Legislation in the knowledge base paradigm: interactive decision enactment for registration duties

Marjolein Deryck*, Jo Devriendt†, Simon Marynissen‡ and Joost Vennekens*
*Department of Computer Science, KU Leuven Campus De Nayer, Sint-Katelijne-Waver, Belgium
Email: firstname.lastname@kuleuven.be
†KTH Royal Institute of Technology Stockholm, Sweden
Email: jhmde@kth.se
‡Department of Computer Science, KU Leuven, Leuven, Belgium
Email: simon.marynissen@kuleuven.be

Abstract—**Recently, a prototype for an interactive decision enactment system for notaries was developed. This prototype follows the Knowledge Base Paradigm (KBP): it consists of purely declarative domain knowledge, to which various logical inference methods can be applied. This paper extends that work in two ways. First, we experimentally validate the claim that the KBP leads to highly maintainable software. Second, we extend the number of additional logical inferences, which allow us to address a number of usability concerns. This provides further evidence for the claim that the KBP is indeed a viable method of developing interactive software systems. The resulting decision enactment prototype is a fully generic system, that can be applied to other domains with minimal effort.**

I. INTRODUCTION

Legal applications have often been used as test cases for knowledge-based AI systems. This was the case for traditional rule-based expert systems [18], but also for more recent systems such as, DROOLS [21]. In recent work [13], an interactive decision enactment system for notaries was developed according to the Knowledge-Base Paradigm (KBP) [11].

The KBP advocates a strict separation between declarative domain knowledge; and the way in which this knowledge is used to perform certain tasks. This is in contrast to typical rule-based expert systems in which knowledge is formalized specifically with a forward-chaining inference algorithm in mind, or Prolog-based systems in which knowledge is formalized with a backwards query-answering algorithm in mind. In the KBP, domain knowledge is formalized as a purely declarative *knowledge base*, which is not tied to a specific *inference* method, allowing the same knowledge to be used by different algorithms in order to achieve different goals.

As claimed by [11], this paradigm has two main advantages. First, the knowledge base is easier to maintain, because it can be considered in isolation from the inference methods. Second, the knowledge base is easier to reuse for other inference tasks, since it is not tied to any specific inference method anyway.

In [13], a decision enactment system that supports Belgian notaries in their handling of real estate sales was developed according to this approach. The Belgian legislation on registration duties that need to be paid when purchasing real estate is quite complex: there exist multiple registration types with different rates, and legislation from the country's three regions may apply in addition to federal regulations. The tool in [13] was developed together with notary Luc Van Pelt. Like other Belgian notaries, this office prides itself on its customer-friendly and confidential service. They are therefore looking for a system that provides support while interviewing clients, without interrupting the natural flow of the conversation. The system should therefore be able to accept relevant information in any order and to provide useful feedback on each piece of information.

In this paper, we further develop and analyze the prototype that was developed in [13]. Our work focuses in particular on validating the two main advantages of the KBP that were mentioned above. First, we update the prototype to cope with a recent change to Belgian legislation. This change was significant enough to warrant substantial coverage by all major Belgian news outlets and therefore presents an interesting and representative test case for the maintainability of the knowledge base. Second, during its evaluation of the prototype, the notary office identified several additional desirable features that were not initially thought of. We were able to add these features to the prototype in a generic, domain-independent way. This supports the claim that the functionality that users desire from an interactive system can indeed be implemented by applying the appropriate inference methods to a purely declarative knowledge base, even when this functionality was not originally foreseen at the time when the knowledge base was constructed.

This work results in a completely generic framework, similar to, but more powerful than that of [8]. This generic framework can be applied to create powerful interactive decision enactment systems for other domains with minimal effort. It is developed using the IDP KBP system [9], which allows it to benefit from both this system's expressive knowledge representation language FO($\cdot$), as well as from its efficient inference algorithms.

This paper is structured as follows. Section II elaborates on the background of the case. Section III introduces the KBP and the IDP system. In Section IV the main characteristics of the original prototype are brought to mind. Sections V and VI put the legal amendments and the interface into practice. Section VII presents related work, followed by a discussion and conclusion in Section VIII.

