Downloaded from UvA-DARE, the institutional repository of the University of Amsterdam (UvA)
http://dare.uva.nl/document/136081

File ID 136081
Filename Chapter 5: Representation of sources of law

SOURCE (OR PART OF THE FOLLOWING SOURCE):
Type Dissertation
Title Legal theory, sources of law and the semantic web
Author A.W.F. Boer
Faculty Faculty of Law
Year 2009
Pages IV, 321

FULL BIBLIOGRAPHIC DETAILS:
http://dare.uva.nl/record/305854

Copyright

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.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)

Chapter 5

Representation of Sources of Law

5.1. Introduction

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 4.4.1 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.

This chapter focuses on two issues:

1. the logical representation of information about legal rules; and
2. 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.

One form of logical representation will conspicuously not be addressed: the normative rule. Normativity is one of the subjects of chapter 6.

Section 5.4 functions as a summary of the important representation decisions made in the rest of the chapter, and relates these to the notion of knowledge components representing sources of law.

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 [260]).

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 of section 6.6 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) of section 6.6 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 propositions* 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 propositions shape and regulate the domain of legislating itself. In the work on *MetaLex*, discussed in section 5.3 (cf. [53, 41]), they are of central importance: relevant *metadata* (cf. section 3.2) is usually found in auxiliary propositions.

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 propositions 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.

[1] Compare Dutch *hulpbepaling*.

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

Publication of legislation, and the development of tools for working with legislation is at the moment still a jurisdiction-specific enterprise, even if it is standardized at the jurisdiction level. What is required is a jurisdiction-independent XML standard that can be used for interchange, but also - maybe more importantly - as a platform for development of generic legal software.

For vendors of legal software this opens up new markets, and for the institutional consumers of legislation in XML it solves an acute problem: how to handle very different XML formats in the same IT infrastructure. Increasing legal convergence between governments in the European Union, and the growing importance of traffic of people, services, goods, and money over borders of jurisdictions has led to an increased need for managing legislation from different sources, even in public bodies and courts. EU tax administrations for instance need access to all VAT regimes of other member countries to correctly apply EU law, and EU civil courts may nowadays for instance be confronted with the need to understand foreign law on labour contracts to decide on cases involving employees with a foreign labour contract choosing domicile in the country where the court has jurisdiction.

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.

This book is inspired by 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. LKIF is yet to be submitted as a proposed standard, and this book deviates in important ways from LKIF.

MetaLex XML is well on its way to becoming formal and de facto standard for legislation in XML. MetaLex XML is also closer to the views of the author of this book: 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.

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 (see section 3.5.1). It is the

result of a legislative act performed with the intent of creating that rule, and functions as evidence of that legislative act.

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*. [260] notes the problem:

> Note a peculiar feature of any reflexive sunset proposition, for example, proposition 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 proposition 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 proposition did not completely swallow itself. Statute S, including sunset proposition 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 4.4, 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). OWL axioms and default 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 RDF statements, 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 – [245] 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

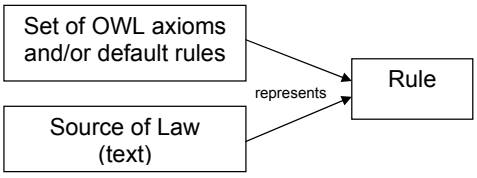

Figure 5.1. Both the original source of law and the knowledge representation are representations of a third object: the rule.

source of law once existed functions as evidence of the existence of a rule[2].

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 authoratively – 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. [39], 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.

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 [196, 217, 216, 218]).

Applicability statements, as found for instance in section 6.2, are usually of the second type. The first type has a purely auxiliary function, and is usually of lesser

[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.

importance in LKBS. Since the context of use of the LKBS 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 LKBS 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* propositions 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 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 clear-cut 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.

[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 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[4]. It is acknowledged that this timeline may be retroactively changed by the legislator, but the timeline is always clear-cut 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 analyzing the history of a document.

5.2.1. Representation of Legal Rules

Knowledge representation of sources of law consists of formulating logical rules *about* legal rules. The knowledge representation rule is not a direct representation of the legal rule. The central question is whether the special requirements of legislating or legal reasoning tell us something about the relation between the logical form of knowledge representation and the functions of the legal rule the knowledge representation is about.

