Legal theory, sources of law and the semantic web

Boer, A.

DOI:
10.3233/978-1-60750-003-2-i

Chapter 1

Introduction

1.1. Motivation of this Book

> Any system simple enough to be understandable will not be complicated enough to behave intelligently; and any system complicated enough to behave intelligently will not be simple enough to understand. (George Dyson, 2008)

Driven by the Semantic Web, increasing legal convergence, and an increasing pace of organizational change in public administration, the interest in legal knowledge representation in public administrations is gradually increasing but also changing in nature.

Initially interest was focused on the utility of the computer systems built using a knowledge engineering approach; More recently the focus shifted to the potential utility of knowledge representation for comparative and maintenance purposes, and for increasing the efficiency of the organizational change process itself. This change in focus has consequences for the knowledge engineering community.

The concept of *knowledge* representation is strongly linked to the ability of computer systems to make decisions. The basic theses of knowledge representation can be summarized as follows:

1. it is possible to implement computer systems that are capable of making decisions that would require relevant knowledge – in this case of the law – if made by a human;
2. therefore the computer system in some way contains knowledge, in the sense that we *ascribe* knowledge to it: it exists as an agent on the knowledge level (cf. [209]);
3. if this (ascribed) *knowledge base* is made explicit and structured in the right way, then it consists of coherent knowledge components, at least some of which are reusable by different decision-making systems that require the same knowledge (cf. [120]).

The focal point of the Semantic Web vision is however not on improving on the state of the art in automated decision making, but on the efficiency of maintenance and the possibility of sharing and comparing knowledge components. The problem the knowledge engineering community faces is the problem of representing this knowledge in *the right way* in order to achieve reusability outside the original context of use. This is essentially an exercise into making unmanageable complex problems manageable by decomposing them into less complex problems.

Compared to the standards set by knowledge engineering research, fielded systems in public administration and elsewhere that use explicit knowledge representation to support decision making processes are technically and theoretically straightforward. The required transparency, and the challenges real world knowledge representation pose for the people implementing them, act as a natural limit to the complexity of these systems. Since the combination of knowledge of knowledge engineering and law is a rarity in the available work force, the complexities of legal theory and the interpretation of sources of law – like legislation, official directives, and case law – are largely ignored.

This book is an attempt to construct an integrated conceptual framework for the

application-neutral and problem-neutral representation of sources of law using Semantic Web technology and concepts, and some technically straightforward extensions to Semantic Web technology based on established practices found in fielded applications. To construct this framework, I disentangle some problems that are in my view too often mixed up in legal theory and – in extension – legal knowledge representation.

The purpose of this framework is to provide a theoretical background for the creation of reusable and maintainable knowledge components representing knowledge of sources of law on the Semantic Web. These components should form a basis for the development for computer applications supporting straightforward, routine decision making problems using traditional methods.

The next section sketches relevant recent developments, in the field of law and public administration, and the objectives of some projects I have been involved in over the last decade. This section functions as a background against which the relevance of the research themes addressed in this book can be understood.

Section 1.3 introduces the research approach and research themes addressed in this book, and section 1.4 gives an overview of the book itself and the topics addressed in it.

1.2. Sources of Law and the Semantic Web

The increasing ubiquity of the Internet in society has put a pressure on public administrations to make previously paper-based services and sources of information available on the Internet. If you can do your bank business online, why should you expect less from your government?

The migration of public administration to the Internet is loosely referred to as *eGovernment*, or electronic government. The enabling technology spurring recent interest in electronic government is the *Semantic Web*, a vision of a next-generation World Wide Web – complemented with certain design principles and core technologies – focused on loosely connected *services* and *knowledge components* instead of pages.

The services in question are in an administrative setting generally public legal acts, performed by public legal personalities, based in formal legislation. Legislation defines what the core functions of public organizations are, and what services it provides. It guides how the organization subdivides itself into administrative units, how it organizes business processes inside the organization, and eventually how the functions of the organization are realized by civil servants and computer systems. Everything the public administration does that cannot be done by a private organization or person is normally speaking based in legislation.

