They describe properties, attributions, and qualifications of single objects, e.g., *Post is a person*—person*(post)* in a Prolog-like form—or relations, for instance, between objects in an event, e.g., *Pierson shoots the fox*—shoots(pierson, fox). The same applies to events, e.g., *action*(shooting), *does*(pierson, shooting). or *happens*(shooting, t1) as in event calculus. However, whatever representation choice we make at epistemic level, we need an interface with the ontological one.

States and situations. Behaviour of systems is usually described as a sequence of states and transitions. The word “state” comes from the Latin status, past participle of stare, “to stay”, “to stay still”. As a matter of fact, we cannot recognize something that we are not able to discriminate from a continuous flow of events. In this case, however, we still recognize an ongoing transition.

Considering a system as an observed object, its states are properties, if we know them by design, or attributions/qualifications, if they are interpreted. In an ontological setting, however, the same object may belong to different (sub)systems, according to the “cut” or stance with which it is observed. The resulting set of states would be a cartesian product of sets of states, whose correlations may be of any type. Because this multiplication is indefinite, we usually do not handle an object or a system in function of its state of affairs (constructed by integration of representational models), but by its situation (defined by deletion: the actual state of affairs has been filtered depending on the relevant properties).

Transitions. Transitions require additional investigation. Let us consider two different descriptions of the same basic story: “8 am: it starts raining, 10 am: it stops” and “it rained yesterday”. The first is the result of direct observation (e.g., a log), the second reports a past experience. Neglecting the observer/narrator, there is only one system/object involved, call it world or environment. Look at Fig. 8.1. Introducing a visualization commonly used in digital systems, we consider facts (in which point of view, nothing occurs. This is because common-sense is basically pre-Newtonian: it does not consider that the stone is still on the table as result of an equilibrium between two opposite forces.

*This supports Quine (1985)’s proposal in considering events simply as objects.

*See for instance Shanahan (1999).
we include interpreted perceptions) as carried by signals, while events correspond to signal transitions. Consequently we recognize three types of transitions: starting (or activating, or enabling), terminating (or inactivating, disabling), and impulse (when the duration of the transition is irrelevant or omitted). Zooming out on the temporal axis, starting and terminating transitions join in an impulse (on the right). When impulses acquire a duration, for example asking the narrator more details, we are making explicit the existence of a sub-process. The transition therefore becomes a transient: we acknowledge an ongoing state associated to the event.

Taxonomies and mereonomies The previous observations unveils that the narrator takes specific categorical and compositional positions when describing the world. This concerns both space (for objects) and time (for events) dimensions. Consequently, we recognize that the model a story requires a taxonomy (associated to categories) and a mereonomy (associated to parts), both of objects and of events. The distinction between the two perspectives is important; in the first, the most fundamental predicate is is-a(instance, type), mapping instance/type pairs to boolean values. In the second, the related predicate is is-part-of(part, whole). They are associated to hierarchical classifications and decompositions of prototypes and of processes as well. In this sense, they are properties of the world in which the story occurs.

Existence from knowledge: dependencies A process, i.e. the (real) mechanism behind (actual) events, can be described using a causal structure, expressed with a conditional (if inputs, then output, or output, if inputs), or with a functional form (output = f(inputs)). An example of causal dependency expressed by a conditional is “if a stone is thrown in the air, it falls.” This formulation evidently overlaps with that for logical dependencies, as in “if Peter is the son of Paul, then Paul is the father of Peter”.

Conditionals related to logical implication, in the form of rules, are commonly used in Prolog and logic programming in general. They allow, for instance, to express a hierarchical classification of concepts, e.g. all foxes are animals: animal(X) :- fox(X). What causal dependencies entail however cannot be reduced to such simple structures. A possible way to overcome this limitation is given by the event calculus, which reifies events in facts. However, this approach, constructed on backtracking, is not the best suited for model execution, which is naturally disposed forward (the next step is decided depending on the current situation). An alternative solution is to introduce adequate operators (allowing and interacting with side-effects) in the language. This is the path followed by AgentSpeak(L), which extends Prolog, introducing event-condition-action (ECA) rules. These structures, distinctive of reactive systems, allow the representation of hierarchical decomposition of tasks, as we will see in the following section. Because when a ECA rule is triggered, it imposes to the system the sequential execution of a plan of actions, ECA rules may be used to model causation.

---

8An example: in our approach, if I see a running fox, it is like I receive, from some external cognitive module, the fact that the fox is running. This is equivalent to the result of interpretation of an external speech act. We may consider this as consequent to the principle of expressibility, assumed in Searle (1969).
8.1.2 Intentional stance

Within the class of objects, we recognize the class of agents. In addition to proactivity, agent theories usually ascribe them autonomy, reactivity and some kind of social ability. These four characteristics are the references of a very important human capability: mentalization, Fonagy and Target (1997). To mentalize is to create an internal representation of an agent (self or other), and to explain and predict his behaviour in terms of attitudes, for example using concepts like beliefs, desires, and intentions, constituting the intentional stance, as defined in Dennett (1987b). Furthermore, humans tend to mentalize not only individuals, but also natural phenomenon, animals, communities, organizations, nations, and cultures.

Beliefs In our simplified perspective, beliefs are expressed as facts. According the speech act theory, a belief has a mind-to-world direction of fit Searle and Vanderveken (1985b). The transformation from a communicated fact to a belief depends on the trustworthiness and the reliability associated by the listener respectively to the narrator, and to intermediate emitters and receivers. Perceived facts are usually transformed directly into beliefs.

Desires, goals The concept of desire has been long debated, from philosophy to psychoanalysis. In our framework, desires are considered only when they are reduced into some more concrete goal, or when we reflect on the irreconcilability of conflicting desires (or preferences, values, etc). Desires cover also the world of impossibility, or they could refer to the imaginary. From a communicative perspective, a desire normally expresses a yet to be realized state of affairs, i.e. a world-to-mind direction of fit. Within goals, we differentiate two main families. An achievement goal is a purpose toward which an endeavour is directed. It may be described in a quite generic way and it may not be strictly measurable or tangible. For instance: I want to live in a nice house. An objective or target is a sub-type of achievement goal, identified with a specific performance or result (as state of affairs) that one’s efforts or actions are intended to bring about. It is measurable or tangible, and compared to a generic achievement goal it is normally deployed in a shorter temporal term. For instance: I want to buy this house by the end of this month. A maintenance goal, instead, defines a state of affairs that should remain true, rather than a state to be achieved Duff et al. (2006). They are important: fundamental needs are often formulated as maintenance goals. For example, the need to breath, be healthy, or maintain one’s self-esteem. During our modeling experience, we noticed that maintenance goals can be decomposed following the PACK (Prevent, Acquire, Cure, Keep) framework Ogilvie and Rose (1995).

Intentions, actions An agent may have multiple goals, but it will have always limited resources. The discrimination between active and non-active goals is managed by intentions. Thus, intent means conscious purpose, and intention is the commitment to be successful in bringing about that intent.

An action is a process driven by an agent and subsumes an intention. On the other hand, a reaction may involve no intention: a volcano erupting is a phenomenon resulting from physical laws. The performance of an action results in an act, term used both for the event (as occurrence of the action) and for the possible object (as

---

9See Wooldridge and Jennings (1995).
occurrence of the product of the action). If the product exists in the physical reality it is called **artifact**.

There is a strong link between actions, plans and goals. An action is a process whose procedural characterization (or plan) remains implicit, and it is symbolically synthesized by an **immediate intent**, resulting from an **immediate goal** (the result of the action) and an **immediate intention**, allowing its execution (its performance). Following the analysis of Pierce, Sowa Sowa (2000) observes that “with different verbs, the same action could be described by its form (Firstness), by its effects (Secondness), or by the agent’s intentions (Thirdness)”. The last determination reveals the presence of a hierarchical task (de)composition, structured in the language, that intervenes in the processes of generation and interpretation of an agent’s behaviour.

**Intentional causation**  
Rationality is commonly defined as the ability of the agent to construct plans of actions to reach a goal, expressible eventually as a hierarchical task decomposition. In our case, agents do not deliberate. Choices have been already taken: the decomposition occurs mostly in the mind of the interpreter, helped by traces in the narrative and according his own conceptualization of the world.

The ECA rule of the activation of a goal, in words, would be: *in order to reach the goal, if certain conditions are satisfied, you perform this plan of actions*. The equivalent code would be something like:

```prolog
+!goal : conditionA & .. & conditionZ
<- actionA; ..; actionZ.
```

Conditions represent what the agent thinks should be true, in that very moment, in order to be successful in executing his plan. Stated differently, their evaluation furnishes the perceived **affordance** Gibson (1979) of that specific behaviour. Because affordances depends on the environment and on the agent, we may expect conditions that refer to both of them. Interestingly, discovering this connection provided us with a point where to start the decomposition. Let us take the case story presented in section 8.2.1. We recognize in the source text several verbs. Let us write, very briefly, which conditions, in this limited world, allow or support these actions:

- you can hunt the fox if you are seeing it
- you can catch the fox if you are near it
- you can intercept the fox if you are near it
- you can kill the fox, if you have intercepted it
- you can take the fox if it is dead

The use of the verb “can” is highly ambiguous: permission, possibility, ability, knowledge, etc. In this case, however, our only concern is on the chronological sequence: the precondition should be satisfied before performing the action. Obviously, other plans may change these phrases. For instance, catching the fox using traps does not require to be near the fox. But this does not occur in our story. In the previous sentences, we have introduced two other predicates, which require further specifications:

- you will be near the fox, if you approach it
- you can approach the fox, if you are seeing it
- if you kill the fox, it dies

The first is an example of causal dependency, the third of logical dependency.
The hunting of Post was interrupted before the catch, while Pierson’s was successful and structured in the sequence: intercept, kill and take. The following excerpt serves as an example of the translation of the previous analysis. Considering Pierson, we have:

```
+!hun(E) : seeing(E) <-
  +hunting(E);
  !intercept(E); !kill(E); !take(E);
  -hunting(E); +hunted(E).
+!intercept(E) : near(E) <- +intercepted(E).
+!intercept(E) : not near(E) <- !near(E); !intercept(E).
+!near(E) <- !approach(E).
+!approach(E) : seeing(E) <- +approached(E).
near(E) :- approached(E).
```

Physical acts are implemented as broadcast speech acts. This is not always fit, because with larger time scales individual interpretations of actions may diverge, as perception becomes abduction Shanahan (2005). In the case of administrative activities, however, atomic actions are always speech acts.

**Mens rea** As elements used by humans to describe their behaviour, intentions are taken into account by legal systems, although the debate about their integration has been continuous in legal theory and philosophy (see for example Hart (2008)). From a legal perspective, a person commits a criminal act with intent when the conscious objective or purpose of that person was to engage in the act which the law forbids. On the other hand, a motive is the reason why a person chooses to undertake a criminal conduct, and generally is a fact acknowledged by the agent. A possible definition of motivation is then the mental entity that, triggered by the motive, produces a (criminal) intent. Interestingly, motivation could be associated to maintenance goals. For example, if someone is caught while trying to stab a person, his intention of killing is explicit. The reason behind that intention is more difficult to investigate. If we discover that this person was offended previously by the victim, we may plausibly assume that he has executed a plan “curing” his “honour”. This investigation is relevant to factual reconstruction, but evidence of a motive, or evidence of the lack of a motive, may also be considered by adjudicators in the resulting legal interpretation.

### 8.1.3 Institutional stance

“In court proceedings, the ‘truth’ of the facts is ultimately determined not by criteria employed in empirical sciences but by those provided by procedural and substantive legal norms” Tuori (2006). In legal discourse, all facts are institutional.

An institution is an intentional social collective entity: collective because a group of people recognizes it, intentional because they intend its existence, and social because it determines aspects of their interactions. This definition unifies games, social informal norms and legal norms. Although defined as collective, an institutional
reality is not directly shared by the agents, but exists and evolves in their individual mental domains. The collective flavour of this type of entity is given by a similar representation in each individual of the community, or, better, it may be regarded as an instance of emergence.