There are indeed some useful recurring patterns revolving around the applicable and constitutes properties. The constitutiveness patterns were already addressed in chapter 5 in relation to constituve rules. Applicability, discussed in the next section, will link the (logical interpretation of the) constitutive rule to the legal rule as an object mentioned in the rule. Constitutive and institutional rules do not need to be strictly distinguished from their logical representation, since we do not need to represent rules about their applicability; Constitutive rules can be applied if it is consistent to do so, and institutional rules must be applied because they are ontological.

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

[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.

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. Section 4.4.1 showed a worked out example, that however contains some simplifications.

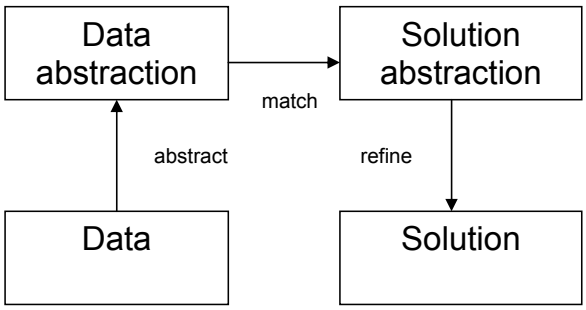

Figure 5.2. Heuristic Classification.

For a functional classification we can find some inspiration in Clancey's well known horseshoe diagram in figure 5.2 (cf. [81]), which describes a simple problem solving method (cf. section 2.4) 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'.

As pointed out in section 2.3.1, 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 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 [272] we describe a system we developed for the *Juridisch Loket*[5] in the Nether-

[5]previously *Bureau voor Rechtshulp*

lands, 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 [272] 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 indicator and requirement pattern was also used in section 4.4.1.

The proposition made here 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 LKBS[6]. 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. The distinction between indicators and requirements is purely a distinction on the basis of logical form: they are both constitutive rules since they specify a mapping from a brute reality to an institutional one.

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[7]. 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 it's 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 LKBS not only implements the *legal rules* the interface shows, it also represents a policy decision 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).

Compare this to argumentation: indicative rules are used to construct pro arguments, while requirements are used to attack them on the grounds that the conclusion only follows if some necessary condition is met. In this case the burden of proof de-

[6]I.e. "customer wants to create this fact" is not part of the explicit conditions on the adviser's screen.

[7]My own observation.

termines 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 (cf. for instance [53]).

Summarizing, we distinguish between institutional and constitutive rules based on their content, dependant on our view about the boundaries of the institution. Constitutive rules can be further distinguished by the direction of the logical mapping into indicators and requirements. Lastly, we still have the integrity constraints of section 4.4.1 to represent a specific burden of proof assignment to a KBS.

The source of law contains rules of the following logical types (notation following section 3.4):

Institutional rules: Terminological axioms are of the form $r : \alpha \sqsubseteq \beta$ or $r : \alpha \equiv \beta$, where α and β 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 4.4.1, constitutive rules sometimes specify necessary conditions on the mapping between institutional reality and brute reality. Necessary rules are of the form $r : \alpha \sqsubseteq \exists\mathsf{constitutedBy}.\beta$, where α and β are concepts, and α 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 following form, where α and β are concepts, and β wholly belongs to the institutional ontology of the institution the legislator intended to create the rule for:

$$(\textsc{Default}\ r\ (\mathsf{known}\ \alpha)(\mathsf{free}\ \exists\mathsf{constitutes}.\beta)(\mathsf{assume}\ \exists\mathsf{constitutes}.\beta))$$

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, the LKBS is normally based on a specific autoepistemic model (cf. section 3.4.2) that must be settled before a decision follows. In some cases such requirements are explicitly found in legislation. The burden of proof assignment is always a rule in accordance with the possibilities sketched in section 3.4.2, 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. Section 4.4.1 shows an example of a constraint.

The constitutes (inverse constitutedBy) property identifies the boundary of the institution.

Normative rules are distinguished by their (institutional) subject matter – allowed or disallowed – and when represented consist of *both* requirements and indicators. This is the reason not to add them to this list. Section 6.2 presents them in detail. 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 LKBS:

1. As pointed out in section 4.4.1 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 ($C \sqsubseteq N$) there is *another* argument for N besides the trivial and tautological ($\{C, C \sqsubseteq N\}, N$).
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 used in the LKBS 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 LKBS (does it know all relevant rules for determining C, or is there reason to believe that some are unknown to the designers of the LKBS?) and policy decisions to be made by the LKBS user (what evidence do we routinely accept for C, and are we open to unforeseen categories of evidence in the main production process?[8]).