In many cases eGovernment is an internet-based portal connected to old, slow, mainly paper-based decision making processes. But electronic government cannot be just a matter of mere window-dressing. Inside public administrations, and on the interfaces between them, ICT and Internet also have a large impact. Some decision making processes are nowadays assisted by computer applications, and others are more or less autonomously performed by the computer. Many tax declarations – sometimes more than 80% of them – are for instance never handled by a human being. The computer decides, randomly or based on properties of the declaration that suggest

intentional or unintentional misrepresentation, which declarations should be looked at by a civil servant.

Business process design and design of specialized computer systems are both usually based on explicit *models* in modeling languages, like the Unified Modeling Language (UML) or, more recently in the context of the Semantic Web, the Web Ontology Language (OWL), of what the business process or computer application should achieve. These models are supposedly used as a specification of the objectives of an organizational change process or application development process. When legislation changes, these models are updated, and the organization's structures and computer programs have to be changed to conform to the models.

In the past these changes were conceived of as temporary interruptions of long periods of everything staying the same. This was certainly the case when the adaptation of existing computer systems was still considered a frightening prospect: things did change but the changes where carefully orchestrated to not impact existing procedures and computer systems. But as the perceived capacity of organizations to organize change processes increases, and the number of fielded computer applications increases, so does the pace of change in legislation directly affecting existing computer applications.

Tax legislation is for instance changed every year, leading to continuous adaptation of relevant computer applications for next year and the years after that, while the legislation of the present year and previous years is still being applied. In the business process design literature, awareness of this phenomenon has led to a new conception of the organization as an entity that is constantly in the process of changing: the organization is constantly conceptualizing and comparing what it is and what it is becoming.

Attention for *knowledge representation* of sources of law – the subject of this book – is very often triggered by such administrative change processes driven by new legislation and other sources of law, such as case law and internal written policies. Knowledge representation is seen as a means to potentially reduce costs and increase efficiency through increased control over the knowledge dimension of the change process.

The increased focus on change is however not just driven by technological possibilities and pipe dreams, and the perception that the capacity to change has increased. The change processes triggered by the legal system are increasingly expensive, especially if they involve changes in ICT infrastructure.

The process of drafting consistent, coherent, and effective legislation itself is indeed getting more complicated, as is that of efficiently using, enforcing and applying valid law. The very fact that legislators recognize each other's activities as legitimate, embrace *legal pluralism*, and want to cooperate and compete with each other, tends to complicate the operations of law.

The increasing legal convergence between governments in the European Union, and the increasing traffic of people over borders of jurisdictions, inevitably leads to an increased interest in the problem of knowing, comparing and harmonizing legislation.

Administrations need to know and understand legislation of friendly governments to be able to assist citizens and reduce negative impact of permitted movements between jurisdictions.

Employees of tax administrations in the EU are for instance increasingly confronted with requests that require them to understand European regulations and directives, and regulations of other EU member states; They need to react to the consequences of increased movement of people, products, and money between EU member states and increased harmonization between tax authorities in Europe.

Subject to certain conditions and exceptions, value added tax for instance has to be paid only once in the member state of choice of the owner of the goods, according to a EU directive. The directive has as a costly administrative side-effect that tax administrations employees suddenly need to be able to apply the VAT rules of all 25 member states to know whether the conditions have been met or the exceptions apply.

Today's public administrations do not yet have an effective answer to demands such as these. And neither does the market.

The market is solving its own comparison problems. Global companies offer products and services in many jurisdictions at the same time, and the product or service has to meet the provisions of all jurisdictions in which it is offered.

The issue is however not just one of compliance. Different regulations also lead to large differences in competitiveness for the product or service. For a financial product, for instance, it is considered important to qualify for tax deductions that make the product more attractive. For a medical product it is for instance important to know whether it will be possible to sell it over the counter without a prescription. For computer applications it is for instance important to know to what extent it will be protected by copyright, and which license conditions are legally enforceable. For outsourced administrative services it is for instance sometimes imperative that privacy of client data is legally protected to the same extent as it would be in the jurisdiction where the client data comes from, and this limits from and to which countries services can be outsourced.

