Versions and Applicability of Concept Definitions in Legal Ontologies*

Szymon Klarman, Rinke Hoekstra, and Marc Bron

Leibniz Center for Law, University of Amsterdam
PO Box 1030, 1000 BA, Amsterdam, The Netherlands
{sklarman, mbron}@science.uva.nl,hoekstra@uva.nl

Abstract. In many domains concept definitions undergo change on a relatively frequent basis. Especially in law, such changes can have far reaching consequences. Existing ontology versioning techniques often do not consider that old and new definitions may need to co-exist side by side in a knowledge base, or they require non-standard language extensions. In this paper we present a description logic based representation that allows us to model and switch between varying definitions of concepts in a single OWL ontology. We show how this representation can be used to model complex versioning schemes in law.

1 Introduction

A serious challenge for ontology design and maintenance, broadly discussed in the recent literature (e.g. [4, 6, 5]), concerns handling dynamic aspects of represented knowledge. A special case of that problem, which is in the focus of this paper, is how to approach changes that affect the definitions of terms in a legal ontology. Processes of *definitional change* are relatively common in many domains. New scientific theories often lead to a revision of concepts such as the recent change in the definition of 'planet'. This is no different in law, where revised legislation can impose a new interpretation on terms defined and used in older legislative documents. Such changes can clearly have far reaching consequences from both an epistemic and pragmatic perspective. For instance, definitional changes in law can bring about new normative consequences and thus directly affect rights and obligations of citizens.[1]

A new concept definition entails a different classification of domain objects, which can effect in a deep restructuring of the domain's representation. Consequently, reasoning over the revised ontology is unlikely to result in the original

* The work reported here was partially funded by the ESTRELLA IST-2004-027655 project, http://www.estrellaproject.org. This paper reflects the views of the authors.

[1] See e.g. the discussion on the changed definition of "Dependent" in the "Working Families Tax Relief Act" (2004), cf. http://www.milliman.com/expertise/employee-benefits/publications/bib/pdfs/BIB11-23-04-dependency.pdf.

set of conclusions. Especially in the case where a revised ontology is incompatible with the original – the union of the two is unsatisfiable – the impact of a change can be significant.

However, there is a considerable asymmetry in the status granted to outdated definitions in different domains. Scientific knowledge that is replaced by more recent results is simply considered false and, hence, deprived of any substantial value. It is always the current version of an ontology that is applied to provide reliable explanations and predictions in science. Conversely, in law there is a justified interest in keeping track of older versions. A legal case can be properly assessed only according to law that is *applicable* at the time the case takes place. Furthermore, the consideration and analysis of previous cases plays an important role in the judicial process: previous representations of law have to stay available. The revision that a legal ontology is submitted to in face of a definitional change is therefore not that of a plain update but rather of versioning, and hence requires a cautious approach.

In this paper we address the problem of handling changes in the definitions of concepts in (legal) ontologies and propose a description logic based representation that provides a simple, but in many respects convenient ontology versioning technique. The representation consists of a set of TBox and ABox axiom schemes, which allow for reasoning about a domain within a declared time interval, strictly according to concept definitions valid during that interval. In the next section we review related work and formulate basic requirements for an ontology versioning formalism. We then discuss an example model inspired by the Nomic game.[2] We close the paper with conclusions and discussion of the proposed solution.

2 Ontology Versioning

Representing change and handling different variants of ontologies is the subject of research on ontology *evolution* and ontology *versioning*. Ontology evolution concerns changing an existing ontology while maintaining consistency. For example, the removal of a concept from an ontology requires the deletion of all references to this concept.

Ontology versioning, on the other hand, takes a copy-first strategy where changes are effected in a new, copied, version of an ontology. This usually involves the recording of changes and relations between different versions: e.g. it is useful to know if a new version is backward compatible with the previous version. This can be essential information for proper functioning of ontology-driven applications. A system that implements such a versioning system is e.g. SHOE [6]. Efforts at combining both strategies in a single system are described in [9].