LKBS 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:

[8]The administrative organization may decide, and it often does, that although there may be unforeseen but valid arguments, the appeals process will sort this out.

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 (cf. section 6.6.3), 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 arguments based on the format given in section 3.3, 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 $\mathsf{represents}(r, r')$, and keep track of the fact that ({r,. . . }, 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 the applicable (inverse appliesTo) property, always attached to the object the logical rule is about (i.e. the condition). The following is an example of a representation of a requirement, where C is the institutional qualification and N the necessary condition:

$$C \sqsubseteq \exists\mathsf{constitutedBy}.N \sqcap \exists\mathsf{applicable}.\{r\}$$

The rule states both its effect and the fact that it is applicable. The institutional rule obviously has the same form.

The following does the same with an indicative rule, where I is the indicative condition:

$$\begin{aligned}(\textsc{Default}\ r\ (known.I)(free.\exists\mathsf{constitutes}.C \sqcap \exists\mathsf{applicable}.\{r\})\\ (assume.\exists\mathsf{constitutes}.C \sqcap \exists\mathsf{applicable}.\{r\}))\end{aligned}$$

The belief b that institutional fact C only arises through the application of r_1 or r_2 (assume $r_1 \neq r_2$ for any pair of rules) would for instance become, if it were an axiom:

$$C \sqsubseteq \exists\mathsf{constitutedBy}.\exists\mathsf{applicable}.\{r_1, r_2\}$$

Unsurprisingly this rule takes the form of a necessary condition, but it is still not required that it is known whether r_1 or r_2 applies. Only that it is so. This rule format cannot be safely used. A corresponding constraint rule, which does have the required meaning is trivial to reconstruct:

$$\begin{aligned}(\textsc{Constraint}\ r_2\ (\mathsf{known}.C)\\ (\mathsf{fails}.(\exists\mathsf{constitutedBy}.\exists\mathsf{applicable}.\{r_1, r_2\})))\end{aligned}$$

[9]This is however obviously not a terminological truth: we often accept institutional facts simply because we trust the source of information.

This rule says that if C is ever asserted, we must *know* that it was derived from the application of r_1 or r_2. If the axiom above would have been present we would again get a trivial argument: we know one of the rules was applied because the axiom says so.

5.2.3. Representation of Applicability Rules

The applicable (inverse appliesTo) property can obviously also be used for applicability restrictions. Applicability statements (cf. [230, 231]), statements of the form *article n is [not] applicable to C*, are similar to the reading of requirements as necessary conditions. The applicability statement is trivially represented (for instance to say that r applies to the acts of civil servants):

$$\{r\} \sqsubseteq \forall \mathsf{appliesTo}.\exists \mathsf{actor}.\textsc{CivilServant}$$

Time limiting the application to occurrences is also trivially done:

$$\{r\} \sqsubseteq \forall \mathsf{appliesTo}.\exists \mathsf{causes}\exists \mathsf{extension}.\exists \mathsf{after}.T$$

Similarly, any restrictions on the matter to which a rule of set of rules is applied can be represented as applicability constraints.

A generic format for stating that r_1 states that r_2 (only) applies to C is:

$$\{r_2\} \sqsubseteq \forall \mathsf{appliesTo}.(C \sqcap \exists \mathsf{applicable}.\{r_1\})$$

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. Since the specific applicability constraint on a rule r is based on a corresponding nominal concept $\{r\}$ – the concept that describes the singleton set $\{r\}$ – the logical form of such constraints is also clear:

$$\textsc{PartOfR} \sqsubseteq \forall \mathsf{appliesTo}.(C \sqcap \exists \mathsf{applicable}.\textsc{PartOfR})$$

PartOfR is a generalization of a category of rules, all rules part of R:

$$\textsc{PartOfR} \equiv \exists \mathsf{partOf}.R$$

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 this means "applies to actions of civil servants only where it concerns actions and civil servants only where it concerns actor roles". A little example for the actions:

$$\mathrm{R} \sqsubseteq \forall \mathsf{appliesTo}.(\exists \mathsf{actor}.\textsc{CivilServant} \sqcup \neg \textsc{Action}) \sqcap \exists \mathsf{applicable}.\mathrm{R})$$

In reality such broad scoped 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 4.4.1 as a less preferred solution), or a bizarre *legal fiction*.