This increased attention for comparing legal systems is for instance evidenced by the number of consultancy firms that advertise their knowledge of multiple legal systems to companies on the World Wide Web. In addition, there are some initiatives – often initiated outside the Computer Science and Law community – for constructing international legal *ontologies* that expose subsumption relations between legal vocabulary in multiple jurisdictions, and make law comparable, on the Semantic Web.

For public administrations the idea of the Semantic Web has great potential. Public administrations that provide different services based on the same legislation can share the effort of making a knowledge representation of that law, or, even better, delegate that responsibility to the market or even to the legislator: future legislation may come with an embedded knowledge representation, and of course over the Internet, to its institutional users.

In addition to that, if you can use knowledge representation for, let's say, calculating how much a taxpayer owes the government in income taxes, and the legislator makes available a new version of the relevant knowledge representation, then it would obviously be possible to compare the old and the new version to see what has changed. Even more ambitiously, if a third country sends a knowledge representation of its solution to the problem of taxing income, then the administration would appear to be able compare it with its own knowledge representation, and more or less automatically

deduce what relevant differences there are.

My recent work for the Dutch Tax and Customs Administration (DTCA; cf. [42]) was for instance clearly related to the huge *change process* triggered by the complete overhaul of the Dutch income tax law in 2001. DTCA knowledge representations intended for use in decision support systems were expected to specify – and justify – how business processes were grounded in legislation, and provided input for the drafting process through formal verification and simulation of draft legislation. The *Juridisch Loket* (cf. [272]) project on *pro bono* legal assistance, and the *DURP* project on spatial planning (cf. [40]) were also driven by an overhaul of legislation.

An ambitious quest – one we tried to realize for the DTCA – is to integrate the process of legislative drafting, including evaluation of formal and executable models of legislation, and deployment of these formal models as software for the Dutch Tax and Customs Administration, into a single architectural framework. Although the project was successful – at least by the standards set for such projects – there is still a lot to do.

Some of the recent knowledge representation projects I was involved in (E-POWER, Estrella) included the *comparison* of representations of similar pieces of legislation from different jurisdictions. Although pairing up different decision support systems solving the "same" problem is certainly a way to gain insight into relevant differences, the field of legal knowledge engineering really has no theoretical framework for comparison of knowledge representation for the purpose of exposing relevant differences in the law.

The idea that this is possible and meaningful, presumes a lot of the degree of inter-coder reliability that can be achieved by knowledge engineers: if there is *no* difference in the source of law represented by two different knowledge representations from two different authors, do we really expect no difference in these knowledge representations? This calls for strong methodological commitment and standardization of approach in legal knowledge representation, which does not exist in practice.

1.3. Research Approach

The Semantic Web, and the assumptions behind it, demand a lot of the structure of knowledge bases.

The Semantic Web is no longer about the knowledge base, as the driving force behind a computer application that is capable of making decisions that require knowledge, but about the flexible interaction between knowledge components made available and maintained by different suppliers. Applications will no longer contain all knowledge they use, but instead they will download the latest version of knowledge components from various locations on the Internet on demand. This is the idea of *federalized* knowledge bases, or of *knowledge syndication* when considered as a business model.

This book is a reflection on the representation of knowledge about the content of sources of law. The following assumptions are taken for granted in this work:

1. *legal* knowledge is mainly based on the formal sources of law, although there is also *legally relevant* knowledge, and;

2. the sources of law are the main locus of legal change, and therefore the appropriate locus of maintenance activity.

Certain design principles based on these assumptions have to be accepted as a given. Reusable components aim to represent a specific source of law, and the content of the component can be traced to and justified by the source of law – this is *isomorphism* as it is used in Computer Science and Law. Maintaining isomorphism to a high degree is one of the tools we have as knowledge engineers to increase methodological rigor and make maintenance manageable in knowledge representation.

An appropriate selection of these components, plus additional relevant knowledge from other sources, form the knowledge base of a decision support system. A new consolidated version of the source of law corresponds with an updated knowledge component that replaces the previous one, within the existing structure of a knowledge base, while minimizing impact on the other knowledge components present in the knowledge base.