II. CASE STUDY

In Belgium, when a party wants to conclude a transaction on the real estate market, a notary is required to affirm the process, providing legal certificates for the requested transaction. This registration gives rise to the payment of registration duties, which depend on the region in which the estate is situated. The standard tax rate can be reduced for certain registration types, which leads to a range of possible tax rates with their associated conditions.[1]

Prior to June 1st 2018, most family houses would either be subject to the standard duty of 10%, or to a reduced duty of 5% for "modest houses". Whether a house would classify as modest depended mainly on its *kadastral income (KI)*, a value which represents its theoretical rental value. This KI was then compared to a threshold, that depended on the buyer's number of children. In addition, there also existed an independent and elaborate system of *abattements* (reductions on the taxable base of the house).

To remedy the complexity of this system, new and simplified legislation came into force. The concepts of *KI* and *abattement* were abandoned. To determine if a house should be considered "modest", its actual selling price is now used instead of the fictitious *KI*. These reforms of the registration duties represent a profound change: of the original 42 law articles of chapter 9 concerning the registration law, the decree of 18 May 2018 abolishes 4 articles, modifies 9 and adds 5 of them.

Our prototype of [13] employs a knowledge base for the original legislation. Its functionality focuses on two requirements:

- *Completeness and correctness:* The application should ensure that all possible discounts are taken into account, and only rule out or apply discounts when warranted by the information provided by the notary.
- *Usability:* In meetings between notaries and their clients, the technology should not disturb the confidential atmosphere. A lot of typing and searching for the correct button is out of the question.

After evaluating this prototype, the notary office came up with these additional requirements:

- *Traceability* The decision outcomes calculated by the application should be easy to check and explain, in order to increase clients' confidence in the application.
- *Efficient information gathering* Only questions relevant to possible discounts should be asked. E.g., as soon as it is clear that one of the discounts cannot be used, questions related to this discount become irrelevant and should no longer be asked.

III. THE KNOWLEDGE BASE PARADIGM AND IDP

We use the IDP knowledge base system, which employs FO($\cdot$) as a formal knowledge base specification language [9]. The core of FO($\cdot$) is typed first-order logic, extended with inductive definitions, aggregates and arithmetic [9]. In this section, we only recall a small propositional fragment of the language, which suffices to explain the notary application.

In our restricted fragment, we assume a set of constants c which each have an associated domain $dom(c)$ of possible values $\{v_1, \dots, v_n\}$. As a running example, we use the selection of an appropriate rate for the calculation of the registration fee. Here, we have a constant $ApplicableRate$ whose domain consists of integers $\{1, 7, 10\}$, and a constant $RegistrationType$ whose domain consists of $\{Social, Modest, Other\}$.

A *partial interpretation* $\mathcal{I}$ assigns to each constant c a non-empty subset $c^{\mathcal{I}}$ of values from its domain. A *total interpretation* I assigns to each constant c a single value c^I from its domain. Partial interpretations can be ordered according to their precision: $\mathcal{I} \leq_p \mathcal{I}'$ if for each c, $c^{\mathcal{I}} \supseteq c^{\mathcal{I}'}$. Total interpretations correspond to precision-maximal partial interpretations. We say that a total I is an *expansion* of $\mathcal{I}$ if for each c, $c^I \in c^{\mathcal{I}}$. As an example, the total interpretation

$$I_{ex} = \{ApplicableRate = 1, RegistrationType = Social\}$$

is an expansion of the partial

$$\begin{aligned}\mathcal{I}_{ex} = \{&ApplicableRate \in \{1, 7\},\\ &RegistrationType \in \{Social, Modest, Other\}\}\end{aligned}$$

and also of the least precise partial interpretation $\mathcal{I}_{ex}^{\perp}$ that assigns $dom(c)$ to all c.

An *atom* is an expression of the form $c = v$ where $v \in dom(c)$. Atoms can be combined into *formulas* by means of the Boolean operators $\neg$, $\vee$ and $\wedge$. A *theory* consists of a set of constraints and definitions. A *constraint* is simply a formula. A *definition* is a set of *rules* of the form $A \leftarrow \varphi$ where A is an atom and φ a formula. Essentially, such a rule states that φ implies A and that, in addition, A may only hold if at least one of the rules of the definition implies it. The formal and informal semantics of rule-based definitions were discussed at length in [12].