Legal fiction is occasionally discussed in legal knowledge engineering literature under the heading of *deeming provisions* (cf. for instance [21, 238]).

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] (cf. for instance [21] for a similar approach, or [238] for another one that does not however address the fundamental problem but only signals its presence).

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].

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,

[10] The legislator may change legislation with the intent of destroying such an established fiction. This may lead to the introduction of new propositions 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 *Aanwijzingen voor de Regelgeving* 1990, art. 61, in the Netherlands, but this is no guarantee 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.

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*, which is discussed in section 6.2 in chapter 6, in which the intended normative order will be the topic of discussion. 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.

Normative rules are however not the only mechanisms through which the legislator manipulates the normative order in a society. Section 4.4 proposed a classification, based on [244], 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 is a condition for application of the rule:

$$\textsc{RaiseHand} \sqcap \exists \mathsf{executes}.(\textsc{RaiseHand} \sqcap \exists \mathsf{constitutes}.\textsc{Bid}) \sqsubseteq \exists \mathsf{constitutes}.\textsc{Bid}$$

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 intended task, then there is an applicability rule that says that the rule only applies to action performed to execute T:

$$\{r\} \sqsubseteq \forall \mathsf{appliesTo}.\exists \mathsf{executes}.T$$

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 6.2: the power is always weak, and its systematic protection can only be organized through non-infringement on it by legislators.

The last recognized category from [244] 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 [265], 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 propositions 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:

$$\{c_1, c_2, c_3\} \sqsubseteq \textsc{Court}$$

And a judge in court c_1 may for instance be appointed by royal decree, creating an institutional fact, i being the individual:

$$\{i\} \sqsubseteq \exists \mathsf{plays}.\exists \mathsf{memberOf}.\{c_1\}$$

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 X'[12]. 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 6.6.3. Choice rules are necessary because the legislator cannot guarantee that pairs of rules will never be contradictory.

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 propositions 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. [38, 289])[13]: this regulates *jurisdiction*, at least in its meaning relating to legislators[14], and which sources of law depend on other sources of law in terms of shared institutional ontology. There is a limited number of legislators whose legislative competence is pre-existent, which includes at least internationally recognized national governments. Their rules de facto apply to a certain groups of

[12] *Aanwijzingen voor de Regelgeving* 1990, art. 184–185

[13] This conceptualization is borrowed from constitutional and administrative doctrine in the Netherlands.

[14] I have once proposed to rename this to *imperium*, and to limit *jurisdiction* to the courts.

persons (personam), in a certain territory (locum), and – de facto – in a certain time interval.

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 LKBS 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:

$$\exists \mathsf{initiatedBy}.\exists \mathsf{causedBy}.(\exists \mathsf{actor}.\text{NLGovernment} \sqcap \forall \mathsf{actor}.\text{NLGovernment}) \sqsubseteq \forall \mathsf{appliesTo}.\exists \mathsf{actor}.\exists \mathsf{playedBy}.\text{PersonUnderNLJurisdiction}$$

This example is a gross simplification. Don't take it as a suggestion of how to do this. This type of jurisdiction if firstly determined by the territory (extended to ship and planes choosing that port flag) where an occurrence happened, with some specific extensions to jurisdiction over persons.

The important part is the form of the condition: the legislator simply 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 propositions in legislation.

[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.

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 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[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 front page of the MetaLex web site[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. [53, 41, 40, 38, 37, 36, 285, 48, 51, 52, 49]. 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. [37]), has been redesigned from scratch in the CEN standardization workshop, taking into account lessons learned from *Norme in Rete*[18] – the Italian standard for legislation – and *Akoma Ntoso*[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) now exists. It does not yet constitute a complete, workable XML standard. 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. Additional agreements are on the agenda. The description in this section reflects the status of proposals in the summer of 2008, and mainly those features of the standard of interest to and mostly designed by the author of this book.