This is a more or less traditional setup, which will be treated as a given – as part of the problem – in this book. This is the role legal knowledge engineers can succesfully adopt in the Semantic Web: represent the sources of law, not the whole of legal problem solving, and one meets the expectations and needs of the market and respects the implicit division of labor presumed by the Semantic Web vision.

This book is a reflection – not a *report* – on ten years of applied research in legal knowledge engineering. Throughout the last ten years I have formulated answers, in the form of a conceptual framework, to three related questions relevant to the Semantic Web idea as applied to sources of law:

1. If a reusable knowledge component is to be used in different settings, and to help realize different problem-solving competences, what part of a knowledge representation is the reusable part and what part is specific to the setting or problem to be solved? This is an inquiry into what is *essential* in knowledge of the sources of law, based on the assumption that this quality determines the degree of reusability of knowledge.
2. How does one integrate existing design principles, core technologies, and technical standards of the Semantic Web – such as its naming and addressing standards, based on the uniform resource identifier or URI, and the description logic OWL DL – with legal knowledge representation?
3. What knowledge should be considered specifically legal, and what knowledge should be considered as a given? This is in essence an inquiry into the appropriate division of labour: the notion of *federalization* of knowledge representation presupposes that reusable knowledge components should be made available by parties specifically competent to do so;

This book adopts the strict distinction between *ontology* – about the *object* of our knowledge – and *epistemology* – about the processes by which knowledge is attained – maintained by for instance by Breuker i.a. in [67]. The distinction between these two aspects of knowledge will be worked out throughout chapter 2.

The appropriate object of study when reflecting on knowledge components for the Semantic Web is *ontology*. Ontology cannot be completely disconnected from

epistemology – and this book will address practical reasoning approaches and problems where relevant – but one can decide to be extremely cautious when making ontological commitments. The motivation for making ontological commitments should not be to achieve a certain level of problem solving competence. The pragmatics of problem solving should be kept at arm's length.

The conceptual framework sketched in this book will therefore leave noticeable gaps if one approaches it as a grand, unified theory of legal reasoning: these are intentional, and filling them in is left as an exercise for each individual reader.

Many individual solutions chosen in this book have a pedigree in literature, and certainly do not aim to improve on the state of the art in automated legal reasoning. As a conceptual and methodological framework, and an attempt to make legislation Semantic Webbable, this work is however new.

Law must be the knowledge domain *par excellence* for testing the concepts of the Semantic Web. Legislation, with its internal references, import mechanisms, applicability conditions, and intricate version management solutions, is the closest thing to a tested and tried, paper-based semantic web in existence.

The Semantic Web is however not designed with law and legal knowledge representation in mind, which brings us to the second question.

The Semantic Web technology of central importance for knowledge representation is OWL DL, a reasonably expressive subset of the Web Ontology Language that can be interpreted as a first order monotonic *description logic* with reasonably attractive computational properties. OWL DL does not support propositions about propositions, for sound engineering reasons.

The monotonicity of OWL DL is attractive from an engineering point of view: it permits problem decomposition in knowledge representation. Monotonicity is essential for the idea of composing knowledge bases from separate knowledge components, in analogy with the design of physical artifacts from physical components with a transparent and predictable interface.

Classical monotonic logics are however not very plausible models of real world decision making. Sometimes we can prove something once and for all; Most of the time we can only construct falsifiable arguments. If we construct these arguments from formal rules, then these rules must be *defeasible*. This is true as much in law as it is true in any other domain of enquiry, but in legal practice, where reflection on the applicability of rules is common, it is especially obvious.

Moreover, the sources of law even explicitly address the falsifiability of legal argument, and occasionally formulate constraints on reasoning that explicitly presume the possibility of falsification of what would otherwise be acceptable as a valid argument. The formulation of these constraints often involves propositions about propositions.

Even if one decides *not* to solve the problem of defeasible reasoning, and accepts that there are many more or less valid ways in which one can – defeasibly – interpret and use the same source of law, perhaps even the same knowledge representation of it, one should at least try to represent these constraints in order to maintain isomorphism of knowledge representation and source of law.