However, these approaches are very much focused on the changes affected during the development of an ontology, and do not consider the case where multiple versions may independently hold. [4] presents an approach where multiple

[2] See `http://www.nomic.net/`, [10]. Nomic was implemented in ESTRELLA as an OWL-DL ontology by Abdallah El-Ali and Xingrui Ji, along with the authors of this paper.

versions can be represented in a single ontology file using *time stamps*. In this approach, an ontology is described by a graph where each node (concept) and edge (property) is annotated with a time stamp that indicates its validity. Such a system can be implemented in three ways. First, using a *meta ontology* that stores the different time stamped versions of an ontology and represents the structural and temporal relations between those versions. Second, by an extension of the knowledge representation language that can express time and change. And finally, by exploiting the standard techniques provided by a representation language to time stamp concepts, e.g. using the versionInfo and isDefinedBy annotations in OWL.[3]

These techniques have drawbacks when implemented in OWL. For instance, if the time dimension is not considered by a reasoner, its conclusions may well be incorrect: it should only infer conclusions that hold at a 'current' time. Another drawback is that OWL does not guarantee correct interpretation of temporal constraints, e.g. a relation can only exist between two concepts that are both valid at the same time. Furthermore, a time stamping approach can cause a snowball effect in the updates it requires. If, for example, a concept A has a relation to concept B and which is updated to a new definition B', then A has to be updated to A', so that the relation of A' points to B'. Every concept with a relation to the original A then needs to be revised as well, and so on and so forth.

Alternatively, one can handle versions of a definition by using rules that classify only relevant individuals at particular points in time. In a rule language like SWRL, the variants of e.g. "block" can be expressed as:

Rule 1: $\mathsf{square}(?x) \wedge \mathsf{wood}(?x) \wedge \mathsf{time}(t_1) \wedge \mathsf{exists\_at}(?x, t_1) \Rightarrow \mathsf{block}(?x)$.
Rule 2: $\mathsf{rectangle}(?x) \wedge \mathsf{metal}(?x) \wedge \mathsf{time}(t_2) \wedge \mathsf{exists\_at}(?x, t_2) \Rightarrow \mathsf{block}(?x)$.

Although this approach would certainly tackle some of the problems discussed above, applying a rule formalism in ontology modelling has its drawbacks. A model can not only become undecidable, but the classification 'by rules' may well conflict with DL-based inferencing in unexpected ways.

2.1 Versioning and Applicability in Law

For the purposes of legal reasoning, an ontology is a particular interpretation of the world that imposes an ordering on the actual entities of that world. Multiple different orderings can hold at the same time, and may be compatible with respect to the world, but not with respect to each other. On the other hand, the conceptual framework of law is fairly stable and based largely in common sense notions: it is generally an extension and refinement of common sense [7].

The status of an officially sanctioned legal text – and consequently its representation – such as a piece of legislation or court decision (jurisprudence) is primarily determined by its relation to other texts. The body of legal texts published within a jurisdiction usually is not consistent as such, and is governed by

[3] Note that OWL 1.0 does not allow the annotation of axioms, where OWL 1.1 does.

three meta legal rules for conflict resolution: *lex superior*, *lex specialis*, and *lex posterior*.

The application of lex posterior, i.e. newer norms overrule old ones, to a set of norms requires a clear insight in the temporal relations between these norms. We can distinguish three types of time stamps used in legislation: *version management*, determining the validity of a document for reference, *drafting*, relating to the procedures that have to be followed by the issuer, and *application to cases*, determining the relation of a norm to events in the world, e.g. for retroactive, immediate or delayed application. Especially the last type makes law peculiar, as activity and applicability can diverge, e.g. a currently active norm may not hold for a current case. [3] proposes a simple event-based model to describe the life-cycle of legal texts in the context of the MetaLex XML standard for legislative sources [2].[4] As regards version management, the status of a particular legal text is successively *fixed* after drafting, *knowable* (published), *active* and *repealed*. In MetaLex these are reflected by four time-related attributes: *date-publication*, *date-enacted*, *date-repealed*, *date-effective*, see Table 1. A text is *active* if the current date lies between the enactment and repeal date (or no repeal has taken place); a text is *applicable* to a case if the date of the case is between the efficacy and repeal date of that text. A formal representation of norms or regulations needs to take into account how these intervals interact with concept definitions in a legal ontology.