The *CEN Workshop on an Open XML Interchange Format for Legal and Legislati-*

[16]Comité Européenne de Normalisation; European Committee for Standardization; Europäisches Komitee für Normung.

[17]http://www.metalex.eu

[18]http://www.normeinrete.it/

[19]http://www.akomantoso.org

ve 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. The scope of the standard covers exactly those things covered by the scope statement in section 3.5.1 of this work.

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. [245]): 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 interpretation 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. Of course chapter 4 argued the priority of actions over states in detail.

In the MetaLex CEN workshop (see e.g. [53]) the widely used classification of *event participants* by Judith Dick (cf. [96]) 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 [182]): 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 [51]). 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 relevant references between documents. Explicitly identifying events increases the reliability of the metadating process.

As noted in [48], 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 corrige, 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 [51] 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 organization, 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 *occurrent* 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.

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,

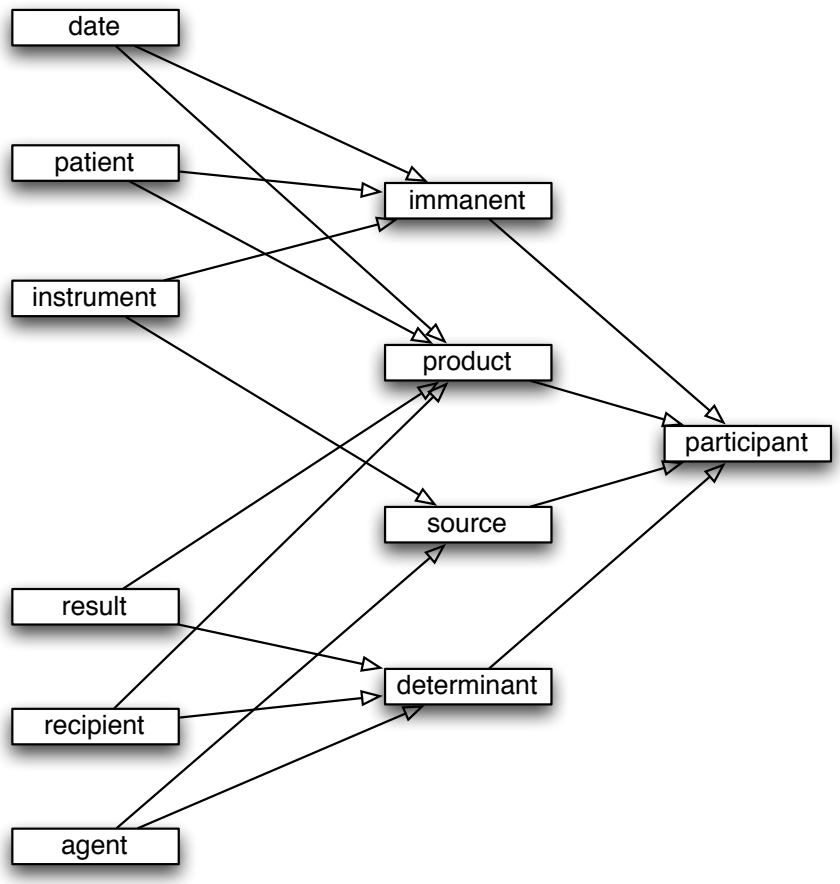

Figure 5.3. Each MetaLex occurrent has one or more participants. The figure shows a taxonomy of participants.