The third question – the distinction between legal and non-legal knowledge – can be approached both from the point of view of *ontology* as of *epistemology*.

The *epistemological* dimension of this question is the more important one. A basic assumption about epistemology underlying the conceptual framework sketched in this book is perhaps a provocative one: *legal reasoning* as such – as a special kind of mental process – does not exist, and we do not have to account for a special legal epistemological theory.

Everybody understands that representation of the traffic rules is not enough to design an intelligent system that can drive a car; There is no reason to expect that law does give rise to a *legal logic* that accounts for legal reasoning. Legal reasoning is straightforward common sense reasoning, but with legal knowledge.

On the other hand the law instructs us not only on what to do but also often on how to reason. But law only restricts the degrees of freedom we have in reasoning, through explicitly formulated rules; We do not have to interpret it in terms of a theory of legal reasoning.

If the Semantic Web does not accommodate defeasible reasoning, one shouldn't have to introduce it to accurately represent knowledge of the law. Falsifiability is not unique to law. If OWL DL is good for anything else, it is also good enough for law. The same applies to higher-order reasoning involving propositions about propositions: OWL DL does not accommodate it, but reflection on propositions is not unique to law. The objective of this book is to represent legal knowledge in OWL DL, without extending or changing the semantics of OWL DL.

Constitutiveness and applicability are for instance familiar concepts in legal knowledge representation, but are normally conceived of as higher-order observations about propositions. In this book they are treated as first order properties of objects: legal acts and other legal *things* take logical priority over legal *facts*, which are left implicit. *Legal rules* are indeed reified propositions if one wishes to see them in that way, but they are never interpreted as propositions in reasoning.

The question is how one can preserve a reasonable degree of descriptive fidelity in knowledge representation without these devices. Special legal logic is unnecessary for most fielded applications, which rely heavily on context of use and a very narrowly defined problem solving competence to avoid complex reasoning techniques.

There is of course also an *ontological* dimension to the distinction between legal and non-legal knowledge: for this purpose the concept of the law as a set of institutions is introduced. The source of law uses legal terms that belong to the institution, and *legally relevant* ones that are on the interface between the institution and the outside world. The institution is distinguished from the normative order it intends to create. Normative order is the ultimate function of law – and an essential one – rather than a part of it. Important parts of the normative order are found only *between the lines* in the sources of law.

The first and central question – how to design knowledge components representing sources of law for *reusability* – is of course not one which can be definitively answered. There are a number of known threats to reusability, and we can do our best to contain them.

I already addressed the problem of keeping the pragmatics of problem solving at arm's length. There are certain relevant contextual assumptions that affect the way in which the knowledge engineer represents knowledge for a specific problem

setting. An important objective of this book is to make a survey of relevant concepts in legal theory with the approach chosen for this book in mind, in order to remove – often epistemological – aspects of these concepts that are non-essential for the mere *representation* of the sources of law in the form of rules.

The distinction between the legal and non-legal made by the second question addresses another reusability concern: if legal knowledge engineering invents its own solutions for every relevant aspect of legal reasoning, or any kind of relevant domain knowledge, it is likely to end up with products that are hard to fit in as a component in more general knowledge engineering projects. Law is more often than not only *part* of the problem context. Respecting existing, more general approaches, even if they are hard to use, should be an objective in itself for knowledge engineers.

A trivial reusability concern, already addressed by the first question, is a lack of technical interopability. Semantic Web standards make technical interoperability possible, but at the same time impose rather arbitrary restrictions on the expression of knowledge. Remaining within the limits set by the standards should be an objective in itself.

Most pressing is however giving falsifiability in general a place without declaring everything potentially falsifiable. Unchecked falsifiability is a license for bad engineering, or for declaring reuse of knowledge in principle impossible. As will be demonstrated in this book, defeasibility is not only a problem because OWL DL does not allow it. Defeasibility is also a central issue for the distinction between ontological knowledge and other, contingent knowledge, and even the distinction between legal and legally relevant knowledge can be understood in terms of defeasibility.