We formulate the following requirements for versioning in legal ontologies. A new version of a concept definition should have *minimal impact* on other definitions. The versioning mechanism should be incremental, and prevent a snowball effect of updates. Furthermore, multiple versions of a concept should be able to *co-exist* within one ontology without causing inconsistencies or other reciprocal effects. One should be able to *switch* between different versions of a concept definition, while maintaining the ability to reason over version-independent (common sense) concepts. The mechanism should be *flexible* enough to account for the distinction between validity and applicability. And finally, it should 'fit' within a single *general purpose* knowledge representation formalism to allow for sharing and reuse of ontologies.

3 Representing Definitional Change

In the following, we propose a solution that allows to capture the core aspects of concept versioning in a single OWL-DL ontology, while preserving the possibility of using a standard reasoner for obtaining classifications that respect temporal constraints imposed on the representation. To this end, we require a description logic language as expressive as $\mathcal{SHOIN}$, which is directly supported by OWL-DL. We refer to an arbitrary definitorial knowledge base $\mathcal{K}(\mathcal{T}, \mathcal{A})$ [1], consisting of a set of concept names N_C, role names N_R and individual names N_I, with a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. A *dynamic concept* C, i.e. a concept whose meaning changes

[4] See `http://www.metalex.eu` and `http://legacy.metalex.eu`

Table 1. Activity and applicability in legal texts, where $t_0 \leq t_1 < t_2 < t_3 < t_4$

publication	**enacted**	**repealed**	**effective**	***active***	***applicable***
t_1	t_2	t_4	t_2	$[t_2, t_4]$	$[t_2, t_4]$ (immediate)
t_1	t_2	t_4	t_0	$[t_2, t_4]$	$[t_0, t_4]$ (retroactive)
t_1	t_2	t_4	t_3	$[t_2, t_4]$	$[t_3, t_4]$ (delayed)

over time, is represented in $\mathcal{K}$ by means of its all definitional variants. Further, we provide a simple mechanism for selecting a valid time $i \in \{1, \ldots, n\}$, such that the respective version $\mathcal{K}^i$ of $\mathcal{K}$ has the same base interpretation as the original ontology, but possibly a different extension $\mathcal{J}_i$, yielding $C^{\mathcal{J}_i} = C_i^{\mathcal{J}_i}$, where C_i represents the proper variant of the concept C in time i.

The representation has a three-layered structure, consisting of a simple temporal framework, a collection of time stamps on individuals and a set of temporal restrictions on concepts. The valid time selection is handled on the TBox level, and allows alternative definitions of a designated concept.

Temporal Framework The representation of time is based on interval algebra and defined as a tuple $\langle T, \preceq \rangle$, which constitutes a discrete, finite time axis, whose atoms are undecomposable time intervals [11], and where $\preceq$ is a total order on T. We represent this in $\mathcal{K}$ by means of a primitive concept $\mathsf{TimeInterval} \in N_C$, such that $\mathsf{TimeInterval}^{\mathcal{I}} \neq \emptyset$, and a role $\mathsf{beforeEq} \in N_R$. The ABox has to be equipped with a set of assertions guaranteeing that $\mathsf{beforeEq}$ is a total ordering on $\mathsf{TimeInterval}^{\mathcal{I}}$, i.e. it is antisymmetric, transitive and complete. For more flexibility we introduce an inverse relation: $\mathsf{afterEq} = \mathsf{beforeEq}^{-}$.

These constructs suffice as a frame of reference for representing temporal aspects of definitional changes, including the dynamics of a domain on the level of individuals.