Proceeding along the work of Searle (1969), an institution is defined by certain institutional rules and some institutional facts. Some of the institutional rules are constitutive rules, because identify which facts (including acts) belonging to the brute reality count as institutional facts. The whole process is illustrated in Fig. 8.2. An important type of constituting act is the declarative act, which represents the double direction of fit (world-to-word-to-world). This is the case of a judge that delivers a sentence to a criminal, declared guilty by a jury. His declaration alters the world, because, if his opinion is upheld, the world must be altered to match the content of the declaration. Fundamental classes of institutional facts are institutional identities (identifying a person, physical or legal), institutional roles (for example, in the sale institution, buyer and seller) and jural relations (obligation, right, etc.). Jural relations are bound by deontic rules. The introduction of a new jural relation triggers a jural reconfiguration.

Let us consider our case story. Tompkins’s main point can be reduced to:

if an animal does not belong to anybody
then any person has the power to start owning it via capturing it

The power we are referring here is one of the fundamental legal concepts Hohfeld (1917), and, specifically, an action-power Sartor (2006b). It implements the institutional recognition of a change at an abstract level, and it is directly related to constitutive rules. Consequently to this rule, the “hunting” institution acknowledges individuals classified as animals and persons, facts about owning, acts of capture, while it generates facts about owning. So far, however, owning is only a empty label. We have to add also some institutional rule for property. Livingston’s point can be

---

13 For instance, if a person owns something, he has an exclusive right on it.

---

188
translated in a similar way.

In our current implementation Sileno et al. (2013), we include an institutional logic based on Sartor (2006b), but other choices (wrong conceptualizations too) are possible as well. See in chapter [XXX] for the actual code.

In order to produce a normative action, jural relations instantiated in the institutional system should have some consequence on the behaviour of the agent, for example as normative indications to be considered during his decision-making cycle (allowed or not, obliged or not)\(^\text{14}\). These indications refer not only to the self, but also to other agents\(^\text{15}\).

Social agents refer to expectations when interpreting other agents’ behaviour, also when monitoring compliance. In our case, each judge exchanges with two institutions (property and hunting) all the facts that are relevant. The result is an ex-post normative indication about the acts performed by Post and by Pierson. Because their hunting institutions have different rules, Tompkins and Livingston’s opinions dissent.

8.1.4 Narrative realm

Functions, roles Removing all verbal considerations, Propp (1968) analysed Russian traditional folk-tales. He introduced the concepts of narrative functions, in the sense of atomic, functional components of a story, and roles, connecting those functions. Barthes extended this approach Barthes (1966), observing four categories of functions. The first two are distributive and express the sequential distribution of the events in the story; the other two categories are integrative: not necessary for the sequence, but they increase the detail of representation for the sake of the interpreter. Translating his approach into our framework, we recognize these correspondences:

- **primary/secondary functions** are facts reporting events (acts), critical or not in respect to the causal structure of the story
- **clues** are facts describing mostly qualifications or attributions, or rules belonging to a certain system (physical, mental, social, ..).
- **informants** are facts describing properties; some rules may be activated by certain informants, in which case they are also clues.

Story, discourse, conversation The story is only one part of the narrative phenomenon. A narrative presents always three ontological levels Young (1984), corresponding to the *what*, the *how*, and the *why* associated to the *speech act*. The narrative content is expressed by a discourse, and, once interpreted, constructs a story. The discourse defines the order and the form (verbal and non-verbal) in which the content is provided. The conversation relates the discourse in a certain social context. As any other act performed by an intentional agent, also a speech act subsumes certain intentions, i.e. individual commitments triggered in that specific context. An agent may intend to share his knowledge, but he may also desire to alter the world, and he may do that performing a specific speech act, when he knows that will produce

\(^{14}\)There is a direct connection between these normative indications and the four types of obligations described in Sergot and Richards (2001b).

\(^{15}\)An example will show the potential orthogonal utilizations of these normative indications. When driving, we may stay on the right lane of the road because we are obliged to (this would be a deontic perspective: norms are followed because they exist), and because we expect the others to drive on our left lane (a consequentialistic position: norms are followed because of the consequences they create).
institutional changes. A witness may have interest in lying, or to give only a partial truth. A public officer can use its institutional power for his personal agendas.

In our case story, the more external narrative container is the source text, probably written by a court stenographer. The factual case, taken from a previous appeal, was probably dictated by some other officer. Assuming public officers and stenographers to report with no errors and with a purely informative intention, their conversation level can be neglected. A similar consideration may apply to the judges: their institutional role is to apply the law, and their position should be neutral in respect to their own or other’s interests, so that the “story” they are telling is the story from the perspective of the legal system.

Agent-roles The definition of agent-role links the concepts of narrative role, institutional role, and intentional agent. Agent-roles are entities identified with a set of abilities, a set of susceptibilities to actions of others, and associated with goals, plans, and beliefs typical of the role they have Boer and van Engers (2011b). An agent-role is embodied by network of physical agents. It can be virtually decomposed by (at least) a coordinator agent and an agent for each institutional role involved in the scenario, so as to reduce the social behaviour in social primitives. In addition to this internal topology, we recognize an external topology, referring to the social environment (e.g., a buyer knows at least one seller). This external topology represents awareness of the other agent-role, and, therefore, the ability to engage in a role-appropriate interaction with that agent-role. With our approach, what we grasp from a story is actually an agent-roles configuration: a system consisting of agent-roles and topological connections between them.

8.2 Acquisition methodology

Reducing the problem to the core, we propose hereby a methodology to transform a sequence of inter-agent interactions in intra-agent characterizations, reproducible in a computational framework. In section 8.2.1, we present our case study. We analyze it at a signal layer and we define the topology and the flow of the story. In section 8.2.3 we show how to enrich the previous representations with an intentional layer, integrating institutional concepts as well. In section 8.2.4 we provide elements about the transformation of the previous models into scripts for cognitive agents.

8.2.1 Inter-agent description

Despite its simplicity, a short story about a sale transaction provides a good case of study.

A seller offers a good for a certain amount of money. A buyer accepts his offer. The buyer pays the sum. The seller delivers the good.

A successful sale is a fundamental economic transaction. Consequently, what the case describes is a behavioural pattern used both in the performance and in the interpretation of many other scenarios.

\[16\] This is a strong assumption. Legal realists, on the contrary, affirm that the institution of law is strongly influenced by moral, political, and social conflicts.

\[17\] Thus, topologies are the first place where to monitor agent-roles expectations.
8.2.2 Signal layer

The story describes four events, namely four acts performed by two agents. The first two acts are easily recognized as speech acts, but all these actions can be teleologically interpreted as bringing some informational change into another agent. From this perspective, we can consider all of them as acts of communications, i.e. as messages going from a sender to a receiver entity. Thus, the previous story can be illustrated using a communication diagram, for instance as a message sequence chart (MSC).

Simplifying the notation, we obtain the illustration in Fig. 8.3.

So far all seems easy, but the sale process described before lacks some important details. For instance, in a marketplace, paying and delivering are physical actions. They produce some consequence in the world: money, goods move from one place to another. Furthermore, payment and delivery are acknowledged by perception. Second, the buyer has accepted the offer, because he was somehow receptive toward that kind of messages. Third, a buyer who already paid usually does not leave without taking the good, just as the seller does not allow the buyer to go away with an object without paying. In our story, all goes well, so the narrative does not provide any element concerning these checks. However, this does not mean necessarily that the buyer and seller had not checked if everything was fine. Fourth, sometimes a buyer takes the good and then pays, sometimes the order of actions is inverted. These four points reflect characteristics which are left implicit in the story, and, consequently, in the MSC:

- acts have side-effects on the environment (at the very least, a transient in the medium transferring the signal),
- an action consists of an emission (associated to an agent) and of a reception (associated to a patient),
- certain actions have a closed-loop control: agents perform some monitoring on expected outcomes,
- the sequence of events/acts in a story is often a partial order, hidden by the linear order of the discourse.

There is also another implicit agent, the narrator, but he will be neglected in this work, supposing he has a pure descriptive intention.

MSCs are the basis of the sequence diagrams, one of the behavioural diagrams used in UML. For further information, see for instance Harel and Thiagarajan (2004).
Apart from the last point, which could be solved with the UML “par” grouping for parallel constructs, we have to find alternative representations to help the modeler in scoping and refining the content of the story.

**Topology of the story**

Inspired by the Actor model Hewitt et al. (1973), we have drawn in Fig. 8.4 the topology of the story. The topology serves as a still picture of the whole case, and show how signals are distributed between the characters. The little boxes are messages queues, the lines identify communication channels. The story describes which specific propositional content is used in the exchanged messages. In order to take eventual side-effects into account, we introduced an explicit “world” actor, disjoining the emission from the reception. The optional part of the communication is visualized with dotted lines. The world would play as intermediary entity also in case of broadcast messages.\(^\text{20}\)

**Flow of the story**

Orthogonal to the topology, we define the flow of the story as the order in which events occurred. As a first definition, we may consider a story as a chain of events (a strictly ordered set):

\[
E = \{e_1, e_2, ..., e_n\}
\]

(8.1)

In narratology this layer is usually called the *fabula*: “a series of logically and chronologically related events [...]” Bal (1997). This name dates back to Propp, which, altogether with the Russian formalists, started considering each event in the the story as *functional*, i.e. necessary to bring the narrated world from initial conditions to a certain conclusion. In addition, specific circumstances may be described in correspondence to the occurrence of an event. As a result, a story corresponds to the following chain:

\[
C_0 \xrightarrow{e_1} C_1 \xrightarrow{e_2} ... \xrightarrow{e_n} C_n
\]

(8.2)

where \(e_i\) are associated to transitions and \(C_i\) is a set of conditions assumed to continue at least until the occurrence of \(e_i\).

**Consequence and consecutiveness**

This definition may look very simple, but the manifold relations between consequence (logical, causal, ..) and consecutiveness (informed by time, ordering, ..) are actually very delicate to assess. In addition, two different chronological coordinates coexist in a narrative: a *story-relative* time, i.e.

\(^{20}\)With a similar spirit, communication acts performed autonomously by the world actor can model natural events.
when the event has occurred in the story, distinct from a discourse-relative time, i.e. when that event has been reported or observed.

In order to unravel this knot, we use a four steps methodology to reconstruct the relations between the elements of the story.

**Strong constraint**  First, we elicit relevant abstractions which are used in the interpretation. In particular, we define an event/condition as free if the interpreter does not acknowledge any relation with another event or condition in the story. We refer to such relations as dependencies. Some dependencies are syntactic. For instance, you can accept an offer only if there is an offer, i.e. if an offer has been previously made. Others are contextual to the domain. For instance, in a web sale, payment usually occurs before delivery. In all cases, dependencies can be used to put a strong constraint on the ordering of events.

**Medium constraint**  Second, there may be clues of the story-relative time within the text. Time positions and durations are usually meant to give some landmark to the listener. They are described in absolute or relative terms. When a listener interprets them, it creates a relation between events, contingent to the story. Such relative positioning constitute the medium constraint.

**Weak constraint**  Third, if we have no clues about dependencies, or temporal relations between events, a possible sequence is at least suggested by the discourse-relative time. This provides a weak constraint on the ordering of free events.

**Application**  If all three constraints are satisfied, we do not expect any concurrent events, at least within one story frame. However, it is easy to object to such a strict determination.

Consequently, at the fourth step, we weaken the previous strict temporal constraints (e.g. from $e_{i+1} > e_i$ to $e_{i+1} \geq e_i$) in two cases: (a) dependencies can be associated to no-time-consuming processes (e.g. logical equivalences); (b) events may occur simultaneously, when triggered by parallel sub-systems. Furthermore, the medium and weak constraints refers to contingent relations (according to the modeler). In order to be able to compare the internal structure of stories, we can neglect them. With these modifications, the set $E$ defined in (1) is a partially ordered set.

Let us take our case story. There is a relation of syntactic necessity between offer, acceptance and performance. In addition, there are two agents. These entities can be considered as parallel systems, that may concurrently interact with the world. Therefore, without further contextual specification, payment and delivery are concurrent events.

**Visualization**  A simple way to visualize the flow of the story is by the use Petri nets, as we did in Fig. 8.5. We opted for a practical naming of places: $sender>receiver:content$. At this point, places represent messages, associated to speech acts. Actions are like “compacted” into transitions.

---

21 Apart having occurred in the same story.

22 The story and discourse contingencies of the medium and weak constraints become contextualities if they are entailed by strong constraints.