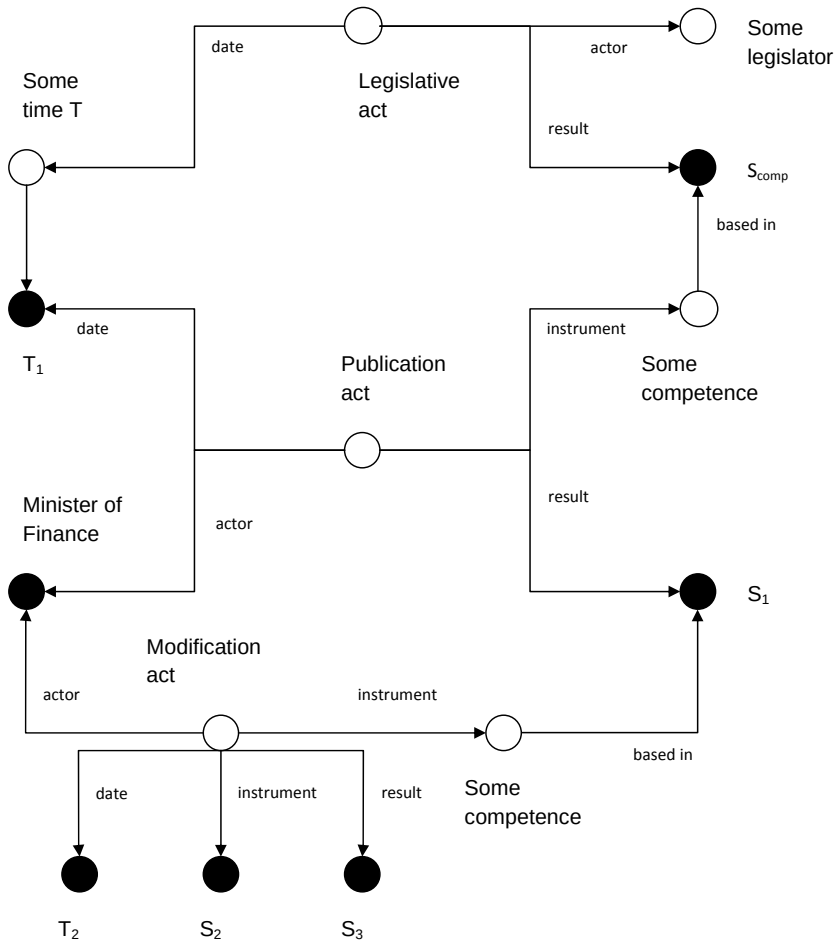

Figure 5.4. Legislative events in MetaLex.

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. [245]):

- 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.
- 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

[20] I.e. it exists only as the object of one's thoughts and communication acts, and not as a physical object.

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 [276], and i.a. [53] in the context of MetaLex.

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;

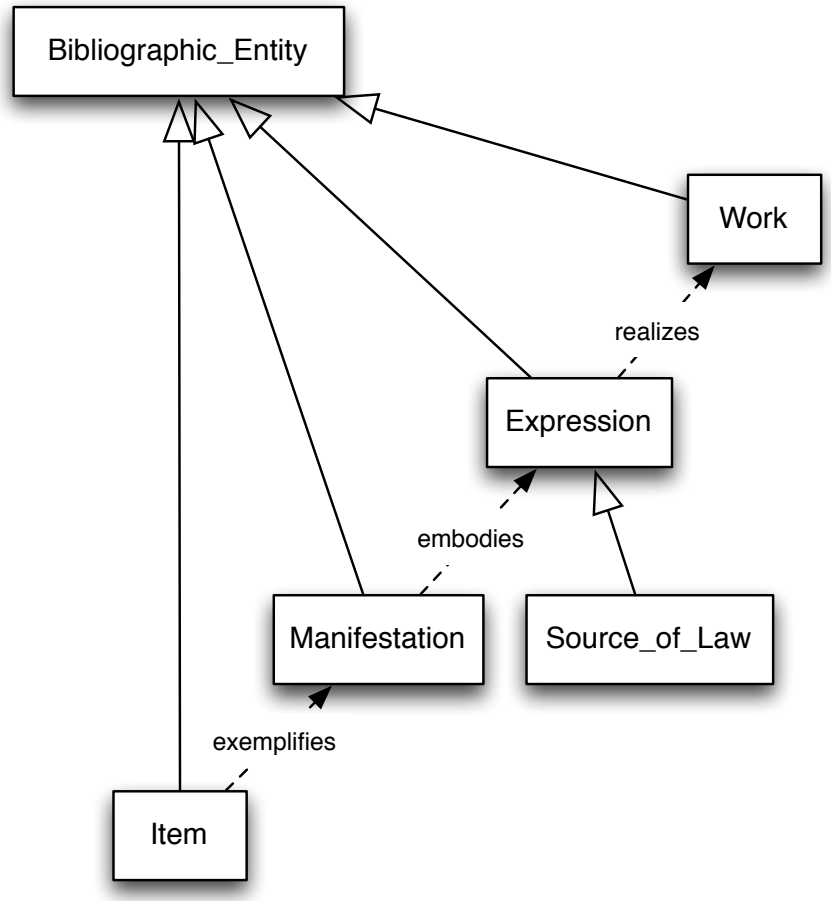

Figure 5.5. A taxonomy of bibliographic entities in MetaLex.