Continuing the example, the theory T_{ex} consists of the following single definition:

$$\left\{\begin{array}{l} ApplicableRate = 1 \leftarrow RegistrationType = Social.\\ ApplicableRate = 7 \leftarrow RegistrationType = Modest.\\ ApplicableRate = 10 \leftarrow RegistrationType = Other. \end{array}\right\}$$

Partial interpretations evaluate formulas (and by extension atoms, constraints, definitions, theories) with a three-valued truth value in the natural way. A partial interpretation $\mathcal{I}$ *satisfies* a formula φ if it evaluates the formula to true, written as $\mathcal{I} \models \varphi$. For atoms in particular, $\mathcal{I}$ evaluates $a = v$ to true if $a^{\mathcal{I}} = \{v\}$, to false if $v \notin a^{\mathcal{I}}$, and to unknown otherwise. We say $a = v$ *holds* in $\mathcal{I}$ if $\mathcal{I} \models a = v$, and *does not hold* if $\mathcal{I} \models \neg(a = v)$.

---

[1] For the Flemish region the applicable legislation is the "Decreet van 13 december 2013 houdende de Vlaamse codex fiscaliteit" with amending decrees from December 19th 2014 and May 18th 2018.

A total interpretation I that satisfies all of the constraints and definitions in a theory T is called a *model* of the theory The above example T_{ex} has three models:

$\{ApplicableRate = 7, RegistrationType = Modest\}$, $\{ApplicableRate = 10, RegistrationType = Other\}$ and I_{ex}.

IDP allows generic *inferences* to be applied to an FO($\cdot$) specification. A fundamental inference is *model expansion*, which, given a theory T, expands a partial interpretation $\mathcal{I}$ into a model of T. In the case of the above example, I_{ex} is a model expansion of $\mathcal{I}_{ex}$ w.r.t. T_{ex}. In general, a given pair $(\mathcal{I}, T)$ may have zero, one, or more model expansions. The *optimisation* inference takes as input a partial $\mathcal{I}$, theory T and objective integer constant O. It then computes the model expansion of $\mathcal{I}$ w.r.t. T that is maximal (or minimal) under O. For instance, I_{ex} is the model expansion of $\mathcal{I}_{ex}^{\perp}$ that minimizes the objective constant $ApplicableRate$ w.r.t. T_{ex}.

Neither model expansion nor optimisation are particularly useful inferences in the context of an interactive application, such as the notary system. Indeed, both inferences search for a *total* interpretation and will therefore always attempt to assign a value to all unknown constants. In an interactive application, this is not the desired behaviour. For instance, if the notary has not yet filled in the number of children that the buyers have, we do not want the system to just guess a value. For this reason, the prototype of [13] relies heavily on the *propagation* inference, which computes information that is common to all possible model expansions, and hence can discover properties that are implied *regardless* of, e.g., the unknown number of children that the buyers have.

Formally, the *propagation* inference takes as input a theory T and partial interpretation $\mathcal{I}$, and outputs the most precise partial interpretation $\mathcal{I}^{prop}$ such that all model expansions I of $\mathcal{I}$ w.r.t. T are also model expansions of $\mathcal{I}^{prop}$ w.r.t. T. We say an atom is *propagated* if it is unknown in the original interpretation $\mathcal{I}$, but true or false in the more precise interpretation $\mathcal{I}^{prop}$. In the running example, given theory T_{ex} and partial interpretation $\mathcal{I}_{ex}$, invoking propagation leads to

$$\begin{aligned}\mathcal{I}_{ex}^{prop} = \{&ApplicableRate \in \{1, 7\},\\ &RegistrationType \in \{Social, Modest\}\}\end{aligned}$$

as both $\mathcal{I}_{ex}$ and $\mathcal{I}_{ex}^{prop}$ have the same two model expansions with regard to T_{ex}, but $\mathcal{I}_{ex}^{prop}$ is most precise.