23 A more complex story may consist of many frames. For “frame”, we consider a sub-story that follows the Aristotelian canon of unities of time, space and action.
The main scope of the flow is to preserve the *story synchronization*. Further layers may be integrated, increasing the granularity of the description, maintaining the previous points of synchronization, in the same spirit of *hierarchical Petri nets*, see Fehling (1993). For instance, in Fig. 8.5, we have disjoint the generation of the message from its reception using an intermediate “world” actor, as we did in the topology.

### 8.2.3 Agentic characterization

In the previous section we referred to the messages exchanged in a social scenario, and narrated through a story. When interpreting them, however, we apply an intentional stance just as we do in our experiential life. We read the actors as intentional agents, attributing them beliefs, desires and intentions. An *ex-post* intentional interpretation of the story results in a hierarchical decomposition of the plans followed by the agents. In addition, in order to trigger or enable the performance of the reported acts, there may be other relevant conditions or hidden acts to be taken into account. They could have been left implicit by the narrator, but plausibly they are known to the target audience of that narrative. This is the main assumption on which our work is based: any reader/modeler can always provide one or several reconstructions of what happens behind the signal layer, supported by common knowledge and domain experts (at least, interviewing the narrator itself, if needed).

### Acquisition methodology

As we did before, we start the acquisition using a MSC diagram. A possible outcome of the interpretation is in Fig. 8.6, introducing adequate extensions. First, we consider externalized intents (with a “!” prefix) as the events triggering the processes of buying/selling. The final outcomes of those actions are then reported with output messages as well, at the end of the chart. Second, we use the *critical* grouping to highlight which conditions (in addition to sequential constraints) are necessary for the *production* of that message. In our story, we add that: (a) the buyer performs an evaluation of the offer, (b) the buyer accepts the offer if it is acceptable for him, (c) the buyer pays (the seller delivers) if he owns the requested money (good).
The diagram in Fig. 8.6 furnishes a good summary of the story: the inputs/outputs provide an intentional characterization, the vertical bars indicate the ongoing activities, while the messages refer to successful acts of emission and reception, whose occurrence is constrained by the critical conditions. However, further refinement is necessary to cover the basic figures encountered in an operational setting. There is no separation between emission and reception, and between epistemic and ontological. For instance, we cannot distinguish between a case in which the buyer thinks he has not enough money, and another in which he thinks that there is enough on his bank account, but the bank does not “agree” with him.

A simple solution to this problem would be to add intermediate actors, localizing where the failure occurred. In a complex case, however, the resulting visualization may be overloaded. In the following sections we will therefore introduce some patterns to be attached to the flow of the story. Instead of using just one visualization, our approach aims to provide alternative representational cuts.

Figure 8.7: Hierarchical decomposition of tasks.
Hierarchical tasks

First, we elicit the hierarchical decompositions of activities performed by the agents. These serve as basic schemes for the behavioural characterization of the agents, and use hierarchical, serial/parallel constructs. In practice, this is obtained by identifying, in the story flow, the activities of the actor as agent (emitter) and patient (receiver), and relating them according to their dependency. Fig. 8.7 reports the result of this step for the buyer.

Emission and reception

Second, activities are anchored to messages. This is a delicate phase: we want to maintain the synchronization given by the story and the dependencies associated to the activities. Fig. 8.8 reports our solution (applied on a single message) which explicitly divide emission from reception. The proposed Petri net is complete and well-formed. It is scalable to multiple agents, adding a reception cluster (e.g. b>content, perception, etc.), in order to connect the message to each agent that is reachable by the communication.

Note: partial orderings may hold independently in the story flow and in the activity diagrams. In this case, for instance, we do not know a priori if payment occurs before delivery (as acts), as we do not know if the buyer pays before monitoring the delivery or vice-versa, or simultaneously (as actions).

Illocutionary acts

Third, we recognize the practical effects of messages. Beside of being signal (or a locutionary act), each message is associated to an illocutionary act, and then, when put in a computational form, should integrate some pragmatic meaning.

For simplicity, we consider only four types of performatives: assertions, commissives, commands and inquiries. Moreover, we interpret commitments as obligations to the self and commands as attempts to instill obligations into the receiver.footnote{This is an operational simplification. Cfr. Kurji (2012) for a recent summary of the philosophical accounts on promises, obligations and commands.} Considering our story, we have:

- the offer is a conditional promise: the seller commits to deliver the good to a buyer who commits to pay his price;
• the acceptance is a promise: the buyer commits to pay a given amount;
• the payment can be interpreted as a command performed by the buyer to the world, plus an assertion performed by the world to the seller;
• the delivery can be modelled similarly to the payment.

Action and power
Fourth, activities are used as anchors for cognitive, motivational and institutional elements, informed by the illocutionary content of the messages. Once we have constrained the external behaviour, we integrate the previous layer with an internal behavioural characterization of the participants in the story. Because we are considering intentional agents, we know that each action presupposes an intention, persisting at least throughout its performance. In addition, intents may be nested: in order to achieve a goal, the agent can start to bring about an associate sub-goal. In the sale story, we identify the agents as buyer and seller since the beginning, so that we know from the start which are their intents. Nevertheless, in other stories, the real intents may be unveiled only at the end of the narrative.

Critical conditions As we have already noticed when drawing the MSC diagrams, in order to trigger the performance of the reported acts, there may be other conditions or hidden acts to be taken into account. For example, a buyer usually accepts an offer only after positively evaluating it and should own at least the requested amount of money. Analyzing these conditions in detail, we observe that the acceptability condition reflects the possibility of the buyer to close what he considers an acceptable deal. It imposes a refinement to a generic buyer script, and it depends on the economic skills of the buyer in assigning and handling attributes and features of the market (average price, scarcity, competitiveness, compliance, ...). The ownership condition is

---

25 The absence of evaluation is symptomatic of a combine scheme: the buyer performs mechanically the acceptance in order to advance the interests of another element of his social network.

26 Ownership, or more in general, control, may be translated in certain contexts as having physical possession. In other cases, it may involve communication with third parties, like banks, warehouses, etc.
instead a precondition to successfully perform the payment. They are both examples of conditions critical for the story to occur.

Such critical conditions are in general associated to the ability or, more in general, to the power of the agent, in a specific context agent+environment. They identify which propositions should be true in the story-system, so that the agent is successful in the performance of the associated action. The subjective evaluation of each of these conditions gives the affordance (Gibson (1979)) of that behaviour, as perceived by the agent. However, if the affordance is a sufficient condition for the performance of the action, it is only necessary for the intended outcome, where the contextual disposition plays a role (cfr. Chemero (2010)).

Action patterns In section 8.2.2 each place of the Petri net was labelled with a message, i.e. a reified expression of the act. We extend that representation adding, before each act, a place referring to the performance, expressing the ongoing state of the action. When the token passes through the input (output) arrows, the action is starting (ending); the whole passage corresponds to the occurrence of the event. Similarly, we add a anticipatory place for the intention associated to the action. If intentions are nested we report them respecting their hierarchy. To complete the view, we add the motivation place, for the mental entity that makes the agent sensible to a certain fact, which is the motive for action. The resulting template is in Fig. 8.10a.

At this point, we integrate on each level the places for critical conditions, in terms of affordance and disposition, and places for motives, according to the pattern on Fig. 8.10b. In respect to a specific action, the motive is a fact acknowledged by a agent, starting the cycle of practical reasoning which brought to the performance of that action. The affordance may be seen as an expression of the contextual perceived power of the agent, while the disposition synthesizes the contextual actual power, where the context agglomerates the agent and the environment. If affordance and disposition are equal, this means the agent has a perfect knowledge-of and control-on the relevant

---

198

---

properties of the context.\textsuperscript{28}

\textbf{Recontextualization} The picture provided by these patterns is not yet complete. For instance, an action may produce side-effects, an intention is usually maintained until the intended objective is reached, etc. For each event in the story, we need a \textit{recontextualization} pattern to model the emergence of a difference in structures or relations between contextual elements (belonging to ontological and epistemic realms). This mechanism is a technical solution to allow the representation of the dynamics of concurrent processes and of persistent states. In Fig. 8.10c we provide some elements about how recontextualization may be represented using Petri nets. We exploit some of the intuitions introduced with classic planning operators (e.g. STRIPS), and in biological modelling Chaouiya (2007): catalyzers are necessary entities, but not affected by the transformation, inhibitor arcs have to be connected to empty places in order to enable the firing of the transition.

\textbf{Institutional characterization} What occurs when conditions are not satisfied? For instance, if the buyer does not own the agreed amount of money, he won’t be able to pay. The incomplete fulfillment of an agreement normally entails some social consequences. On the contrary, the “acceptable

\textsuperscript{28}Generally speaking, an affordance has components expressed in an \textit{analytic} form (e.g. we cannot drink from a glass if there is no water), but it mostly refers to \textit{synthetic} propositions (e.g. the “drinkability” of a container is a \textit{feature} recognized via heuristics learned by experience), which overcome non-systemic and non-deterministic aspects of reality. In this respect, narratives act as bridge to symbolic expressions, passing over the synthetic act; the relevant factors are reported, if they are critical to the explanation of the story.
offer” condition, although critical to the story, can plausibly be false, without creating any failure according to social norms (in terms of contract). The difference between these conditions can be expressed integrating an institutional layer. In particular, we refer to the fundamental legal concepts that compose the hohfeldian squares Hohfeld (1917), shown on Fig. 8.11.

The seller promises to deliver a certain good to a buyer, if the buyer will in turn promise to pay a certain amount of money. Offering is a conditional promise, while acceptance is a direct promise. With the offer, the seller declares to be subjected to the buyer’s power to accept, whose action would trigger an obligation to deliver. On the other side, with the acceptance, the buyer also creates his own obligation of paying.\(^{29}\) In order to be able to promise, however, the seller and the buyer need to have the power to make these promises, i.e. to have the power, in respect to a social (sub)system, of changing the associated normative positions.

The use of the term disability in the second square suggests that, in this context, it is common to use ability as synonym for action-power. However, in both intentional and institutional domains, depending on context, the term power may refer to ability, to affordance and to disposition.

**Directed obligations and power** A deontic conceptualization of norms typically emphasizes that norms are in itself a reason for action. An agent should immediately execute a certain action if an obligation in such respect is formed. Conversely, a consequentialistic approach would add some form of evaluation associated to the obligation, before generating the intention. Both perspectives are combined in Fig. 8.12a. People comply with obligations because they have some motivation to do so. For instance, it may be for respect to authority, habits, or for fear of reinforcement actions. Permissions are instead processed in the evaluation of (perceived) institutional power. If the agent does something even if he is not allowed to, this means that he considers himself immune from the obligation of not performing the action. This introduces us to another perspective on obligations. In order to be relevant, any obligation should be associated to the subjection to be sued, when it is not respected. This is true also in case of informal social institutions, where the subjection may refer to actions expressing social blame (and aiming to construct social shame).

**Power** The hohfeldian squares show how the fundamental legal concepts bind two institutional actors. In the common use, however, these concept are usually interpreted with a teleological characterization Sartor (2006b). In this line of thought, it makes sense to analyze the concept of power in the prevent-acquire-cure-keep (PACK) framework Fonagy and Target (1997), which translates Skinner’s operant conditioning concepts into motivational terms.

An action results in a successful outcome only if the agent has the related action-power. Power is acquired (lost) via an adequate enabling (disabling) act, which may also be performed by another agent (see Fig. 8.12b).\(^{30}\) Integrating the PACK framework, we obtain the analysis illustrated in Fig. 8.13 (\(s\) stands as self agent, \(o\) as other

\(^{29}\)A natural strategy to avoid failures is that the agent performs a check on (expected future) affordances before acquiring obligations. In doing that, all critical conditions of potential obligatory actions are inherited up to commitments. In the sale story, the buyer should check if he owns the money before accepting.

\(^{30}\)For a different perspective on the same subject, cfr. the role enacting/deacting solution presented, amongst others, in Dastani et al. (2004).
In total, we have four possible actions: **acquire action-power = cure disability** corresponds to an enabling action; **acquire immunity = cure subjection** means to go outside of the sphere of action of another agent (for instance, in order to make obligations not relevant); **keep action-power = prevent disability** can be achieved in three ways: (i) inhibiting the disabling power of other agents, (ii) preventing others’ intentions to perform a disabling action or (iii) preventing a positive evaluation of the affordance associated to the disabling action; **keep immunity = prevent subjection** is dual with the previous case.