1.4. Outline of this Book

Chapter 2 of this work examines the field of knowledge engineering. It starts gently, for readers not familiar with the field, with an introduction into knowledge representation and knowledge-based systems. The chapter also sets the stage for the rest of the book: it introduces ontology, epistemology, ampliative or abductive reasoning, ontological stratification, and abstraction as used in this book.

Chapter 3 presents the idea of a knowledge component on the Semantic Web, based on OWL DL extended with some reasoning mechanism for autoepistemic reasoning. It defines the concept of a source of law, the object of the knowledge to be represented, and introduces certain notational conventions and an important distinction between several different types of logical rule used throughout the book.

Chapter 4 gives an account of certain fundamental theoretical concepts and issues in law and legal knowledge engineering. It introduces institutions and their rules, normality, and normativity, and explains some relations between the rules of the institution, intention, and action. The primary purpose of this account is to separate the rules themselves from assumptions about their use in planning. For general knowledge engineers it serves as an introduction into the theoretical problems of this field. For legal theorists most concepts will be well known, but the chapter will give a new perspective on some issues.

In this work the device of *ontological stratification*, introduced in chapter 2, is used to structure legal knowledge, inspired by the use of institutions and constitutiveness in legal theory. Stratification also plays a key role in containing defeasibility in reasoning: the only inferences that are permitted to be defeasible are those that bridge ontological strata. The central stratum of chapter 4 is the legal institutional reality.

Law is a domain of *institutions* whose main purpose is to create an intended normative order by formalizing it. The legislator as engineer of his own institutional reality is responsible for the rules and structures of the institution being coherent and transparent, understandable, justifiable, and capable of effectively and efficiently achieving the intended normative order. The legislator shares the problems of the legal knowledge engineer: both manage an intangible and very complex artifact whose structure must be designed for understandability, transparency, reusability, maintenance requirements, etc.

A central ontological theme of this book is the representation of the institution, and the relation between the institution and the formal sources of law that express it. What legislative action does, is to change the rules and structures of the institution. What the legislator *intends* is however to change prevailing normative order, towards an intended one, as chapter 4 explains.

Chapter 5 focuses on the logical representation of information about legal rules and the problems legislators encounter in managing a large body of legal 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. The chapter is the main body of the work from a knowledge representation point of view.

In this chapter the *legal rule* is dissociated from the fragment of a source of law that represents it and from both the logical rule of chapter 3 and the rules of the institution in 4. It also refines the relationship between the logical rules introduced in chapter 3 and the rules of the institution.

The appropriate relation between the source of law *qua expression*, the legal rules *qua institutional entity*, the intended effect of those legal rules on the normative order, and the logical sentences of the knowledge representation is a central theme in this work. My work on MetaLex, also reported in chapter 5, revolves around the sometimes non-trivial relation between changes to the sources of law and changes to the rules and structures of the institution.

Chapter 5 concludes with a reflection on the relationship between sources of law and knowledge components composed of logical rules.

Chapter 6 discusses the normative order the legislator intends to create through the institution and its rules. In this chapter the most familiar ingredient of normative order created by law – the normative rule – is worked out in OWL DL. The chapter also argues that it is necessary to ascribe an intended normative order to the actions of the legislator to explain non-trivial varieties of legal reasoning such as court adjudication and comparative law.

The usefulness of decomposition of systems into reusable components largely depends on the recognition of the functions of the individual component. These functions of the source of law can be inferred from the intended normative order, in the understanding that ascribing such intentions to the legislator is not a clear-cut problem. The

representation of normative order, beyond the fragment made explicit by normative rules, still presents us with a number of open research questions; Chapter 7 presents some suggestions for further research into the representation of normative order, besides its obvious purpose of recapitulating the essentials of the preceding chapters.

Throughout this book regular references are made to two Semantic Web-related public specifications that I have been closely involved in: the MetaLex CEN/ISSS standard and the Legal Knowledge Interchange Format (LKIF) ontology. Appendix A positions this work in relation to MetaLex and the LKIF ontology.