Finally, given a theory T, a formula φ is T-*implied* by some partial interpretation $\mathcal{I}$, denoted $\mathcal{I} \models_T \varphi$, if φ holds in all model expansions of $\mathcal{I}$ w.r.t. T. Equivalently, $\mathcal{I} \models_T \varphi$ if φ holds in $\mathcal{I}^{prop}$ obtained by propagating $\mathcal{I}$ w.r.t. T.

IV. ORIGINAL PROTOTYPE

At the start of the case study the notary's application requirements were rather vague. The most important concern was to use the obtained information in an intelligent way, i.e., use the information instantly to derive conclusions. Because of this we opted for the use of the earlier developed *automatic configuration* interface that is available on the IDP homepage [1]. As the use of it is independent of the domain described in the vocabulary and theory in the knowledge base, this was an easy way to create a first visual prototype to solicit further application requirements from the notary. Figure 1 shows a screenshot of the original prototype that was developed in [13] using the technology from [8]. The interface allows to construct a partial interpretation: the "+" serves to assign a unique value to a constant, while the "−" removes the corresponding value from the domain of the constant. Applying propagation then leads to a more precise interpretation $\mathcal{I}^{prop}$. For each propagated atom the corresponding box is colored green if the atom is true in $\mathcal{I}^{prop}$, and red if it is false. In addition, the user may also invoke model expansion to complete the current partial interpretation into some total model, and optimization to complete it into the model expansion that minimizes the duties that need to be paid.

V. LAW AMENDMENTS AND THE KBP

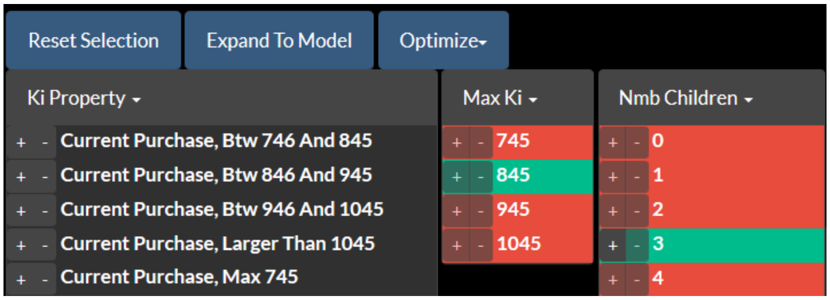
As the former version of the legislation concerning registration duties was unstructured and difficult to read, the model was built on more accessible information from the Federal Public Service Finance [14]. We analyzed and formalized the domain using the Decision Model and Notation (DMN) methodology.[2] This resulted in a model consisting of a glossary and multiple connected decision tables. This was then translated into the IDP-language. The result is an initial prototype that formalizes 11 articles of law, resulting in a knowledge base of 53 concepts, 6 constraints and 14 rules [13]. Building this knowledge base required an effort of approximately 10 person-days. A significant part of this time was attributed to the creation of the set of symbols representing concepts in the domain (i.e.; the *vocabulary*. To this end some analysis beyond the level of the DMN model was needed.

To evaluate the maintainability of the knowledge base, we examined the effort necessary to update it to the changes in legislation enacted in 2018. These changes consisted of 5 new articles, making it the most significant change to real estate sales law since the transfer of jurisdiction from the national to the Flemish regional government in 2013. At the time of constructing the original knowledge base, the content of these changes was not yet known. Therefore, this provides a realistic test case to judge the maintainability of the knowledge base.

Updating the knowledge base required only 0.5 person-days, a fraction of the time required for the initial version. 16 of the original 53 concepts were removed and 18 new ones added; 11 existing constraints and rules needed to be updated or deleted, while 4 new constraints were added. Crucially, 9 of the 20 existing constraints/rules did not need to be touched at all. This demonstrates that the inherent modularity of the KBP indeed leads to significant advantages in practice.

---

[2] The reasons and way of working with DMN are discussed in our earlier work, see [13].



Ki Property	Max Ki	Nmb Children
+ - Current Purchase, Btw 746 And 845	+ - 745	+ - 0
+ - Current Purchase, Btw 846 And 945	+ - 845	+ - 1
+ - Current Purchase, Btw 946 And 1045	+ - 945	+ - 2
+ - Current Purchase, Larger Than 1045	+ - 1045	+ - 3
+ - Current Purchase, Max 745		+ - 4

Fig. 1. Propagation in the original prototype.