Let us consider some example of its application. A group of people, including the owner of a confiscated real estate and someone internal to the public administration, wants to take advantage of a public auction to gain the property back. Considering the power to acquire the property, they are in a position of immunity in respect to other competitors, because they are the only ones to know about the auction. In order to keep this immunity, they may: hide the publication of the auction, provide people with false information (influencing their affordance), or threaten/bribe them so that they prefer to renounce to take part in the auction (inhibiting their intention). A fraudulent seller that uses an identity outside the legal jurisdiction is instead keeping immunity by inhibiting the disposition factor.

**Failures**

In consequence to the introduction of the institutional layer, we are able to model the occurrence of a social failure. Any action, however, may produce results that differ from expectations. Considering for instance a physical action, if it is performed in a wrong disposition, we observe a misalignment with the *immediate intent* related to that action. This failure may be due to an incorrect evaluation of affordance, or,

---

Preventing actions are visualized as inhibitor arcs, without making their description explicit.

---
outside the narrative world, to non-deterministic factors. It is perfectly plausible that an ability taken for granted may sometimes fail. For example, even if we are able to drink, we may let the water go down the wrong way.

Non-deterministic factors can be modelled as free events, in the sense that their occurrence is not explained by a causal reconstruction. A related Petri net model is shown on Fig. 8.14a. It is similar to the normal action pattern. The wrong result is made explicit in the net, as it is statically settled in the story.

A slightly different case of failure is the one occurring because of timeout (Fig. 8.14b). Because we know that the transition on the right will never be fired, we may omit to report it in the Petri net. The timeout place is a synchronization place, defined in relation to other events in the story.

Consequently, in our representation, we refer in total to four layers, each of which addresses specific components:

- the signal layer — acts, side-effects and failures (e.g. time-out): outcomes of actions
- the action layer — actions (or activities): performances intended to bring about a certain result (the action layer)
- the intentional layer — intentions: commitments to actions, or to nested intentions,
- the motivational layer — motives: events triggering the creation of intentions.

The closure of the sensing-acting cycle of the agent is guaranteed by the fact that certain signals, when perceived by agents, becomes motives for action. But how the motive is translated in intention(s)? How an intention is transformed in action(s)? What permits that an action effectively produces a certain result?

These questions can be answered introducing additional elements, dual to the previous components, such as:

- dispositions: contextual alignments of the agent with the environment (consisting of other agents and of the world actor) in respect to the actions he performs,
- affordances: perceived alignments of the agent with the environment in respect to his intents,
- motivations: mental states catalyzing the creation of intentions.
Affordance and disposition  Learning is generally a process that helps the agent to select which action are successful or not. If the agent thinks that the environment affords his behaviour, he also thinks he has the power to achieve that goal associated to that behaviour.\textsuperscript{32} Affordance practically corresponds to perceived power. Differently, disposition is connected to actual power: it is a precondition to the modifications that a certain action of the agent will imply.\textsuperscript{33} In a general case, if the buyer starts buying, he is assuming he has the power to do it. Despite of his intent, however, the seller may have the intention of getting the money without delivering. As a result, the disposition for a successful transaction would be false, blocking the normal completion of the sale.

In the light of this analysis, we observe how these categories corresponds to the critical conditions reported on the MSC (e.g. acceptability, ownership in Fig. 8.6). The subjective evaluation of each of these conditions gives the affordance of that behaviour, as perceived by the agent. However, if the affordance is a sufficient condition for the performance of the action, it is only necessary for the intended outcome, where the contextual disposition plays a role.

Motivation  In our framework, motivation refers to some mental condition that makes the agent sensible to a certain fact, which becomes the motive for starting an action.

Visualization  To sum up the concepts introduced so far, we have reported in Fig. 8.9 a possible reconstruction of the step of payment.\textsuperscript{34} The triggering motive is the reception of the acceptance. The illocutionary content of the offer entails the duty to pay of the offeror, if someone accepts. This duty to pay is followed for instance if the buyer desires to be compliant. An obligation is then formed and used to construct the correspondent intention. The intention, if there is an action or a course of actions (i.e. plan) which is afforded by the environment (e.g. the buyer thinks he owns enough money), supports the selected performance. Finally, if the action is performed in a correct alignment (e.g. the buyer owns enough money), it results in the expected act.

Model validity

Each observed scenario can be explained with a set of possible interpretations. The story provides only a foreground, which the interpreter can anchor to alternative backgrounds, explaining and completing it further. Our focus, in this frame, is not the “scientific” validity of the systems integrated in the background, i.e. how far they represent what is the case in the experiential world. The choice of backgrounds is

\textsuperscript{32}Assuming an agentic perspective, we could say there is no behaviour without intent. This assertion has no relation with consciousness, however. Intent can be implemented by design. A certain tool, or even a business process, is designed to achieve a certain goal in a certain environment, but artefacts or processes in themselves are not aware of such goal or environment. Intent and affordance may transcend actual performance, but they still exist.

\textsuperscript{33}For a recent overview on the theories of affordance, see Chemero (2010).

\textsuperscript{34}To reduce the visual burden, we have combined the layers action, intentional, motivational in one agentic layer. Nevertheless, it would be sufficient to add starting and finishing synchronization places for each layer to have a complete multi-layered model. The picture maintains in fact the vertical organization of our conceptualization: it is easy to recognize to which layer each element belongs.
informed by the ontological commitments of the interpreter (e.g. judges giving alternative interpretations to a legal case). In our perspective, if an interpretation produces all the messages provided in the foreground, then, such model is valid.

8.2.4 Intra-agent synthesis

In the following section we provide elements of the integration of our framework with current practices in MAS.

Agent-roles

Rationality is commonly defined as the ability of the agent to construct plans of actions to reach a goal, possibly referring to a hierarchical decomposition of tasks. In our case, agents do not deliberate, or better, the decisions they deliberate have been already taken. Consequently, their behaviour is fully deterministic. Instead of considering a full account of agency, we opted to base our framework on a more constrained concept: the agent-role, which integrates the narrative and institutional role concepts in an intentional entity.

Agent-roles are self-other representations, i.e. used to interpret, plan or predict oneself’s or others’ behaviour. They are indexed by: (a) a set of abilities, (b) a set of susceptibilities to actions of others. In a social scenario, both abilities and susceptibilities become manifest as messages exchanged with agents playing complementary agent-roles. Thus, the topology of a scenario (e.g. Fig. 8.4) provides a fast identifications of these indexes. Differently from objects (and actors), however, agent-roles are entities associated also to motivational and cognitive elements like desires, intents, beliefs and plans.

From scenarios to agents

A scenario agent is an agent embodying an agent-role. It is deterministic in its behaviour: all variables are either set, or determined by messages from other agents. This determinism, consequent to the internal description of the agent, constrains the temporal order of messages between agents, but not the behaviour of an entire multi-agent system. In sections 8.2.1 and 8.2.3 we have constructed and used a message layer in order preserve the synchronization given by the case. If we remove this layer, the system becomes non-deterministic. For instance, in a case in which there are two buyers and a seller, we do not know a priori who is the first buyer to accept.

Computational implementation

For its structural determinism, the behaviour of an agent-role can be described in terms of rules, translating logical and causal dependencies. Let us analyze the patterns used in the Petri net, for instance in Fig. 8.9. In general, transitions have input arcs connected to places that refer to an impulse and to continuing (for a certain time) conditions. This configuration easily relates our representation to event-condition-action (ECA) rules, which are commonly used in reactive systems. In respect to MAS theory, this connection has been exploited for instance in AgentSpeak(L) Rao (1996), a logic programming language for cognitive agents, later extended and operationally implemented in the platform Jason Bordini et al. (2007).

The connection of AgentSpeak(L) with Petri nets has been extensively studied in Behrens and Dix (2008), with the purpose of performing MAS model checking.
using Petri nets. In the present work, however, we have a different perspective: we started from a representation of the scenario on a MSC chart, we refined it with Petri nets patterns, and now we want to extract from this representation the correspondent agent-role descriptions (as agent programs).

**Components of AgentSpeak(L)/Jason** There are two main constructs in this language. The first is the ECA rule associated to the activation of a goal. Put in words: *in order to reach the goal, if certain conditions are satisfied, perform this plan of actions*. The equivalent code is something like:

```prolog
+!goal : conditionA & .. & conditionZ  
<- actionA; ..; actionZ.
```

Conditions represent what the agent thinks should be true, in that very moment, in order to be successful in executing his plan. The propositional content is written in a Prolog-like form. In addition, Jason permits the attachment of annotations (a sort of optional predicated parameters, expressed within squared brackets). As for actions, they are either direct operations with the environment, or `!g` (triggering the activation of a goal `g`), `+b, -b` (respectively adding and removing the belief `b` from the belief base).

The second construct is an ECA rule concerning the addition of a belief: *when a certain belief is added, if certain conditions are satisfied, perform this plan of actions.*

```prolog
+belief : conditionA & .. & conditionZ  
<- actionA; ..; actionZ.
```

Although similar, these rules have different semantics. The motivational component `!goal` disappears when the plan ends successfully and also when it fails (in this case, the event `-!goal` is triggered). The knowledge component `belief` is instead maintained.

**Case example: buyer’s payment** As example of synthesis, we translate the interpretation of payment illustrated in Fig. 8.9. This is an excerpt of the code of the buyer agent-role:

```prolog
+!accept(offer(Good, Amount)[source(Seller)])  
<- .send(Seller, tell,  
    accept(offer(Good, Amount)));  
    +obl(pay_to(Amount, Seller)).
+obl(pay_to(Amount, Agent)) <-  
!pay_to(Amount, Agent);  
-obl(pay_to(Amount, Agent)).
+!pay_to(Amount, Agent)  
: owning(Money) & Money >= Amount  
<- .send(w, achieve, pay_to(Amount, Agent));  
    +paid_to(Amount, Agent).
```

For completeness, we included in the first rule also the generation of the speech act of acceptance.\(^{35}\) Neglecting this action, we have three rules, hierarchically dependent,\(^{35}\)

\(^{35}\) `send/3` is an action provided by the MAS platform that generates speech acts. The first parameter is the target agent, the second is the illocutionary force (`tell` for assertions, `achieve` for directives), the third is the propositional content.
and the last one performs a speech act. It is easy to observe a one-to-one correspondence between these elements and the four layers in Fig. 8.9.

8.3 Related research

The Agile methodology for public administrations introduces the concept of agent-role, and targets the exploitation of agent-role descriptions, as components of a knowledge-base corresponding to the deep model Chandrasekaran et al. (1989) of a certain social reality. Such a model can be used to feed design and diagnostic applications with the purpose of supporting the activity of the organization. The legal system in many areas presupposes the use of informal or semi-formalized models of human behaviour in order to operate. If we aim to support an administrative organization on those points, ABM is the most natural choice.

With this objective, we started LN-core, a conceptual framework investigating legal narratives. Our perspective, however, is not the same as legal case ontologies, e.g. Wyner and Hoekstra (2012). We are not concerned with the computational systematization of law, but with the representation of an individual legal “story”. We aim to frame the social behaviour (and the correspondent legal interpretation) that each case contains. If our story model produces the outcomes described in the narrative, then the model is valid.

From a higher-level perspective, the present work connects scenario-based modeling with multi-agent systems technologies. The idea at the base is that, in order to acquire representations of social behaviours, we need cases to be valid models, and we can validate them by their execution. Mueller (2003) already observed that, although several story understanding programs—starting from BORIS, in Charniak (1972)—have used sort of multi-agent systems for their internal representation, this choice is not easy for the programmer: such agents are difficult to write, maintain, and extend, because of the many potential interactions. His experience matched with ours. However, we think that the connection of agent-based modeling with MAS is too strong and important to be easily discarded. We aim to couple on the same simulation framework designed systems (e.g. IT infrastructures) and representations of actual social behaviours.

The modelling exercise runned through the chapter served as an example of application of our methodology. Each type of representation (MSC, topology, Petri net, AgentSpeak(L) code) has his own weakness and strengths. The cross-relations between them are often not simple isomorphisms, but we think of using alternative visualizations as a way to achieve a more efficient elicitation (targeting also non-IT experts). In this line of thought, we plan to implement and assess an integrated development environment in support to the proposed methodology.