Valid Time In order to control the focus of a knowledge base we introduce a mechanism for selecting the current 'valid' time. This is done by pointing to a particular interval, with respect to which relevant concepts and individuals are activated. Although there are other ways to achieve this, we propose a simple method based on a single concept definition using a nominal. For some i, such that $\mathsf{interval}_i^{\mathcal{I}} \in \mathsf{TimeInterval}^{\mathcal{I}}$, let $\mathsf{CurrentInterval} \in N_C$ be defined as:

$$\mathsf{CurrentInterval} \equiv \{\mathsf{interval}_i\}$$

The rest of the formalism relies on the interpretation of the $\mathsf{CurrentInterval}$. This has two apparent advantages. First, it enables a very simple way for switching between versions. Second, it does not affect the base interpretation of the knowledge base, which may potentially simplify certain reasoning tasks. If two knowledge bases $\mathcal{K}(\mathcal{T}, \mathcal{A})$ and $\mathcal{K}'(\mathcal{T}', \mathcal{A})$ differ from each other only with respect to the choice of the current interval, none of the primitive concepts obtains a different interpretation: a switch entails changes only for extensions of the base interpretation.

Stamps and Temporal Restrictions Individuals are indexed in time using the relations $\mathsf{from}, \mathsf{to} \in N_r$. A time stamp links an individual to intervals of time, marking exactly two limits of its existence in a domain of application [11]:

$$\begin{aligned}\mathsf{TimeStamped} \equiv\ & (\exists \mathsf{from}.\mathsf{TimeInterval} \sqcap \geq 1\ \mathsf{from} \sqcap \leq 1\ \mathsf{from}) \\ & \sqcap\ (\exists \mathsf{to}.\mathsf{TimeInterval} \sqcap \geq 1\ \mathsf{to} \sqcap \leq 1\ \mathsf{to})\end{aligned}$$

In other words, for any individual a, such that $\mathsf{TimeStamped}(a)$, the ABox must contain two assertions: $\mathsf{from}(a, \mathsf{interval}_i)$ and $\mathsf{to}(a, \mathsf{interval}_j)$, meaning that a came into existence in $\mathsf{interval}_i$ and ceased to exist in $\mathsf{interval}_j$. Naturally, it should be assured that $1 \leq i \leq j \leq n$.

In order to enable an appropriate interaction between time-stamped individuals and selected concepts, we will use the following axiom scheme:

$$\begin{aligned}\mathsf{GeneralTRestriction} \equiv\ & \exists \mathsf{from}.(\exists \mathsf{beforeEq}.(\bigsqcap_{1 \leq i \leq m} \mathsf{TConstraint}_i)) \\ & \sqcap\ \exists \mathsf{to}.(\exists \mathsf{afterEq}.(\bigsqcap_{1 \leq i \leq m} \mathsf{TConstraint}_i))\end{aligned}$$

$\mathsf{GeneralTRestriction}$ plays a central role in the proposed framework, serving as an additional restriction to be imposed on all concept definitions that require a temporal qualification. Every $\mathsf{TConstraint}_i$, represented as a subset of time intervals, is meant to express a different type of temporal constraint that is essential for assessing whether an individual falls under a particular definition or not. A pivotal feature of $\mathsf{GeneralTRestriction}$ is that its satisfiability is conditional on the relationships between the involved time constraints. Observe, that whenever $(\bigsqcap_{1 \leq i \leq m} \mathsf{TConstraint}_i)^{\mathcal{I}} = \emptyset$, i.e. when the constraints do not have a single member common to all of them, then $\mathsf{GeneralTRestriction}$ is necessarily unsatisfiable, and consequently, so are the concepts restricted by it.

Recall that our goal is to confine classification only to individuals and concepts relevant at a specific time. More specifically, it has to be assured, that given the current interval the definition of an applicable concept can be satisfied only by the individuals that exist in that interval, and conversely, that none of the definitions of currently inapplicable concepts are satisfiable. We obtain this feature by incorporating two time constraints in the $\mathsf{GeneralTRestriction}$ of every (time dependent) concept definition:

1. $\mathsf{TConstraint}_1 \equiv \mathsf{CurrentInterval}$
2. $\mathsf{TConstraint}_2 \equiv\ \exists \mathsf{afterEq}.\{\mathsf{interval}_i\}\ \sqcap\ \exists \mathsf{beforeEq}.\{\mathsf{interval}_j\}$

where $\mathsf{interval}_i$ and $\mathsf{interval}_j$, such that $1 \leq i \leq j \leq n$, indicate the limits of the concept's applicability.[5] The $\mathsf{GeneralTRestriction}$ ensures two possible outcomes of classification. If the current interval is contained within the period during

[5] Notice, that given a total ordering of time intervals, an interpretation function maps $\mathsf{TConstraint}_2$ to the subset of exactly those intervals that are placed between $\mathsf{interval}_i$ and $\mathsf{interval}_j$ (including the two) on the time axis.