VI. IMPROVED PROTOTYPE

A screenshot of the interface of the new prototype is shown in Figure 2. This interface contains several usability improvements, such as hover-over tooltips to explain the meaning of the constants and custom input fields for numerical domains. The user is also initially presented with a small set of predetermined *core* constants, and can expand this set to *relevant* constants or simply *all* constants. A more important update is the clear distinction between *chosen atoms* explicitly set by the user, and *propagated atoms* implied by the chosen atoms. The interface visualizes chosen atoms by a ⟲-symbol to indicate that this choice can be reconsidered. Propagated atoms are visualized by question marks, indicating that they can be explained.

The most significant improvements come from two new inferences, *relevance* and *explanation*. Both are refinements of algorithms that already existed in IDP, but that had not yet been used in the context of interactive decision enactment.

a) Explanation: To increase user confidence in the system, it is important that the system is able to explain why it derived certain conclusions. Moreover, the user sometimes would like to flip the propagated atom's truth assignment. The identification of chosen atom assignments allows the user to revise his choices and perhaps change the outcome of his query. The *explanation* inference takes care of this job. As input, it takes a theory T, a partial interpretation $\mathcal{I}$ and a propagated atom a. As output, it returns a least precise partial interpretation $\mathcal{I}^{expl}$ such that a still is T-implied by $\mathcal{I}^{expl}$. When a user clicks on the question mark of a propagated atom a, the system constructs the partial interpretation $\mathcal{I}_{chosen}$ from all the chosen atoms, and feeds a and $\mathcal{I}_{chosen}$ to IDP's explanation inference together with the theory T containing all domain knowledge. The output then represents a minimal subset of all chosen atoms that T-imply a's propagated value, which is presented to the user as an explanation for the propagation. As shown in Figure 2, the user can consult the related law article directly by using the information button in the explanation box.

b) Relevance: One of the key problems with the original prototype was that it encouraged notaries to ask irrelevant questions. For instance, the knowledge base included the concept of a *licensed seller*: only when the seller is licensed, can the property be eligible for a social registration. In particular, the definition of $RegistrationType$ contains the following rule:

$$\left\{\begin{array}{l} RegistrationType = Social \leftarrow \\ Seller = Licensed \;\land\; Purpose = SocialHabit. \end{array}\right\}$$

Moreover, this is the only formula where the licensed seller concept is used. Once the notary has determined that the purpose of the real estate is not social habitation, there is no longer any need for the notary to ask whether the seller is licensed. However, the original prototype would keep on displaying this as an undecided atom, tempting the notary into inquiring about it.

Our new prototype makes use of the *relevance* inference to avoid this problem. This inference takes as input a theory T, a partial interpretation $\mathcal{I}$ that is closed under propagation, and a set of *goal constants* C. Its output is a set of *relevant* atoms $a = v$ that can still affect the interpretation of the constants in C, given T and the information that is already in $\mathcal{I}$. In the case of our example, if $\mathcal{I}$ is such that $SocialHabit \in Purpose^{\mathcal{I}}$ and $C = \{RegistrationType\}$, then the atom $Seller = Licensed$ is relevant, as choosing it true might imply $RegistrationType = Social$; alternatively, choosing it false implies $RegistrationType \neq Social$. If $SocialHabit \notin Purpose^{\mathcal{I}}$, then $RegistrationType = Social$ is false in all model expansions of $\mathcal{I}$ w.r.t T and $Seller = Licensed$ is therefore irrelevant.

IDP's relevance inference is based on justification theory (e.g., [10]). As mentioned in Section III, the language that we use in this paper is a highly simplified version of the real FO($\cdot$) language used in IDP. Similarly, the concepts we introduce in the following paragraphs are highly simplified versions of the original justification theory and of the implementation of the relevance inference that is available in our software tool.