Scenario-based modeling MSCs (and collections of them, e.g. HMSCs) were standardized as support for the specification of telecommunication software, in order to capture system requirements and to collect them in meaningful wholes, see for instance Harel and Thiagarajan (2004). Later on, other extensions, like LSCs (Damm and Harel (2001)) and CTPs (Roychoudhury and Thiagarajan (2003)), were introduced to support the automatic creation of executable specifications. Their application of scenario-based modeling consists in collecting multiple inter-object interactions and synthesizing them in intra-object implementations.
In principle, we share part of their approach. Our work promotes the idea of using MSCs, although integrated with intentional concepts. However, in their case, the target is a specific closed system (to be implemented), while in our case, a scenario describes an existing behavioural component of an open social system. At this point, we are satisfied by transforming the MSC of a single case in the correspondent agent-roles descriptions. The superposition of scenarios, with the purpose of associating them into the same agent-role will be investigated in the future.

**Narratives in AI: story understanding**  
AI started investigating stories in the '70s, with the works of Charniak (1972), Abelson and Schank (1977), introducing concepts like *scripts, conceptual dependency* Lytinen (1992), *plot units* Lehnert (1981). The interest towards this subject diminished in the early '80s, leaking into other domains. Mateas and Sengers (1999) and others tried a refocus in the end of '90s, introducing the idea of Narrative Intelligence, but again, the main stream of AI research changed, apart from the works of Mueller Mueller (2004). All these authors, however, are mostly interested in story understanding. We are investigating instead the steps of construction of what they called *script* Abelson and Schank (1977). According to our perspective, common-sense is not constructed once, in a script-like knowledge, but emerges as a repeated pattern from several representations. Furthermore, we explicitly take account of the integration of fault and non-compliant behaviours, increasing the “depth of field” of the representation.

During the last few years, because of the advent of digital and computational humanities, studies concerning narratives have been increasing. The drift from narrative intelligence to computational models of narratives may be associated to that from *agent-based modeling* to *computational social science* observed in Conte and Faolucci (2011). Against this trend, this paper is deeply focused towards agent-based modeling.

**Computational implementation**  
Reproducing a system of interacting subsytems needs concurrency. Models of concurrent computation, like the Actor model Hewitt et al. (1973), are implemented today in many development platforms. In our story-world, this solution would be perfect for objects. We would need instead to add intentional and institutional elements in order to implement agent-roles. The connection with another programming paradigm (intended to handle concurrency) will plausibly solve most of the problems of scalability that usually haunt MAS platforms, often developed by a logic-oriented community.\(^{36}\)

\(^{36}\) Preliminary experiments of translating Jason on a functional programming paradigm, like for instance in Erlang Diaz et al. (2012), have proven to be computationally very efficient.
Chapter 9

Designing and animating implementations of the law

This section will briefly resume some of the practical results achieved until now. There are two main sections. In the first, few examples will present outcomes of the MAS modeling, and show how they can be used for animation. In the second, we will use such agent-role based modelling and execution steps to operationalize a model-based diagnosis on a famous logic puzzle.

9.1 Animation toolkit

As we have repeated in many points of this document, we have targeted for our development the MAS platform Jason. It is based on AgentSpeak(L), an elegant language for programming cognitive agents. Despite of being easy to be installed and used, it comes unfurnished of basic user interfaces for agents. Furthermore, we need to implement basic functions handling institutional thinking and other modules (maintenance goals, epistemic identities). We are consequently developing two extensions to the platform:

- \textit{jEthos}, providing an institutional stance to the agents, and some relevant extensions to transform the Jason MAS platform in agent-based modelling platform, specifically intended for legal scenarios,
- \textit{jAnimas}, agent user interfaces builder: jAnimas stands for java Animator for MAS; it is a sort of middleware intended to support the creation of simple user interface for agents.

The current implementation is still far to be complete and suitable for practical use in production, but our demos works correctly and are downloadable from our site. They serve mainly as a way to test in practice our theoretical contributions.
9.2 Hunting foxes

As example of legal narrative, we will consider a well known case in Property Law and in AI & Law: Pierson v Post (1805). The case, in the form of a sequence of facts already established in a previous decision, is reported to the judges. They express two different positions, embodied by Tompkins (majority) and Livingston (dissent). Reduced to a minimum, the story is the following:

Post was hunting a fox with a horse and hounds in a wild and uninhabited land, and was about to catch it, but Pierson, although conscious of Post’s pursuit, intercepted the fox, killed it and took the animal.

Both claimed the fox, the first appeal found for Post, but this court reverts the previous result. The different positions are expressed by two judges: Tompkins (majority) and Livingston (dissent). We summarize their discourse below. Tompkins, supported by classical jurisprudence, states that:

Possession of a fera naturae occurs only if: 1) there is occupancy, i.e. taking physical possession, 2) it has been mortally wounded and it is still pursued, 3) it has been secured with nets and trolls. Pierson took the animal, so he owns it.

where fera naturae is an animal wild by nature. Livingston’s main line is:

If someone starts and hunts a fox with hounds in a vast and uninhabited land has a right of taking the fox on any other person who saw he was pursuing it.

Further considerations may be drawn analyzing Livingston’s discourse in detail. He starts from an informal social rule, a customary law followed by hunters. He accepts (with some reluctance) the references to classical jurisprudence. However, he observes that the animal involved in this case is a fox, which he qualifies as noxious and hostis humani generis, so, its killing becomes of public interest. Legal institutions should thus promote it, advantaging who is investing resources (like dogs, hounds, time) to reach this objective. This meta-institutional reasoning is not in our scope, for now. We take both interpretations as relevant for our purposes. If the dissent opinion had been majority, then we would have had actually another (hi)story.

9.2.1 Implementation

We present hereby excerpts of the code of the agents.

Post

The methodology we followed to create this code has been explained with some details in section [XXX].

!start(own(dogs)).
!start(own(horse)).
!start(see(fox)).

!hunt(fox)[with(hounds)].

+!hunt(Fox)[with(hounds)] : ongoing(see(Fox)) <=
   !start(hunt(Fox)[with(hounds)]);

1Source text: http://www.facstaff.bucknell.edu/kinnaman/Pierson v.htm.
!catch(Fox).
+!catch(Fox) : near(Fox) <-
   !success(catch(Fox)).
+!catch(Fox) : not near(Fox) <-
   !near(Fox).
+!near(Fox) <-
   !approach(Fox).
+!approach(Fox)
   <- !start(approach(Fox)).
+taken(Person, Fox) : ongoing(hunt(Fox))
   <- !fail(hunt(Fox)); !stop(approach(Fox)).

near(Fox) :- success(approach(Fox)).

+success(asi(Person, take(Fox))) <- +taken(Person, Fox).

Common libraries  The asi functor introduced in the last line is an acronym for action, state and intent. It is a way to reify in a reusable form the different relations of an agent with these concepts.

Event transitions  The first place where we use such construct is in handling the transition of events conducted/observed by the agents. We report an excerpt of code shared by all acting agents (slightly different from that of interpreting agents).

+!start(Asi) : .my_name(Ag) <-
   +ongoing(Asi); .print("is starting ", Asi);
   .broadcast(tell, start(Asi)).

+!stop(Asi) : .my_name(Ag) <-
   .abolish(ongoing(Asi)); .print("is stopping ", Asi);
   .broadcast(tell, stop(Asi)).

+!success(Asi) : .my_name(Ag) <-
   +success(Asi); .abolish(ongoing(Asi));
   .print("brought about ", Asi);
   .broadcast(tell, success(Asi)).

+!fail(Asi) : .my_name(Ag) <-
   +fail(asi(Ag, Asi)); .abolish(ongoing(Asi));
   .print("failed in bringing about ", Asi);
   .broadcast(tell, stop(Asi)).

+start(Asi)[source(Ag)] : not (Ag = self) <-
   +ongoing(asi(Ag, Asi)); -start(Asi)[source(Ag)].

+stop(Asi)[source(Ag)] : not (Ag = self) <-
   -ongoing(asi(Ag, Asi)); -stop(Asi)[source(Ag)].

+success(Asi)[source(Ag)] : not (Ag = self) <-
Institutional logic  The asi construct has been introduced mainly for the institutional logic, used by interpreting agents (e.g. judges). In our implementation we translated some of the fundamental legal concepts of obligations and permissions as defined in Sartor (2006b) in a form that could be used by agents in Jason. We are referring namely to these deontic rules (possibly extended with a teleological attribute):

- \( \text{Obl}(A) \Leftrightarrow \text{Forb}(\neg A) \)
- \( \text{Forb}(A) \Leftrightarrow \text{Perm}(A) \)
- \( \text{Facult}(A) \Leftrightarrow (\text{Perm}(A) \land \text{Perm}(\neg A)) \)
- \( \text{Obl}(A) \Rightarrow \text{Perm}(A) \)

The parameter \( A \) may be characterized in three ways, explaining the asi acronym:

- Behavioural characterization/Action: \( A \leq \text{does(Person, Action)} \),
- Qualification, attribution and state characterization/State: \( A \leq \text{is(Person, Qualification)} \),
- Productive characterization/Intent, Outcome: \( A \leq \text{brings(Person, Result)} \).

To give an idea of our implementation, we report the code for the deontic operator for obligation, to be read as ”it is obligatory that.” Given any (positive or negative) asi description \( \Lambda \), we have:

```plaintext
// obl(\Lambda) \iff forb(\neg \Lambda)
// \neg obl(\Lambda) \iff \neg forb(\neg \Lambda)
// obl(\Lambda) \Rightarrow perm(\Lambda)
// \neg perm(\Lambda) \Rightarrow \neg obl(\Lambda)
```

```
+obl(asi(Person, Asi)) <- ~obl(asi(Person, Asi));
+perm(asi(Person, Asi)); +forb(~asi(Person, Asi)).
+obl(~asi(Person, Asi)) <- ~obl(~asi(Person, Asi));
+perm(~asi(Person, Asi)); +forb(asi(Person, Asi)).
+-obl(asi(Person, Asi)) <- ~obl(asi(Person, Asi));
+-obl(~asi(Person, Asi)) <- ~obl(~asi(Person, Asi));

-pobl(asi(Person, Asi)) <- -forb(~asi(Person, Asi)); -perm(asi(Person, Asi)).
-pobl(~asi(Person, Asi)) <- -forb(asi(Person, Asi)); -perm(~asi(Person, Asi)).
~obl(asi(Person, Asi)) <- ~forb(asi(Person, Asi));
~obl(~asi(Person, Asi)) <- ~forb(~asi(Person, Asi)).
```

Pierson  The code of this agent is developed similarly to the previous one.

```
!start(see(fox)).
!hunt(fox).

+!hunt(Fox) : ongoing(see(Fox)) <-
  .wait(1000); // Pierson starts later
```
Tompkins: majority opinion  This agent is an interpreter: he observes the actions of other agents and interprets according to certain rules. The integrated interpreting rules however allow just one interpretation.

// Possession of a fera naturae occurs only if there is occupancy, // i.e. taking physical possession

 /////////////////////////////////////////////////////////////////////////////
 // constitutive rules
 // i.e. institutional rules, triggered by an event
 /////////////////////////////////////////////////////////////////////////////

 /* HUNTING */

 feranaturae(Object) :- fox(Object).

 +!success(Ag, Asi) :
   perm(asi(Ag, Asi)) &
   Asi =.. [Action, [Object], _] &
   feranaturae(Object) & // fera naturae &
   Action = take // corporal possession
   <- +owns(Ag, Object); // -> property
   +start(aso(Ag, own(Object)));


.print("fera naturae & corporal possession -> property
  => ", Ag, " owns ", Object, ".").

// if nobody owns the object, than anyone may take it
+!asi(Ag, Asi):
  Asi =.. [take, [Object], _] &
  not owns(_, Object)
  <- +perm(asi(Ag, Asi)); +perm(~asi(Ag, Asi)).

+!~asi(Ag, Asi):
  Asi =.. [take, [Object], _] &
  not owns(_, Object)
  <- +perm(asi(Ag, Asi)); +perm(~asi(Ag, Asi)).

/* PROPERTY */

// exclusivity of ownership~
owns(Ag, Object) :- owns(X, Object) & not (X = Ag).

// if someone starts owning an object
// the previous owner loses the property
+!start(Ag, asi(own(Object)))
  : owns(X, Object)
  & not (X = Ag)
  <- +owns(Ag, Object);
  -owns(X, Object).

/////////////////////////////////////////////////////////////////////
// institutional rules
// not triggered by an event
/////////////////////////////////////////////////////////////////////

// if the person owns the object,
// then he is permitted doing or not doing whatever he wants
+!asi(Ag, Asi):
  Asi =.. [_., [Object], _] &
  owns(Ag, Object)
  <- +perm(asi(Ag, Asi)); +perm(~asi(Ag, Asi)).