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 MetaLex type `hcontainerType`, and create an element belonging to the substitution group of one of the abstract elements implementing the MetaLex type specified, for instance:

```
<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.

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[21].

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

[21]Consult `http://www.w3.org/TR/xhtml-rdfa-primer/` for RDF/A.

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:

```
<meta id="m1" about="" rel="metalex-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*.

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/`). 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.

If `YYYY-MM-DD` is the date of the MetaLex agreement, and one correctly sets the mime type, the associated OWL schema can be found at:

```
GET http://www.metalex.eu/metalex/YYYY-MM-DD
Accept: application/rdf+xml
```

A description logic syntax rendering is found in an appendix of [39] (see also appendix A of this book). 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

[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/

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 5.3.3). 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 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 happen 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 hierarchical 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, al-

ternatively, 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 a 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 appear to overlap in time *when viewed from any specific vantage point in time*. Note however the possibility of *virtual* and *ex tunc* expressions, which may create a substantial difference between vantage points in time.
- A **version ex tunc** of a bibliographic work is the metalex-owl:result of a METALEX-OWL:FICTIONALEXPRESSIONCREATION, which is a temporal fiction of (metalex-owl:temporalFictionOf) some other event – the constitutive event – that happened *after* the fictional event. The fictional expression creation allows rewriting institutional history, for instance through errata corrige, 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.
- A **consolidation** of a bibliographic source of law 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:LEGISLATIVEMODIFICATION 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. This is often when it enters into force, although the modification may of course be conditional upon some other event.
- 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.
- 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">
```

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 an identifier comformant to 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 `<<ManifURI>>` be the identifier of the manifestation, `<<ExpURI>>` be the identifier of the Expression, `<<WorkURI>>` be the identifier of the work, all declared as in the previous section. Datatyped values `<xsd:date>`, etc. are values conforming to the XML Schema datatype `xsd:date`, etc. The relevant set of metadata for the work is the following:

```
<meta id="w1" about="<<WorkURI>>"
     rel="metalex-owl:resultOf" href="<<WorkCreationEventURI>>"/>
<meta id="w2" about="<<WorkURI>>"
     property="metalex-owl:workClassID" content="<xsd:string>"/>
<meta id="w3" about="<<WorkCreationEventURI>>"
     rel="metalex-owl:date" href="<<CreationEventDateURI>>"/>
<meta id="w4" about="<<WorkCreationEventDateURI>>"
     property="metalex-owl:xsdDate" content="<xsd:date>"/>
<meta id="w5" about="<<WorkCreationEventURI>>"
     rel="metalex-owl:country" href="<<CountryURI>>"/>
<meta id="w6" about="<<CountryURI>>"
     property="metalex-owl:countryCode" content="<xsd:string>"/>
<meta id="w7" about="<<WorkCreationEventURI>>"
     rel="metalex-owl:issue" href="<<IssueURI>>"/>
<meta id="w8" about="IssueURI"
     property="metalex-owl:issueID" content="<xsd:string>"/>
```

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="<<ExpURI>>"
     rel="metalex-owl:resultOf" href="<<ExpCreationEventURI>>"/>
<meta id="e2" about="<<ExpURI>>"
     property="metalex-owl:lang" content="<xsd:language>"/>
<meta id="e3" about="<<ExpCreationEventURI>>"
     rel="metalex-owl:date" href="<<CreationEventDateURI>>"/>
<meta id="e4" about="<<CreationEventDateURI>>"
```

```
     property="metalex-owl:xsdDate" content="<xsd:date>"/>
<meta id="e5" about="<<ExpCreationEventURI>>"
     rel="metalex-owl:issue" href="<<IssueURI>>"/>
<meta id="e6" about="<<IssueURI>>"
     property="metalex-owl:issueID" content="<xsd:string>"/>
```

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:

```
<meta id="e7" about="<<ExpCreationEventURI>>"
     rel="metalex-owl:temporalFictionOf" about="<<RealEventURI>>"/>
<meta id="e7" about="<<RealEventURI>>" rel="metalex-owl:date"
     about="<<RealDateURI>>"/>
<meta id="e8" about="<<RealDateURI>>"
     property="metalex-owl:xsdDate" content="<xsd:date>"/>
```

The temporalFictionOf property is for us simply a kind of constitutedBy. The relevant metadatum for the manifestation is the following:

```
<meta id="m1" about="<<ManifURI>>" property="metalex-owl:fileType"
     content="<xsd:string>"/>
```

This metadatum may be encoded in the XML manifestation as metadata, but it is assumed that extraction of RDF metadata from a standard MetaLex manifestation already provides the information that the file type is `xml`. The OWL schema provides more identifying metadata that **may** be added.