The *dependency graph* of a theory T has all of the subformulas of the theory as its nodes and has an edge from each formula to all of its subformulas. In addition, for each rule of the form $A \leftarrow \varphi$, there is also an edge from the atom A to the formula φ. Intuitively, each directed edge from φ to ψ in this graph means that the truth of φ is defined (or can be *justified*) by the truth of φ. Finally, we also add each of the goal constants C to the graph and include an edge from each goal constant $c \in C$ to all atoms of the form $c = v$. The idea behind these edges is that value of the

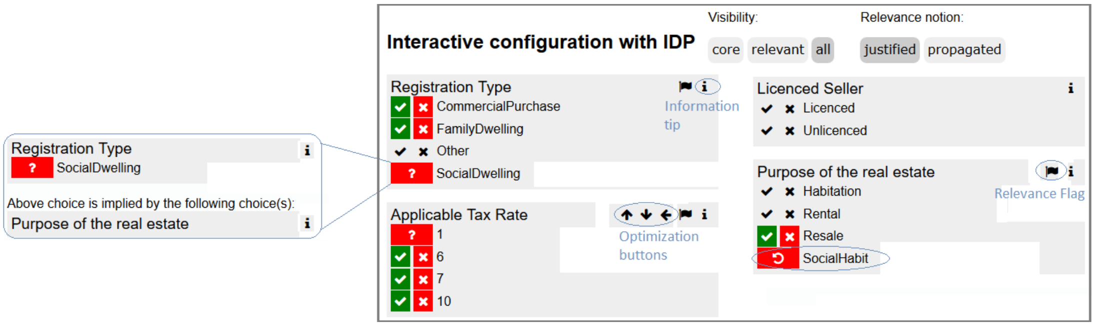


Fig. 2. Relevance and explanation demonstrated in the new interface.

goal constant c is of course influenced by the truth of these atoms $c = v$. We denote the resulting graph by G_T^C. We say that a formula φ is T-*determined* by $\mathcal{I}$ if either $\mathcal{I} \models_T \varphi$ or $\mathcal{I} \models_T \neg\varphi$. Finally, we define that an atom $c = v$ is *relevant* if it is not T-determined by $\mathcal{I}$ and there exists a path in G_T^C from this atom to one of the goal atoms, which does not traverse any node that is T-determined by $\mathcal{I}$. Intuitively, an atom is relevant if its truth value may affect the value assigned to one of the goal atoms *and* this possible effect is not blocked by the fact that the truth value of some intermediate formula is already fixed by the current partial interpretation. In case of the above example, $Purpose \neq SocialHabit$ would T-imply $RegistrationType \neq Social$ and therefore making this choice would make $RegistrationType = Social$ T-determined. Hence, $RegistrationType = Social$ blocks the path from $Seller = Licensed$ to the goal constant $RegistrationType$, making this atom irrelevant. This basic notion of relevance can be further refined in order to better handle special cases such as unfounded choices and recursive definitions [16], but this is out of scope of this paper.

In the interface, a relevant choice $c = v$ can be made with a red and green button, while irrelevant choices can still be made, but the buttons are grey. In addition, the box for constant c is flagged in the upper right corner to indicate that at least one relevant atom $c = v$ still exists. Finally, the view level *relevant* displays only relevant unknown atoms. Figure 2 shows the atom $Seller = Licensed$ is indeed irrelevant (for the implicit goal constant $RegistrationType$) in the given context.

VII. RELATED WORK

We are not the first to model legislation into a logic-based language. In the United Kingdom, several pieces of legislation were represented as executable logic programs. For instance, the British Nationality Act [18] and a large part of the Supplementary Benefit Legislation [4] were modelled in Prolog.

Later, a shift from logic programs to description logic knowledge bases occurred. Early examples include Valente's functional ontologies [19] and Van Kralingen's frame-based ontologies [20]. The advantages of description logics over logic programs are that these are simpler and decidable logics of which the decision procedures are tractable for a machine. However, this advantage also comes at a cost: by limiting complexity, expressivity is often limited as well. Hence, to express a complex legal statement, auxiliary symbols will often be required. In the extreme case, it might not even be possible to express certain laws.

Nevertheless, there have been European projects that model legislation into description logic knowledge bases, such as the HARNESS project [7], its successor AGILE [5], and Emerald [15]. Alongside them, XML standards were developed to express such description logic knowledge bases. Examples include the Legal Knowledge Interchange Format (LKIF) [6], Akoma Ntoso [2], and the Legal Metadata Interchange Format (LMIF) [17].