+!~asi(Ag, Asi):
  Asi =.. [_., [Object], _] &
  owns(Ag, Object)
  <- +perm(~asi(Ag, Asi)); +perm(asi(Ag, Asi)).

// if the person does not own the object,
// then he is not allowed to do anything
// but he is allowed to stop doing it
+!asi(Ag, Asi):
  Asi =.. [Action, [Object], _] &
  ~owns(Ag, Object)

213
/* HUNTING */
feranaturae(Object) :- fox(Object).

%!start(Ag, Asi) :
  Asi =.. [Action, [Object], _] & // (start) hunting
  fox(Object) & // fox
  in_array(with(hounds), Annot) & // with hounds
  not ~perm(asi(Ag, take(Object))) // there is not another having the right
  <- +perm(asi(Ag, take(Object))); // -> right of taking the fox on anybody else
  .print("hunting fox with hounds \rightarrow right of taking it
  \rightarrow ", Ag, " has exclusive right of taking ", Object, ").

%!success(Ag, Asi) :
  perm(asi(Ag, Asi)) &
  Asi =.. [take, [Object], _] & // corporal possession +
  feranaturae(Object) // fera naturae \rightarrow
  <- +owns(Ag, Object); // property
  .print("corporal possession & fera naturae \rightarrow property
  \rightarrow ", Ag, " owns ", Object, ").

// if nobody owns the object, than anyone may hunt and take it
%!asi(Ag, Asi) :
  Asi =.. [Pred, [Object], _] &
  (Pred = hunt |
  Pred = take & not ~perm(asi(Ag, take(Object)))) &
  not owns(_, Object)
  <- +perm(asi(Ag, Asi)); +perm(~asi(Ag, Asi)).

%!~asi(Ag, Asi) :

Livingston: dissent opinion  This agent is an interpreter too, but he has different interpreting rules then the previous one.

// If someone starts and hunts a fox with hounds in a vast and uninhabited
// land has a right of taking the fox on any other person who saw he was
// pursuing it.

/////////////////////////////////////////////////
// constitutive rules
// i.e. institutional rules, triggered by an event
/////////////////////////////////////////////////

/* HUNTING */
feranaturae(Object) :- fox(Object).

%!start(Ag, Asi) :
  Asi =.. [Action, [Object], _] & // (start) hunting
  fox(Object) & // fox
  in_array(with(hounds), Annot) & // with hounds
  not ~perm(asi(Ag, take(Object))) // there is not another having the right
  <- +perm(asi(Ag, take(Object))); // -> right of taking the fox on anybody else
  .print("hunting fox with hounds \rightarrow right of taking it
  \rightarrow ", Ag, " has exclusive right of taking ", Object, ").

%!success(Ag, Asi) :
  perm(asi(Ag, Asi)) &
  Asi =.. [take, [Object], _] & // corporal possession +
  feranaturae(Object) // fera naturae \rightarrow
  <- +owns(Ag, Object); // property
  .print("corporal possession & fera naturae \rightarrow property
  \rightarrow ", Ag, " owns ", Object, ").

// if nobody owns the object, than anyone may hunt and take it
%!asi(Ag, Asi) :
  Asi =.. [Pred, [Object], _] &
  (Pred = hunt |
  Pred = take & not ~perm(asi(Ag, take(Object)))) &
  not owns(_, Object)
  <- +perm(asi(Ag, Asi)); +perm(~asi(Ag, Asi)).

%!~asi(Ag, Asi) :

214
Asi =.. [take, [Object], _] &
(Pred = hunt |
Pred = take & not ∼perm asi(Ag, take(Object))) &
not owns(_, Object)
<- +perm asi(Ag, Asi); +perm (∼asi(Ag, Asi)).

/* PROPERTY */

// exclusivity of the right of taking (without reification)
perm asi(Ag, take(Object)) :- perm asi(X, take(Object))) & not (X = Ag).

// exclusivity of ownership (without reification)
owns Ag, Object) :- owns(X, Object) & not (X = Ag).

// if someone starts owning an object
// the previous owner loses the property
+!start(Ag, asi(own(Object)))
  : owns(X, Object)
  & not (X = Ag)
  <- +owns(Ag, Object);
  ∼owns(X, Object).

// institutional rules
// not triggered by an event

// if the person owns the object,
// then he is permitted doing or not doing whatever he wants
+!asi(Ag, Asi) :
  Asi =.. [_, [Object], _] &
  owns(Ag, Object)
  <- +perm asi(Ag, Asi); +perm (∼asi(Ag, Asi)).

+!∼asi(Ag, Asi) :
  Asi =.. [_, [Object], _] &
  owns(Ag, Object)
  <- +perm (∼asi(Ag, Asi)); +perm asi(Ag, Asi)).

// if the person does not own the object,
// then he is not allowed to do anything
// but he is allowed to stop doing it
+!asi(Ag, Asi) :
  Asi =.. [Action, [Object], _] &
  ∼owns(Ag, Object)
  <- +∼perm asi(Ag, Asi); +perm (∼asi(Ag, Asi)).

+!∼asi(Ag, Asi) :
  Asi =.. [Action, [Object], _] &
  ∼owns(Ag, Object)
9.2.2 Outcome

The demo gives the following result on the console when executed:

[post] is starting own(dogs)
[pierson] is starting see(fox)
[post] is starting own(horse)
[post] is starting see(fox)
[post] is starting hunt(fox)[with(hounds)]
[post] is starting approach(fox)
[livingston] hunting fox with hounds $\rightarrow$ right of taking it
$\Rightarrow$ post has exclusive right of taking fox.

[pierson] is starting hunt(fox)
[pierson] brought about approach(fox)
[pierson] brought about intercept(fox)
[pierson] brought about kill(fox)
[pierson] brought about take(fox)
[pierson] is stopping hunt(fox)
[post] failed in bringing about hunt(fox)
[livingston] FAULT: pierson is not permitted to take(fox)
[tompkins] fera naturae & corporal possession $\rightarrow$ property
$\Rightarrow$ pierson owns fox.
[pierson] brought about hunt(fox)
[post] is stopping approach(fox)

9.3 A real estate fraud

As complete example for the animation framework, we consider a well known type of real-estate fraud, which is a specific patterns of swap scheme. Basically, a swap scheme is a coordinated transaction operating on different institutional layers. An example of swap scheme is the vote-buying case: a candidate, in exchange of the vote of an elector, promises him the performance of a certain action. At a first look, this seems not different from any political campaign, directed often towards specific communities. These social aggregates use their social influence on their members to support the politician in exchange of something in return. This is not forbidden. What changes in a vote-buying scheme is that the coordination is strict: there is a control on the individual vote performance.

Considering the domain of our example, we are concerned about the sale institution. In this frame, a swap scheme may occur between dual coupling of buyer and seller agents, as in Fig. 9.1. The coordinator role may be played by a physical person, a group of people, an organization, etc. It identifies something that determines an intentional connection between the buyer and the seller. When the exchange occurs, we acknowledge a higher-level coordination, reified by some kind of agreement (e.g. a contract). On a small scale this is not forbidden: e.g. “if you make me pay less the guitar that your father is selling, I would make you buy my brother’s motorcycle for less.” However, in real-estate transactions, when we include in the picture the tax
administration, applying the transfer tax, a swap scheme can be a way to reduce the amount of due taxes, as shown in Fig. 9.2.

The practical discrimination used by investigators to discover potential tax-fraud is the comparison with the average market price.\(^2\)

### 9.3.1 Implementation

Our scenario is quite simple: we have buyers and sellers, buying and selling houses of type \(a\) and of type \(b\), following a normal script, defined with a certain economic rationality. They make public offers so the (public) average market price can be calculated. A bank and a store actors handle the finalization of the transaction.

We have then coordinators, which are entities able to coordinate a buyer and a seller targeting opposite types of real-estate. Coordinators do not know each other in advance, so they can ask for a swap scheme to offerors dual to their intents. E.g., a coordinator knows that a certain seller offers the house he is looking for. He asks to the

\(^2\)A way to overcome the problem of assessing the market price of a certain real-estate property is to check the value of the same real-estate in previous sale transactions. In case of a swap scheme, the new owners tend to sell the just acquired property after a relatively small time, but for a much higher price.
seller who is his coordinator. If he receives an answer, he proposes him a swap-schem.
If this coordinator is interested in his offer, and accept to start a swap-scheme, a new
coordination is put in place, performing a real-estate fraud.

The tax administration does not have any control over these communications. However, it does receive all sale declarations. Then, for each sale declaration, it compares the paid price against the market value. If the value is strangely low, it starts an investigation. The investigation starts looking for coordinators behind the buyers and sellers involved in such suspicious cases. If the investigation find coordinators dually involved (one’s buyer with other’s seller, and vice-versa), then the tax administration attributes to the coordinations the performance of a swap-scheme.

Code excerpt of buyer

The buyer/seller script are compliant to the sale transaction: they always pay and deliver. They can have proactive and/or passive behaviours, i.e. play the roles of offeror and/or offeree. The following is a simplified excerpt from the code of the buyer. Despite of being pruned, the code is still quite long: it is handling already many of the possible problem of synchronization involving the reception of multiple acceptances, offers, etc.

```plaintext
//---------------------------------------------
// when I want to buy a certain good,
// I look in my beliefs for the price I want
//---------------------------------------------
+!buy(Good)
  : .my_name(Ab) & price(Good, Sum) & money(Money) &
  not acceptance(_, sale(Ab, _, Good, _), _) & (Money >= Sum) & proactive
  <- m("I want to buy ", Good, ", and my price is ", Sum,"."");
  // I could ask for the good, proposing already my price
  !demand(Good, Sum);
  .wait(1000); .random(R); .wait(R*2000);
  !buy(Good).
+!buy(Good)
  : .my_name(Ab) & price(Good, Sum) & money(Money)
  & not acceptance(_, sale(Ab, _, Good, _), _) & (Money >= Sum) & not proactive
  <- m("I want to buy ", Good, ", and my price is ", Sum,". Waiting for an offer...").
+!buy(Good)
  : .my_name(Ab) & price(Good, Sum) & money(Money)
  & not acceptance(_, sale(Ab, _, Good, _), _) & (Money < Sum)
  <- m("I want to buy ", Good, ", but I don’t have enough money (", Money, ")");
  !decrease_my_price(Good);
  .wait(1000); .random(R); .wait(R*2000);
  !buy(Good).
+!buy(Good)
  : .my_name(Ab) & price(Good, Sum) & acceptance(_, sale(Ab, _, Good, _), _)
  <- m("I want to buy more, but I’m already doing a deal now. I’ll try again later.");
```

218
.wait(1000); .random(R); .wait(R*2000); // wait to finish the deal before restarting
!buy(Good).

// first thing to do: check if I have money to buy with
+!buy(Good)
  : not money(Money)  
  <- m("I don't know if I have money..");  
  !lookbank;  
  .wait(1000); .random(R); .wait(R*1000);  
  !buy(Good).

--------------------------
// passive behaviour
//

// receive an offer from the seller
+supply(Good, Sum)[source(As)]
  <- !ack(As, supply(Good, Sum)).

// acceptable offer
+!ack(As, supply(Good, Sum))
  : .my_name(Ab) & acceptable(Good, Sum)
  & not acceptance(sale(Ab, _, Good, _), _) & money(Money) & not (Sum > Money)
  <- m(As, " offers ", Good, " for ", Sum, ", that's good for me.");  
  !decrease_my_price(Good); // update my research price  
  !supply_acceptance(Good, Sum, As);  
  -supply(Good, Sum)[source(As)]. //forget the offer just received

// not acceptable offer. review internal price
+!ack(As, supply(Good, Sum))
  : .my_name(Ab) & not acceptable(Good, Sum)
  & not acceptance(sale(Ab, _, Good, _), _) & money(Money) & not (Sum > Money)
  <- m(As, " offers ", Good, " for ", Sum, ", no use.");  
  !increase_my_price(Good); // update my research price  
  -supply(Good, Sum)[source(As)]. //forget the offer just received

// not enough money
+!ack(As, supply(Good, Sum))
  : .my_name(Ab) & money(Money) & (Money < Sum)
  <- m("I don't have enough money to accept..");  
  -supply(Good, Sum)[source(As)]. //forget the offer just received

// do not know if he has the money
+!ack(As, supply(Good, Sum))
  : not money(_)
  <- m("I don't remember how much I have..");  
  !lookbank;  
  .wait(1000); .random(R); .wait(R*1000); // wait for an answer  
  !ack(As, supply(Good, Sum)).
// buyer says to seller he's accepting his offer
+!supply_acceptance(Good, Sum, As)
  : .my_name(Ab) & not acceptance(Ab, sale(Ab, _, Good, _), _)
  & not acceptance(sale(Ab, _, Good, _), _) & money(Money) & (Money >= Sum) & time(Now)
  <- m("I'm telling ", As, " I'm accepting his price (" Sum, ")");
  !first_acceptance(Ab, sale(Ab, As, Good, Sum)).