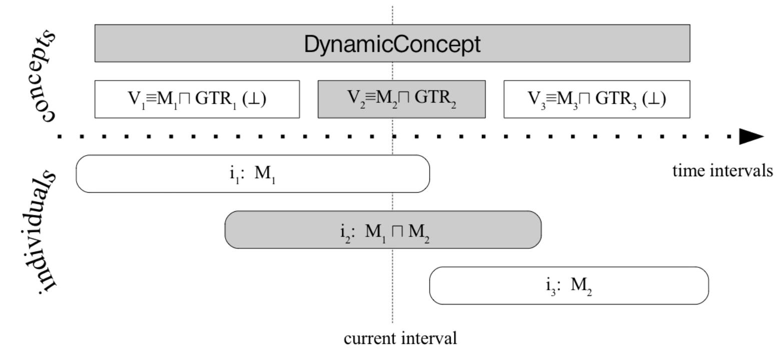

Fig. 1. Relation between a generic concept, its variants and domain objects.

which a concept is applicable ($\mathsf{TConstraint}_1^{\mathcal{I}} \subseteq \mathsf{TConstraint}_2^{\mathcal{I}}$), then the potential interpretation of the concept is limited exactly to those time-stamped individuals that currently exist. In the opposite case, the concept is unsatisfiable with respect to the TBox, as there is no individual that can satisfy the $\mathsf{GeneralTRestriction}$.

Definitional Variants We can now use the introduced constructs to describe definitional change. Let $\mathsf{DynamicConcept} \in N_C$ be a concept whose definition changes over time and $\{\mathsf{Variant}_1, \ldots, \mathsf{Variant}_m\} \subseteq N_C$ its m consecutive variants, where index $k \in \{1, \ldots, m\}$ uniquely identifies each of them. We further assume (though it is not a general requirement), that the variants exclusively and exhaustively cover the time axis. The definition of $\mathsf{DynamicConcept}$ is simply stated as the union of all the variants:

$$\mathsf{DynamicConcept} \equiv \mathsf{Variant}_1 \ \sqcup \ \ldots \ \sqcup \ \mathsf{Variant}_m$$

Formally, we define every variant as an intersection of its proper meaning, expressed in terms of some terminological restrictions and its $\mathsf{GeneralTRestriction}$:

$$\mathsf{Variant}_k \equiv \mathsf{Meaning}_k \ \sqcap \ \mathsf{GeneralTRestriction}_k$$

This accomplishes the representation. Observe that in each point of time there is exactly one valid variant. Therefore, if $\mathsf{interval}_i$ represents the current interval and $\mathsf{Variant}_k$ is the currently applicable variant, then interpretation of $\mathsf{DynamicConcept}$ is determined by the following equality:

$$\mathsf{DynamicConcept}^{\mathcal{J}_i} = \mathsf{Variant}_k^{\mathcal{J}_i}$$

which naturally entails that $\mathsf{DynamicConcept} \equiv \mathsf{Variant}_k$.

Concluding, independently of the choice of the valid time, $\mathsf{DynamicConcept}$ always denotes these and only these individuals that exist during the intervals specified by the general temporal restriction and fall under the currently valid meaning of $\mathsf{DynamicConcept}$ (see Fig. 1).

3.1 Activity, Applicability and Validity

Finally, we briefly outline how the representation can be usefully applied to express the legal notions of active and applicable concepts, and provide an example. Informally, a *legal* dynamic concept can be defined as a concept where all general temporal restrictions of the variant are scoped in at least three ways: by the CurrentInterval, some applicability period AppInterval of a concept definition, and the activity period ActInterval of the relevant legal text:

$$\begin{aligned}\mathsf{AppInterval} &\equiv \exists \mathsf{afterEq}.\{\mathsf{interval}_i\} \sqcap \exists \mathsf{beforeEq}.\{\mathsf{interval}_j\}\\ \mathsf{ActInterval} &\equiv \exists \mathsf{afterEq}.(\exists \mathsf{from}^-.\{\mathsf{norm}\}) \sqcap \exists \mathsf{beforeEq}.(\exists \mathsf{to}^-.\{\mathsf{norm}\})\end{aligned}$$

where $1 \leq i \leq j \leq n$, and norm is the time-stamped individual representing the respective legal text. A variant of a legal concept is valid only if the intersection of the three constraints is nonempty.