All research above is focused on a single kind of a reasoning (deductive reasoning or satisfiability checking), whereas our approach is multi-inferential by construction. This multi-inferential nature allows us to perform different reasoning tasks all with the same modelled legislation, which is crucial for an interactive decision enactment system.

Regarding the formalization of the legal domain, some interesting suggestions have been done by [3] and [20]. They suggest the use of an intermediate model between the knowledge domain and the final knowledge base. The purpose of this intermediate model is to ensure a thorough analysis of the domain, independent of the implementation goal. Although we share this concern, we see an additional role for the intermediate model, i.e., to facilitate communication between the domain expert and the modeller. The use of the DMN-based tool of OpenRules, allows to define concepts and attributes in a business glossary, while rules are formalized in decision tables. These parts are analog to the *class hierarchy* and *rule base* parts suggested by [3]. Especially in [3], the importance of an analog structure of the knowledge base and legislation, what they refer to as *isomorphism*, is stressed. While our program shows some isomorph characteristics , we sometimes deviate from the principle. For example, subsection 2 contains a number of articles that each describe a separate registration type with its applicable tax rate. In the knowledge base of our application, the registration type is defined in one definition (with each rule referring to a separate article). The tax rate however, is defined in a separate definition (referring to the same articles). The dogmatic use of isomorphism seems less relevant in the limited scope of our application and with the implemented features of explanation and relevance.

VIII. DISCUSSION AND CONCLUSION

This paper presents an advanced prototype of an interactive decision enactment system, developed to support notaries during client meetings. It improves the original prototype of [13] in two ways. First, the knowledge base was updated to reflect substantial changes to the regulations that came into force on June 1st 2018. Second, new inferences were integrated to meet additional requirements articulated by the notary. These results validate two central claims of the Knowledge Base Paradigm. First, the effort to update the knowledge base (0.5 person-days) was very small in comparison to the effort to create the initial knowledge base (10 person-days), especially when taking the size of the legal changes into account. This demonstrates the maintainability of an approach based on the KBP. Second, the improvements to the user interface demonstrate the feasibility of an approach in which the knowledge base is developed separately from the inference methods that can be applied to it. In particular, we have implemented two pieces of functionality that were not originally foreseen when the knowledge base was developed, but that were demanded later by the notary office. We did so by applying two fully generic inferences to the existing knowledge base.

The *relevance* inference addresses the need for efficient information gathering, as it narrows down the entire set of undecided atoms to those that matter for top-level decisions. This helps the notary avoid requesting superfluous information from his clients. Once an atom is propagated by the system, the *explanation* inference allows to explore why this particular outcome is implied.

More generally, the contributions of this paper are validation of the claims that knowledge bases are easy to maintain, even in the face of considerable changes in the domain; and that knowledge bases can be reused for other, unanticipated inference tasks. It also shows a first-time integration of the relevance and explanation inferences in an interactive application, and demonstrates their practical utility. The resulting interactive decision enactment system is applicable to a wide range of applications. Future work with regards to the developed interactive decision enactment system will focus on the use of the system in other legal domains and the application of new generic inferences.

REFERENCES

[1] https://krr.bitbucket.io/.

[2] Gioele Barabucci, Luca Cervone, Monica Palmirani, Silvio Peroni, and Fabio Vitali. Multi-layer markup and ontological structures in akoma ntoso. In *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue - International Workshops AICOL-I/IVR-XXIV Beijing, China, September 19, 2009 and AICOL-II/JURIX 2009, Rotterdam,The Netherlands, December 16, 2009 Revised Selected Papers*, pages 133–149, 2009.

[3] T. J. M. Bench-Capon and F. P. Coenen. Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, 1(1):65–86, Mar 1992.

[4] Trevor J. M. Bench-Capon, G. O. Robinson, Tom Routen, and Marek J. Sergot. Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In *Proceedings of the First International Conference on Artificial Intelligence and Law, ICAIL '87, Boston, MA, USA, May 27-29, 1987*, pages 190–198, 1987.