+!supply_acceptance(Good, Sum, As)
  : .my_name(Ab) & not acceptance(Ab, sale(Ab, _, Good, _), _)
  & not acceptance(sale(Ab, _, Good, _), _) & money(Money) & (Money < Sum)
  <- m("I don't have enough money (" Money, " vs ", Sum, ").").

//--------------------
// proactive behaviour
//
// buyer tells seller he's looking for a good at a certain price
+!demand(Good, Sum, As)
  <- m("I'm telling ", As, " I'm looking for ", Good, " costing ", Sum, ");
  !private_purchase_offer(Good, Sum, As). // PRIVATE OFFER

// buyer tells everyone he's looking for a good at a certain price
+!demand(Good, Sum)
  <- m("I'm telling everyone I'm looking for ", Good, " costing ", Sum, ");
  !public_purchase_offer(Good, Sum). // PUBLIC OFFER

+!ack(As, demand_accepted(Good, Sum))
  : .my_name(Ab)
  <- !select_acceptance(sale(Ab, As, Good, Sum));
  !execute_acceptance(sale(Ab, As, Good, Sum)).

// buyer receives the acceptance from seller, but he has accepted it too!
+!select_acceptance(sale(Ab, As, Good, Sum))
  : .my_name(Myself) & Myself = Ab & acceptance(Ab, sale(Ab, As, Good, _))
  & acceptance(As, sale(Ab, As, Good, _))
  <- m("Eheh, I've already accepted your price,", As, " (", Sum, ")!! I'll drop your acceptance.");
  -acceptance(As, sale(Ab, As, Good, _));
  es(acceptance(As|Ab)).

// buyer receives the acceptance from seller
// in case buyer receives more orders, just the first one pass.
+!select_acceptance(sale(Ab, As, Good, Sum))
  : .my_name(Myself) & Myself = Ab & not acceptance(Ab, sale(Ab, _, Good, _))
  & not acceptance(sale(Ab, _, Good, _), _)
  <- m(As, " accepted my price (" Sum, ");
     !second_acceptance(Ab, sale(Ab, As, Good, Sum)).

// buyer has already a deal in act
// the other orders are not accepted
// select acceptance
+!select_acceptance(sale(Ab, As, Good, Sum))
  : .my_name(Myself) & Myself = Ab & (acceptance(sale(Ab, sale(Ab, _, Good, _)), _)
  | acceptance(sale(Ab, _, Good, _), _))
  <- m("I've already started a deal to buy some ", Good,
    ", my offer is not valid any more.");
  -acceptance(As, sale(Ab, As, Good, Sum), _) [source(_)];
  es(acceptance(As|Ab)).

// if the acceptance is valid then buyer executes the deal
+!execute_acceptance(sale(Ab, As, Good, Sum))
  : .my_name(Myself) & Myself = Ab & acceptance(sale(Ab, As, Good, _), _) & time(Now)
  <- !decrease_my_price(Good); // update my research price
    m("I start the deal..");
    !start_deal(sale(Ab, As, Good, Sum));
    !execution(sale(Ab, As, Good, Sum)).

// if there is not acceptance he makes the other drop
+!execute_acceptance(sale(Ab, As, Good, Sum))
  : .my_name(Myself) & Myself = Ab & not acceptance(sale(Ab, As, Good, _), _)
  <- m("I tell ", As, " that the offer was not valid any more.");
    !drop_acceptance(As, demand_accepted(Good, Sum)).

/////////////////////////////
// buyer common functions
/////////////////////////////

// wait for legal confirmation of the deal
+!buyer_execution(sale(Ab, As, Good, Sum))
  : .my_name(Myself) & Myself = Ab & deal(sale(Ab, As, Good, Sum))
  <- !pay(Sum, As);
    !confirm_delivery(Good, As);
    !close_deal(sale(Ab, As, Good, Sum)).

// make payment
+!pay(Sum, As)
  <- m("Pay ", As, " with ", Sum, ".");
    !transfer(Sum, As).

// receive delivery
+!confirm_delivery(Good, As)
  : .my_name(As) & delivered(Good, As)
  <- m("Delivery from ", As, " OK.").

+!confirm_delivery(Good, As)
  : .my_name(As) & not delivered(Good, As)
  <- m("No delivery yet from ", As, ".");
    .wait(1000); .random(R); .wait(R*1000);
    !confirm_delivery(Good, As).

+delivered(Good, As)
9.3.2 Code excerpt of the tax administration

We have selected two parts of the code of the "tax administration". The first is the part checking for suspicious transactions. We define a risk rate of 20%. If the paid price compared to market price is lower than this rate, then investigation starts.

```prolog
riskrate(20).

acceptable_price(Good, Price) :-
  riskrate(RiskRate) & starting_market_price(Good, MarketPrice)
  & (Price > ((MarketPrice * (100 - RiskRate))/100)).
```

```prolog
%!check_declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now)
  :- m("This declarations seems plausible: ",
        declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now)).
```

```prolog
%!check_declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now)
  :- not acceptable_price(Good, Price) & riskrate(RiskRate)
  & starting_market_price(Good, MarketPrice)
  <- m("This declaration is strange ",
         declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now));
         m("The price does not follow the market. Further investigation is required.");
         !suspect(A); !suspect(B); +suspect_transaction(sale(A, B, Good, Price), Time).
         .wait(500).
```

```prolog
%!check_declarations
  :- declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now)
  & not checked(declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now))
  <- !check_declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now);
     +checked(declaration(Agent, transaction(sale(A, B, Good, Price), Time), Now));
     .wait(500);
     !check_declarations.
```

The second part is the actual investigation, looking for dual combinations of coordinators:

```prolog
%!investigate(transaction(sale(A, B, Good, Price), Time))
  :- not coordinator(_, A)
  <- m("Looking for the coordinator behind ", A, ", "...);
     !get_coordinator(A); .wait(300);
     !investigate(transaction(sale(A, B, Good, Price), Time)).
```

```prolog
%!investigate(transaction(sale(A, B, Good, Price), Time))
  :- not coordinator(_, B)
  <- m("Looking for the coordinator behind ", B, ", "...);
     !get_coordinator(B); .wait(300);
```
9.3.3 Results

Integrating the user interface provided by jAnimas we allow the following actions: for the buyers/sellers, buy/sell for just one time, or as a maintenance goal. Fig. 9.3 reports a screenshot taken from an execution: broadcast messages are on the left, buyer and seller interfaces on the right. The top of an agent UI contains the buttons with commands available to the user. The bottom shows the states (ex. persistent intentions) the agent currently has.

For the coordinators, the agent UI gives to the user the possibility of make them trying to perform a swap-scheme. If we create in that market dual coordinators, and make them available to a swap scheme, they interact and starts an adequate coordination. The tax administration however in monitoring all transactions. As soon as it receives strange declarations, it starts investigating suspicious deals (Fig. 9.4).

When the tax administrator has enough information, i.e. finds matching dual coordinators, it can attribute to the involved agents the performance of a swap-scheme (Fig. 9.5).
9.4 Knights and knaves

In this section we show how we could use the MAS platform in a model-based diagnosis application. We use as sandbox a variation of the famous puzzles by Smullyan et al. (2011):

You are on a island of knights and knaves. In front of you there are three people: A, B and C. B can talk with A, and you can talk with B and C. You ask B to ask A what he is, and B responds “A says he is a knave”. You ask C what B is, and C responds “B is a knave”. Who are the knaves?

The puzzle can be solved with logical tools and it has been widely investigated in epistemic logic. From another perspective, however, it approaches the wider problem of performing and evaluating interpretations of social (norm-guided) behaviours.

Target knowledge and problem solving method

Suppose you just arrived on the island. In order to start, or adequately react to, any social transaction, you need to: (a) acquire the social rules in force; (b) be able to apply them correctly, hopefully in support to your intents. For these purposes, you may: (i) elicit behavioural patterns from direct or indirect sources; (ii) act and learn

---

3Knights always tell the truth, while knaves always lie. People of both classes will tell, if asked, whether someone (including themselves) are knights or knaves; they can also forward questions to other persons, and return the answers.
Figure 9.4: *jAnimas* interface: highlighting suspicious deals.
Figure 9.5: *jAnimas* interface: highlighting fraud coordinators and performers.
from outcomes of actions; (iii) use known patterns to interpret others’ behaviour (ex. identifying liars)\(^4\).

The first and second actions stand as forms of model elicitation and verification. Guidance (ex. law) is usually expressed in form of rules, associated to roles, which in turn are enacted by classes of agents (ex. knight and knaves). In a real case, however, humans may play many interconnected roles (not formally described) concurrently. When someone discovers a successful scheme of behaviour, after guidance, via mimesis or by chance, and possibly goes through subsequent repeated reinforcements, he acknowledges goal-oriented plans of actions, dual to specific social dispositions. A possible representation model for this knowledge is given by the agent-role\(^5\) Boer and van Engers (2011d).

The third action requires an adequate knowledge base, consisting in a set of agent-roles, which describes patterns of behaviour that are prototypical in the target social domain. Some of them represent normal behaviours (ex. knight), while others are associated to faulty, non compliant ones (ex. knaves), in the sense of being at fault with the (normatively characterized) behaviour of the social system\(^6\). Thus, the proposed puzzle can be seen as a diagnostic problem, that can be resolved with a model-based diagnosis.

**Ambient intelligence and reasonableness**

What happens if a knight does not know what another agent is? The puzzle gives no clues on how agents process the unknown, nor how beliefs about other agents are created. Agents which perform ambient interpretation\(^7\) may be classified in skeptical agents, which believe something only when it is deductively true, and confident (to various extent) agents, which generate best possible explanations, according to subjective beliefs and heuristics. In a social system with agents from both classes, skeptical agents are obliged to face the change of interpretations of confident agents, and then to implement some form of belief revision. From another point of view, this implies that the agent rationality may produce wrong (in an ex-post perspective) outcomes too, because of faulty inputs, incomplete knowledge or wrong processing.

### 9.4.1 Operationalization

In this section, we outline the application of the model-based diagnosis on Smullyan’s puzzle. First, we create executable models of the subsumed agent-roles. Second, we generate all explanatory hypotheses, allocating known agent-roles to scenario agents. Third, we evaluate all explanations, given the messages reported in the puzzle.

---

\(^4\)This can be done from a neutral perspective (ex. anyone may be knight as much as knave), or considering certain prior assumptions (ex. apart few people, they are all knights).

\(^5\)For support in psychological studies, see Fonagy and Target (1997). In addition, correlating the concept of agent-role to that of script, you may refer also to Abelson (1981).

\(^6\)It is worth to observe that compliance and non-compliance are qualifications relative to the position of the diagnostic agent in the social system. In a world of knaves, knights would fail in respect to the social practice of systematically lying.

\(^7\)Diagnostic agents are part of this class, but they require to be activated, e.g. by some alarming message.
Implementing agent-roles as agent programs  The puzzle presents some
behavioural rules, which entail logical and causal dependencies. As development
solution, we choose to translate the first as Prolog-like rules, the second as event
condition action (ECA) rules. This is possible for instance using AgentSpeak(L) Rao (1996), a
logic programming language for cognitive agents, implemented on the platform Jason
Bordini et al. (2007). We report as example excerpts from the scripts of a knight. A
knight replies to any question about the group he or another person belongs to:

\[+tell(k(X))[source(M)] : k(X) <- .send(M, tell, k(X)).\]
\[+tell(k(X))[source(M)] : ~k(X) <- .send(M, tell, ~k(X)).\]

Going further, we model the three rules that concern the question forwarding,
the internal transformation of messages (perceptions) to observations (beliefs), the
forwarding of answers:

\[+ask(k(X), K)[source(M)] <- +asking(M, ask(k(X), K)); .send(K, achieve, tell(k(X))).\]
\[+k(X)[source(K)] <- +says(K, k(X)); ~k(X)[source(K)].\]
\[+says(K, k(X))[source(self)] : asking(M, ask(k(X), K))
<- .send(M, tell, says(K, k(X))); ~asking(M, ask(k(X), K)).\]

A logical dependency entailed by the puzzle could be: if \(K\) is a knight, and he says
that \(X\) is a knight too, then \(X\) is a knight:

\[k(X) :- said(K, k(X)) & k(K).\]

Note that this expression is based on previous assignments of agent-roles.