This kind of perspective on the life-cycle and validity of legal definitions has been adopted in the model of the Nomic game [10]. The game is an illustration of the paradox of self-amendment — a phenomenon inherent to functioning of legal systems, where in principle, every law is potentially a subject to change, justified by other laws effective in the system. Accordingly, every move in the game is a change of one of its rules, some of which constitute definitions of domain concepts. The following example shows the process of definitional change that takes place in a hypothetical Nomic scenario.

Example Consider a short, 10-turn long game with 3 participants, entering and leaving the game at different stages. The participants are characterised by the properties:[6]

$$\begin{aligned}\mathsf{John} &: \mathsf{from}(\mathsf{John}, \mathsf{turn}_1), \mathsf{to}(\mathsf{John}, \mathsf{turn}_{10}), \mathsf{Male}(\mathsf{John}), \mathsf{Player}(\mathsf{John})\\ \mathsf{Ann} &: \mathsf{from}(\mathsf{Ann}, \mathsf{turn}_3), \mathsf{to}(\mathsf{Ann}, \mathsf{turn}_{10}), \mathsf{Female}(\mathsf{Ann}), \mathsf{Player}(\mathsf{Ann})\\ \mathsf{Mary} &: \mathsf{from}(\mathsf{Mary}, \mathsf{turn}_1), \mathsf{to}(\mathsf{Mary}, \mathsf{turn}_8), \mathsf{Female}(\mathsf{Mary}), \mathsf{Player}(\mathsf{Mary})\end{aligned}$$

Initially, the set of rules contains the following regulation, defining the concept of Voter.

rule_303: Until turn 4 all players shall be eligible voters. After that, until the end of the game, only male players shall be considered as such.

[6] Note that whereas DL typically adopts the Unique Name Assumption (UNA), OWL DL does not. Intervals need therefore be distinguished using the owl:allDifferent construct.

Clearly, rule_303 constitutes two definitional variants of the concept and explicitly states their applicability intervals. At the end of turn 6, however, the rule is repealed and replaced by another one, taking effect immediately from turn 7, which brings about a revision of the meaning of Voter:

rule_304: All and only female players shall be eligible voters.

We describe the setting with the following set of axioms, which express definitional variants of Voter and their applicability periods, enactment and repeal time of rules, along with their corresponding activity periods:

$$\begin{aligned}
\mathsf{Voter}_1 &\equiv \mathsf{Player} \sqcap \mathsf{GeneralTRestriction}_1\\
\mathsf{Voter}_2 &\equiv \mathsf{Player} \sqcap \mathsf{Male} \sqcap \mathsf{GeneralTRestriction}_2\\
\mathsf{Voter}_3 &\equiv \mathsf{Player} \sqcap \mathsf{Female} \sqcap \mathsf{GeneralTRestriction}_3\\
\mathsf{AppInterval}_1 &\equiv \exists \mathsf{afterEq}.\{\mathsf{turn}_1\} \sqcap \exists \mathsf{beforeEq}.\{\mathsf{turn}_4\}\\
\mathsf{AppInterval}_2 &\equiv \exists \mathsf{afterEq}.\{\mathsf{turn}_5\} \sqcap \exists \mathsf{beforeEq}.\{\mathsf{turn}_{10}\}\\
\mathsf{AppInterval}_3 &\equiv \exists \mathsf{afterEq}.\{\mathsf{turn}_7\} \ \sqcap \exists \mathsf{beforeEq}.\{\mathsf{turn}_{10}\}\\
\mathsf{rule\_303} &: \mathsf{from}(\mathsf{rule\_303}, \mathsf{turn}_1), \mathsf{to}(\mathsf{rule\_303}, \mathsf{turn}_6)\\
\mathsf{rule\_304} &: \mathsf{from}(\mathsf{rule\_304}, \mathsf{turn}_7), \mathsf{to}(\mathsf{rule\_304}, \mathsf{turn}_{10})\\
\mathsf{ActInterval}_{303} &\equiv \exists \mathsf{afterEq}.(\exists \mathsf{from}^-.\{\mathsf{rule\_303}\}) \ \sqcap \ \exists \mathsf{beforeEq}.(\exists \mathsf{to}^-.\{\mathsf{rule\_303}\})\\
&\equiv \exists \mathsf{afterEq}.\{\mathsf{turn}_1\} \ \sqcap \ \exists \mathsf{beforeEq}.\{\mathsf{turn}_6\}\\
\mathsf{ActInterval}_{304} &\equiv \exists \mathsf{afterEq}.(\exists \mathsf{from}^-.\{\mathsf{rule\_304}\}) \ \sqcap \ \exists \mathsf{beforeEq}.(\exists \mathsf{to}^-.\{\mathsf{rule\_304}\})\\
&\equiv \exists \mathsf{afterEq}.\{\mathsf{turn}_7\} \ \sqcap \ \exists \mathsf{beforeEq}.\{\mathsf{turn}_{10}\}
\end{aligned}$$

Given all the conditions we constrain the definitional variants via the respective general temporal restrictions with respect to the time intervals listed below, and finally, posit the generic definition of the concept Voter:

$$\begin{aligned}
\mathsf{GeneralTRestriction}_1 &: \mathsf{CurrentInterval},\ \mathsf{AppInterval}_1,\ \mathsf{ActInterval}_{303}\\
\mathsf{GeneralTRestriction}_2 &: \mathsf{CurrentInterval},\ \mathsf{AppInterval}_2,\ \mathsf{ActInterval}_{303}\\
\mathsf{GeneralTRestriction}_3 &: \mathsf{CurrentInterval},\ \mathsf{AppInterval}_3,\ \mathsf{ActInterval}_{304}\\
\mathsf{Voter} &\equiv \mathsf{Voter}_1 \sqcup \mathsf{Voter}_2 \sqcup \mathsf{Voter}_3
\end{aligned}$$

Observe, that depending on the choice of the current interval the following interpretations of Voter, complying to the imposed limits of validity, will be inferred:

CurrentInterval	Active rule	Valid variant	$\mathsf{Voter}^{\mathcal{I}}$
$\mathsf{interval}_1$	rule_{303}	Voter_1	$\{\mathsf{John}, \mathsf{Mary}\}^{\mathcal{I}}$
$\mathsf{interval}_3$	rule_{303}	Voter_1	$\{\mathsf{John}, \mathsf{Ann}, \mathsf{Mary}\}^{\mathcal{I}}$
$\mathsf{interval}_5$	rule_{303}	Voter_2	$\{\mathsf{John}\}^{\mathcal{I}}$
$\mathsf{interval}_8$	rule_{304}	Voter_3	$\{\mathsf{Ann}, \mathsf{Mary}\}^{\mathcal{I}}$
$\mathsf{interval}_{10}$	rule_{304}	Voter_3	$\{\mathsf{Ann}\}^{\mathcal{I}}$

Regardless of the varying interpretation, all domain concepts that refer to Voter remain intact.

4 Conclusions and Discussion

In this paper we presented a representation that allows the expression of definitional changes in ontologies, motivated predominantly by requirements from the legal domain. It can be expressed as a set of TBox and ABox axioms in $\mathcal{SHOIN}$ and is thus easily implementable in OWL-DL. The representation allows the correct classification of individuals in an ontology using different variants of concept definitions, applicable within specified scopes of time intervals. Furthermore, the inherent representation of time provides a framework for capturing the basic dynamics of the domain on the level of individuals, and for including temporal aspects directly into the definitions of concepts. We have tested the representation by modelling an instance of the Nomic game.