5.3.4. Citation, Inclusion, and Reference

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-owl: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 stand-in 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 *(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.

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.

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 expressions are obviously rare in legal resources.

There is however a difference between the reference *It* referring to theft as defined in the previous sentence and *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 text contains endophoric proform expressions crossing the border of sentences, the sentences obviously cannot be read separately.

The following is an example of a reference metadatum in RDF/A:

```
<meta about="#x" rel="metalex-owl:refersTo"
href="http://gov.tv/act/2004-02-13/concepts#theft">
```

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 use 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:

```
<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 a 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 (for instance 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:

```
<meta id="x" about="#x" rel="metalex-owl:component"
metalex:src="http://gov.tv/tv/act/2004-02-13/2/tv@/table1">
```

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. Summary and Conclusions

At this point we have assembled all ingredients for a representation of the rules and structures of legal institutions in OWL DL knowledge components, and we have established a detailed mapping between the sources of law as documents and the rules and structures of the legal institution. At this point it is therefore possible to give a presentation of a methodology for representing sources of law in OWL, and maintaining isomorphism between these resources over time.

Note however that the normative order is in no way taken into account.

5.4.1. 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 (refer to chapter 3). 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 LKBS 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 LKBS that use the same source of law at a single point.

Single point maintenance is based on the assumption that the LKBS 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 LKBS 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 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 $n_1 : C_1$, 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 *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, certain areas of legislation, including the existing works, have for instance be delegated to a lower level. What was once 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 propositions, follow very similar rules. All propositions identified in a source of law expression are identified by a URI, for instance $n_1 : r_1$, that identifies the proposition as it occurs in that source of law expression. All propositions *created* in a source of law are in the same namespace, and no other propositions are in that namespace. A new URI is assigned only if the proposition 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. If we 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, we can do that as follows:

$$\exists \mathsf{representedBy}.\exists \mathsf{metalex{:}embodies}.\{w\} \sqsubseteq \forall \mathsf{appliesTo}.(C \sqcap \exists \mathsf{applicable}.\{r\})$$

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 propositions are identified by OWL constants: it is absolutely essential to state that each proposition is different from all other known constants in contexts where there is a risk of unification of constants. Both legal propositions 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.

5.4.2. 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. This conclusion is inescapable: also in civil law

jurisdictions case law has the potential to change institutional reality. How and why will be explained in section 6.7.2.

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 propositions 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. propositions 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 book like this one is not only to retain *descriptive fidelity* in representation, but also to increase *intercoder reliability*. The patterns of logical rules in this chapter and in chapter 6 in this sense function as a limited menu of options for representation that increases the chance that knowledge engineers 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. As pointed out in section 3.5.2, one should however be cautious in choosing a core ontology: many generic knowledge components are not suitable for this purpose, since they represent abstract theories that embody a specific *epistemic competence*.

The primary distinction between knowledge sources for the LKBS remains the distinction between descriptive ones, and autoepistemic 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. LKBS-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 [42]), 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 [226, 173, 210]). 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.

5.4.3. Applicability and Knowledge Sources

The isomorphic approach to representing sources of law is simple: take a source of law and represent it. Making an LKBS 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 (cf. section 6.6).

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[24] 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

[24] `http://wetten.overheid.nl`

constitutional court that can annul legislation after it went into effect[25] 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 nonmonotonic reasoning in the institutional ontology, but we need to engineer a special solution for each such fiction pattern. Arbitrary use of fictions is very disagreeable from a knowledge management point of view.

[25]The concept of *judicial review*, as contrasted to *legislative supremacy* or *parliamentary sovereignty*. In the EU many members appear to have such a possibility; Finland, the Netherlands, and the UK at least do not.