[5] Alexander Boer and Tom Van Engers. The agile project: Reconciling agility and legal accountability. In *Proceedings of the 2nd International Conference on ICT Solutions for Justice*, pages 41–49, 2009.

[6] Alexander Boer, Radboud Winkels, and Fabio Vitali. Metalex XML and the legal knowledge interchange format. In *Computable Models of the Law, Languages, Dialogues, Games, Ontologies*, pages 21–41. Springer, 2008.

[7] J. Breuker, S. van de Ven, A. El-Ali, M. Bron, R. Hoekstra, S. Klarman, U. Miloševíc, L. Wortel, A. Förhécz, and P Estrella. Deliverable n : 4.6 developing HARNESS towards a hybrid architecture for LKIF. 2008.

[8] Ingmar Dasseville, Laurent Janssens, Gerda Janssens, Jan Vanthienen, and Marc Denecker. Combining DMN and the knowledge base paradigm for flexible decision enactment. In Tara Athan, Adrian Giurca, Rolf Grütter, Mark Proctor, Kia Teymourian, and William Van Woensel, editors, *Supplementary Proceedings of the RuleML 2016 Challenge, Doctoral Consortium and Industry Track hosted by the 10th International Web Rule Symposium, RuleML 2016, New York, USA, July 6-9, 2016.*, volume 1620 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[9] Broes De Cat, Bart Bogaerts, Maurice Bruynooghe, Gerda Janssens, and Marc Denecker. Predicate logic as a modelling language: The IDP system. *CoRR*, abs/1401.6312v2, 2016.

[10] Marc Denecker, Gerhard Brewka, and Hannes Strass. A formal theory of justifications. In Francesco Calimeri, Giovambattista Ianni, and Mirosław Truszczyński, editors, *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *Lecture Notes in Computer Science*, pages 250–264. Springer, 2015.

[11] Marc Denecker and Joost Vennekens. Building a knowledge base system for an integration of logic programming and classical logic. In María García de la Banda and Enrico Pontelli, editors, *ICLP*, volume 5366 of *LNCS*, pages 71–76. Springer, 2008.

[12] Marc Denecker and Joost Vennekens. The well-founded semantics is the principle of inductive definition, revisited. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *KR*, pages 1–10. AAAI Press, 2014.

[13] Marjolein Deryck, Faruk Hasic, Jan Vanthienen, and Joost Vennekens. A case-based inquiry into the decision model and notation (DMN) and the knowledge base (KB) paradigm. In *Rules and Reasoning - Second International Joint Conference, RuleML+RR 2018, Luxembourg, September 18-21, 2018, Proceedings*, pages 248–263, 2018.

[14] Federale Overheidsdienst Financien. Kopen en verkopen met registratierecht.

[15] A. Förhécz, G. Korösi, A. Millinghoffer, and G. Strausz. Emerald: Legal knowledge engineering using OWL and rules. In *Legal Knowledge and Information Systems - JURIX 2009: The Twenty-Second Annual Conference on Legal Knowledge and Information Systems, Rotterdam, The Netherlands, 16-18 December 2009*, pages 53–58, 2009.

[16] Joachim Jansen, Bart Bogaerts, Jo Devriendt, Gerda Janssens, and Marc Denecker. Relevance for SAT(ID). In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 596–603. IJCAI/AAAI Press, 2016.

[17] M. Palmirani, L. Cervone, and F. Vitali. Legal metadata interchange framework to match CEN metalex. In *The 12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 8-12, 2009, Barcelona, Spain*, pages 232–233, 2009.

[18] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The british nationality act as a logic program. *Commun. ACM*, 29(5):370–386, 1986.

[19] André Valente. *Legal knowledge engineering: A modelling approach*, volume 30. Penn State Press, 1995.

[20] Robert W. van Kralingen, Pepijn R. S. Visser, Trevor J. M. Bench-Capon, and H. Jaap van den Herik. A principled approach to developing legal knowledge systems. *Int. J. Hum.-Comput. Stud.*, 51(6):1127–1154, 1999.

[21] Tomasz Zurek. Conflicts in legal knowledge base. *Foundations of Computing and Decision Sciences*, 37(2):129–145, 2012.