Obviously, the proposed solution is just one of all possible implementations. If the
model reproduces the expected messages (in content and in order), then it is valid.
The complete code excerpt for the knight is then:

\[+tell(k(X))[source(M)] : .my_name(X) <-
.print("Agent ", M, " is asking me if I am knight or knave.");
.send(M, tell, k(X)).\]
\[+tell(k(X))[source(M)] : k(X) <-
.print("Agent ", M, " is asking me about ", M, ". (I know he is a knight.");
.send(M, tell, k(X)).\]
\[+tell(k(X))[source(M)] : ~k(X) <-

\(^8\)An adequate epistemic analysis is beyond the scope of this paper. We assume as pragmatic
discrimination the presence/absence of temporal factor in the associated if .. then statement.

\(^9\)The first rule can be read as: when [event] you receive a request, from a monitor agent
\(M\), (of achieving the goal) to tell what you know about the proposition \(k(X)\), if [condition]
(you believe that) \(k(X)\) is true, then [action] perform the correspondent assertive speech act,
directed to \(M\). .send/3 is an action provided by the MAS platform that generates speech acts.
The first parameter is the target agent, the second is the illocutionary force (tell for assertions, achieve for requests). Note the strong negation operator in the second rule.

\(^{10}\)As an additional example of the language semantics, the last rule in the code can be read as:
when [event] (you start believing that) agent \(K\) has said that \(k(X)\) is true, if [condition]
(you believe that) \(M\) has asked you to interrogate \(K\) about \(k(X)\), then [action] report to \(M\) the
answer just received and forget \(M\)'s request.
knight / knave

IN
OUT
monitor
OUT
IN
IN OUT
!tell(k(X))
K
k(X), ~k(X)
!ask(k(X), K)!tell(k(X))
says(K, k(X))
knight / knave
knight / knave
k(X), ~k(X)

Figure 9.6: Messages exchanged between knight/knave and monitor agent-roles

```prolog
+!tell(k(X))[source(M)] : not k(X) <-
  .print("Agent ", M, ", " is asking me about ", M, ", . I know he is a knave.").
  .send(M, tell, ~k(X)).

+!ask(k(X), X)[source(M)] <-
  .print("Agent ", M, ", " is asking me to ask ", X, ", about himself.");
  +asking(M, ask(k(X), X)); .send(X, achieve, tell(k(X))).

+!ask(k(X), K)[source(M)] <-
  .print("Agent ", M, ", " is asking me to ask ", X, ", about ", K, ", ".");
  +asking(M, ask(k(X), K)); .send(X, achieve, tell(k(k))).

+k(X)[source(K)]
  <- +said(K, k(X)); -k(X)[source(K)].
+~k(X)[source(K)]
  <- +said(K, ~k(X)); -~k(X)[source(K)].

+said(K, k(X))[source(self)] : asking(M, ask(k(X), K))
  <- .send(M, tell, said(K, k(X))); ~asking(M, ask(k(X), K)).
+said(K, ~k(X))[source(self)] : asking(M, ask(k(X), K))
  <- .send(M, tell, said(K, ~k(X))); ~asking(M, ask(k(X), K)).

k(X) :- said(K, k(X)) & k(K).
~
k(X) :- said(K, ~k(X)) & k(K).
```

Allocation of agent-roles in hypothetical explanations Our scenario consists of three agents A, B, C, plus a monitor agent M. According to the domain knowledge, we know that scenario agents (apart from the monitor) can play the agent-roles of knights or of knaves. The initial space of explanatory hypotheses is given by all possible permutations, plus a no good explanation $E_*$, giving in total $2^3 + 1 = 9$. 

229
Probabilistic analysis and evaluation  The puzzle proposes an observation $O$ consisting of four messages, strictly ordered:

1. $M$ asking $B$ if he is a knave: $[\text{ask}(k(a), a)]^B_M$
2. $B$ saying that $A$ says he is a knave: $[\neg k(a)]_A^B$
3. $M$ asking $C$ if $B$ is a knave: $[\text{tell}(k(b))]_C^M$
4. $C$ saying that $B$ is a knave: $\neg k(b)]_C^M$

Given $O$, we aim to evaluate the different explanations, as proposed in 7.4.6. We need to find both $P(O|E_i)$ and $P(O|\neg E_i)$. By definition, $O$ is characterized by a content (set of messages $O_{\text{msg}}$) and an ordering (sequence of messages $O_{\text{ord}}$) components. Thus, $P(O) = P(O_{\text{msg}} \cap O_{\text{ord}})$. In the following section we will consider only the first component, that corresponds to an atemporal image of all received messages, but a similar treatment may be done on the second one. Furthermore, in conditions of perfect observability, $P(O_{\text{msg}}) = P(M_1 \cap \ldots \cap M_4)$. Thus, for each explanation (excluding $E_*$) we create a multi-agent system, assigning to the scenario agents the related agent-roles allocation, and we instruct the monitor agent to produce the same stimuli. Consequently, $P(M_1) = P(M_3) = 1$. By executing the model, we acknowledge the production or the non-production of a certain $M$. The outcome of all scenario executions is synthetized on the upper half of Table 9.1.

Assuming a neutral perspective, i.e. indifference towards prior probabilities of explanations, we have $P(E_i) = \frac{1}{3}$. From another point, indifference corresponds to assume $P(k|\neg E_*) = 0.5$. At this point, with some manipulation, we can compute the confirmation factors associated to the observation. Furthermore, these can be calculated on incremental observations, i.e. $O_1$ concerns only $M_1$, $O_2$ includes also $M_2$, etc, and $O$ corresponds to $O_4$. The results are shown in the lower part of Table 9.1, including two other probabilistic settings. $O_1$ and $O_3$ (omitted in the table) are irrelevant, as we expect.

Besides $E_*$, our measure confirms two scenarios ($E_3, E_4$), but the degrees of confirmation vary according the probabilistic configuration of the setting. In the indifference scenario (a), they are all equal. When we assume to be in a world where there are three knights per knave (b), then $E_3$ (the scenario with one knave) is confirmed to a greater degree than $E_4$ (with two knaves). However, the probability associated to $E_4$ is lower than $E_*$, so its confirmation results lower than the control value. But, if we are more confident of our knowledge of the domain, we may decrease $P(E_*)$. In doing that, the confirmation value of $E_4$ will go beyond the control value at a certain moment, as occurs in (c).

---

11 We use the notation $M = [\text{content}_{\text{receiver}}^{\text{emitter}}$, where content is expressed in AgentSpeak(L) syntax. Messages can be encapsulated. The receiver and emitter may be intermediate entities.
12 Actually, another step is necessary after $M_3$. Without previous knowledge, $C$ does not know anything about $B$. We add to him a belief assigning the right agent-role to $B$.
13 This shows how prior assumptions are usually given not counting $E_*$. The no good explanation stays outside from the domain we know (and then also outside our statistical, probabilistic measures). It serves as a sort of control measure on our conclusions.
Table 9.1: Explanations with allocations and observed messages. Confirmation factors in different settings.
Bibliography


235


Index

ability, 83, 86
action, 47, 77
addressee, 47
agency, 73
agent causation, 74
agility, 5
allowed, 67, 69
also-applicability, 104, 135
applicability, 103
attempt, 75
auxiliary clause, 94
auxiliary fact, 107
auxiliary rule, 106
backing, 94
basiswettenbestand, 36
better, 67
bibliographic citation, 116
bibliographic expression, 116
bibliographic item, 117
bibliographic manifestation, 116
bibliographic object, 116
bibliographic source of law, 123
bibliographic work, 116
BPD, 6
bring about, 78
brute fact, 51
burden of proof, 58, 100, 102
bystander, 47
CAS, 6
choice, 71
choice rule, 107
citation, 128
collective, 50
collective entity, 91
competence, 55, 83
consolidation, 123
constitutive act, 52
constitutive rule, 24, 44, 51
contrary-to-duty obligation, 68
deadline, 79
delayed application, 98
delegation, 91, 107
deontic logic, 68
descriptive fidelity, 110, 133
design, 12
disability, 87, 89
disallowed, 67, 69
duty, 88
dyadic, 67
emergence, 78
endophoric proform expression, 127, 130
epistemic role, 59
equality of conditions, 63
evaluative, 66
everyone, 90
ex post facto, 97
ex tunc, 98, 112, 135
exclusive jurisdiction, 108
exophoric proform expression, 127
expectation, 59, 60, 65, 81, 85
first representation, 132
fluent, 78
formal act, 56
FRBR, 132
function, 65
Functional Ontology of Law, 83
Grundnorm, 45
hierarchical task network, 49
Hohfeld, 86
I-intention, 53
immunity, 89
imperium, 107
impotence, 87, 90
inclusion, 129
incompetence, 87, 90
IND, 2
indicative rule, 100
initial version, 123
institution, 43, 50
institutional reality, 24
institutional rule, 51
intent, 100
intention, 47, 52, 53, 74, 80, 86, 97, 105
intercoder reliability, 133
interest, 85
isomorphism, 127, 130
item, 95, 121
judicial review, 135
jural opposite, 86
jural relationship, 86, 88
jurisdiction, 107
knowledge, 11
language variant, 124
legal act, 54
legal fact, 54, 58, 130
legal fiction, 104, 135
legal norm, 69
legal ontology, 54
legal rule, 34, 54, 130
legally relevant, 93
legislation, 43
legislative act, 56
legislative drafting, 74
legislative supremacy, 135
legislator, 47, 55, 62, 70, 75, 106
legitimate expectation, 85
lex posterior, 93, 131
lex specialis, 23, 93
Lex superior, 23
lex superior, 93
liability, 89
man on the Clapham Omnibus, 76
mandate, 107
merging concepts, 134
messages that will (not) be sent between agents, 20
metadata, 94, 95, 110
metalegal knowledge, 84
methodological priority of ex ante intention over ex post intention, 47
models of agent behaviour & social structure, 20
negative causation, 77, 81
norm, 45, 59, 69
norm of analysis, 63
normality, 50, 58
normative conditional, 78
normative position, 82
normative rule, 44, 62, 66, 67, 69, 105
normativity, 58
NTCA, 2
nulla poena sine lege, 97
obligation, 25, 45, 66, 69, 74, 78, 88
ought-to-be, 74
ought-to-do, 74
parliamentary sovereignty, 135
participant, 111
period, 79
permission, 25, 45, 66, 70, 89
plan, 47
plan execution, 48
planning, 12, 49
potestative right, 55, 86, 89, 90
power, 55, 82, 83, 89
practical, 25
preference, 45, 48, 59, 69, 85
preference logic, 69
privilege, 89
problem, 12
problem solving method, 99
prohibition, 25, 45, 66, 69
prospect theory, 60
punishment, 66
reaction, 83
reactive knowledge, 83
reference transaction, 60, 61
regulative rule, 24
repertory grid, 72
representation, 34
restorative justice, 84
retroactive applicability, 97
right, 25, 88–90
role, 46, 75
script, 50, 61, 88
service, 7
shared jurisdiction, 108
sichttag, 98, 112
silent, 67
situation, 78, 81
skeletal plan, 62
SOA, 7
social, 50
social role, 63
source of law, 94
state, 78
state of obligation, 67, 82
stative, 83
status quo bias, 60
stichtag, 98, 112
subdelegation, 107
subjunctive betterness, 67, 70
submandate, 107
term, 34, 130
theft, 67
translation, 124
Trias Politica, 28

uniform resource locator, 117
unique bibliographic identifier, 116
utility theory, 60

variant, 123
version, 123
version in force, 123
victim, 84
violation, 66, 70, 82

we-intention, 53

xml base, 122, 132