The solution satisfies the basic requirements with respect to ontology versioning formalisms and exhibits several interesting features. It is supported directly by standard reasoning tools such as Pellet: no additional formalism nor external ontology management system is necessary. And furthermore, a definitional change is implemented incrementally, only by introducing new concepts to the current ontology. No other elements, even those referring to a generic concept, are altered. This property seems especially valuable for reuse and maintenance of legal ontologies, taking into account the entrenchment of concepts in models of legislation.

Although the scheme is very space efficient, this is achieved at the expense of reasoner performance when switching between versions. Whether this trade-off is well balanced depends on the intended application. Presumably, three factors should play a role here: the size of an ontology, the scope and frequency of introduced changes and the frequency of version switching requests.

Related to this, we investigated the use of an individual (instead of a concept) to represent the current interval. Because OWL does not adopt the unique name assumption, this interval can be equated with any interval by using the owl:sameAs construct: the choice of the current interval is held outside of the TBox. This representation seems to perform slightly better than the one based on the use of nominals, and significantly better than an alternative using pseudo-nominals.[7] It may also be useful to introduce a last_interval individual, equated to the last interval on the time axis to serve as the right limit of time stamps and scopes of all those individuals and concepts which have not in fact become outdated. Extending the time axis in an ontology can be done by merely changing the referent of last_interval.

In future research we aim to investigate in more detail the issues concerning efficiency of reasoning over the representation, and moreover, to explore the

[7] Note, however, that so far we have not tested the approaches on large scale examples. On a minimal example, realisation in Pellet for nominal and pseudo nominal solutions takes respectively 2 and 4 times longer than the solution using individuals.

possibility of formulating and answering time-related queries that move beyond mere selection of the currently valid version of an ontology (cf. [8]). Extending the formalism to a different dimension would be another interesting direction. As the applicability of law is not only relative to time but also to location, legal concepts could be analogously restricted with spatial constraints. To this end the choice of a suitable representation of space would have to be carefully considered since the analogy between temporal and spatial aspects of legal applicability is likely to fall short in many ways.

Bibliography

[1] F. Baader and W. Nutt. *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter Basic Description Logic, pages 47–100. Cambridge University Press, 2003.

[2] A. Boer, R. Hoekstra, R. Winkels, T. van Engers, and F. Willaert. ^{META}*lex*: Legislation in XML. In T. Bench-Capon, Aspassia Daskalopulu, and R.G.F. Winkels, editors, *Legal Knowledge and Information Systems. Jurix 2002: The th Annual Conference*, pages 1–10, Amsterdam, 2002. IOS Press.

[3] A. Boer, R. Winkels, T. van Engers, and E. de Maat. A content management system based on an event-based model of version management information in legislation. In T. Gordon, editor, *Legal Knowledge and Information Systems. Jurix 2004: The 17th Annual Conference.*, pages 19–28, Amsterdam, 2004. IOS Press.

[4] J. Eder and C. Koncilia. Modelling changes in ontologies. In *Proceedings of the On The Move - Federated Conferences*, 2004.

[5] P. Haase, Y.Sure, and D.Vrandecic. Ontology management and evolution: Survey, methods and prototype. SEKT Formal Deliverable D3.1.1, AIFB, University of Karlsruhe, 2004.

[6] J. Heflin and J. Hendler. Dynamic ontologies on the web. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 443–449, 2000.

[7] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer. The LKIF Core ontology of basic legal concepts. In Pompeu Casanovas, Maria Angela Biasiotti, Enrico Francesconi, and Maria Teresa Sagri, editors, *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, June 2007.

[8] Natalya Keberle, Yuriy Litvinenko, Yuriy Gordeyev, and Vadim Ermolayev. Ontology evolution analysis with owl-met. In Giorgos Flouris and Mathieu d'Aquin, editors, *International Workshop on Ontology Dynamics (IWOD 2007)*, 2007.

[9] M. Klein and N.F. Noy. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440, 2004.

[10] Peter Suber. *The Paradox of Self-Amendment.* Peter Lang Publishing, 1990.

[11] G.P. Zarri. Representation of temporal knowledge in events: The formalism, and its potential for legal narratives. *Information & Communications Technology Law*, 7:213–241, 1998.