2019 IEEE 13th International Conference on Semantic Computing (ICSC)

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 allows 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 interactive 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 (e.g., [13]). In [11], an interactive decision enactment system for notaries was developed according to the Knowledge Base Paradigm (KBP) [9].

The KBP advocates a strict separation between declarative domain knowledge and the use of this knowledge to perform certain tasks. This separation allows the same knowledge to be used by different *inference* algorithms in order to achieve different goals. As claimed by [9], 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 [11], a decision enactment system that supports Belgian notaries in their handling of real estate sales was developed according to the KBP. 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 was developed together with notary Luc Van Pelt. Like other Belgian notaries, his office prides itself on its customer-friendly and confidential service. Therefore, he is looking for a system that provides support while interviewing clients, without interrupting the natural flow of the conversation.

In this paper, we further analyze and develop the prototype that was developed in [11]. This work focuses on validating the two advantages of the KBP 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 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 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 can indeed be implemented by applying domain-independent inference methods to a purely declarative knowledge base, even if this functionality was not originally foreseen when the knowledge base was constructed.

This work results in a completely generic framework, similar to, but more powerful than that of [6]. 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 [7], which allows it to benefit from both this system's expressive knowledge representation language FO($\cdot$), as well as from its efficient inference algorithms.

The next section elaborates on the background of the case, the main characteristics of the original prototype and the changes in legislation. Section III introduces the KBP and the IDP system. Section IV elaborates on new inferences in the revised interface. Section V presents related work, followed by a discussion and conclusion in Section VI.

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]

[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.

978-1-5386-6783-5/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSC.2019.00040

Prior to June 1st 2018, the central concept to determine the registration type was the *kadastral income (KI)*, a value which represents its theoretical rental value. In the new legislation, the concept of *KI* was abandoned. To determine if a house should be considered "modest", its actual selling price is now used instead of the fictitious *KI*. The reforms of the registration duties represent a profound change: of the original 42 articles of chapter 9 concerning the registration law, the decree of May 18th 2018 abolishes 4 articles, modifies 9 and adds 5.

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 an earlier developed *automatic configuration* interface available online [1]. In this prototype, the user is presented a list of unknown *atomic* statements (e.g., whether the property can be considered "modest") which can be assigned appropriate truth values. The prototype continuously *propagates* all consequences following from such assignments under the rules specified by registration duty law, further restricting the values of the atoms. In addition, the user may also request an *expansion* of the current assignment to a satisfying configuration, which optionally minimizes the duties that need to be paid.

After evaluating this prototype, the notary office came up with two additional requirements. First, *traceability*: the outcomes propagated by the application should be easy to check and explain, in order to increase clients' confidence in the application. Second, *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, atoms related to this discount become irrelevant and should no longer displayed.

The initial prototype formalizes 11 articles of law, resulting in a knowledge base of 53 concepts, 6 constraints and 14 rules [11]. 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 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.

III. The knowledge base paradigm and IDP

The prototype uses the IDP knowledge base system, which employs FO($\cdot$) as a formal knowledge base specification language [7]. The core of FO($\cdot$) is typed first-order logic, extended with inductive definitions, aggregates and arithmetics [7]. In this section, we only recall a propositional fragment of the language, applied to 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 $\{v_1, \ldots, v_n\}$. As a running example, we use the selection of an appropriate rate for the calculation of the registration fee. Here, we have a constant $ApplicableRate$ with domain $\{1, 7, 10\}$, and a constant $RegistrationType$ with domain $\{Social, Modest, Other\}$.

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, $c^I \in c^{\mathcal{I}}$.

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 (see also [10]).

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 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.

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, namely one for each possible $ApplicableRate$.

IDP allows generic *inferences* to be applied to an FO($\cdot$) specification. The *optimisation* inference takes as input a partial interpretation $\mathcal{I}$, a theory T and an objective integer constant O. It then computes the model expansion of $\mathcal{I}$ w.r.t. T that is maximal (or minimal) under O.

In an interactive application it is not always desirable to search for a *total* interpretation. 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 [11] 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

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

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.

IV. Improved prototype

The new interface contains usability updates, such as information tooltips, custom input fields and a start screen with a limited set of predetermined *core* constants. A more important improvement is the clear distinction between *chosen atoms* (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 improvement is the application of the *relevance* and *explanation* inferences for interactive decision enactment.

a) Explanation: to increase user confidence, it is important that the system is able to explain why it derived certain conclusions. Moreover, the user sometimes would like to flip a propagated atom's truth assignment. The explanation inference identifies the chosen atoms that imply a propagated atom, and allows to revise these choices. 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 would still be propagated by $\mathcal{I}^{expl}$. This is demonstrated in Figure 1. 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 still imply a's propagated value, which is presented to the user as an explanation for the propagation.

b) Relevance: A key problem of 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:

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

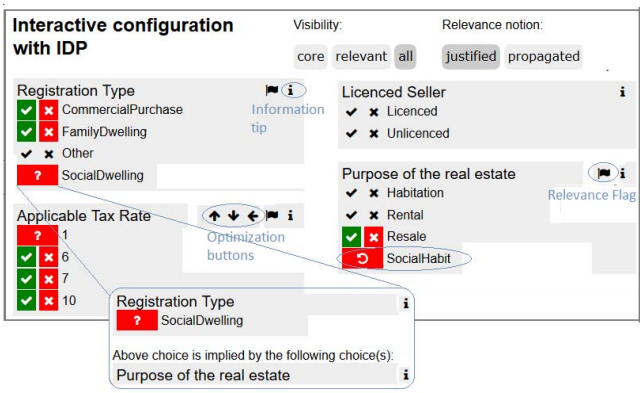

Fig. 1. Relevance and explanation demonstrated in the new interface.

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 to determine whether the seller is licensed.

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 present in $\mathcal{I}$. In 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 its truth value determines whether $RegistrationType = Social$. 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 not relevant.

IDP's relevance inference is based on justification theory (e.g., [8]). 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 goal constant c is 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 $\mathcal{I}$.

In the interface, a relevant choice $c = v$ is highlighted with red and green buttons (to assign it true or false), while irrelevant choices can still be made, but the buttons are grey. In addition, the box for a constant c is flagged in the upper right corner to indicate that at least one relevant atom over c still exists. Finally, it is also possible to hide irrelevant unknown atoms. Figure 1 shows the atom $Seller = Licensed$ is indeed irrelevant (for the implicit goal constant $RegistrationType$) under the given truth assignment.

V. 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 [13], [3]. Later, a shift from logic programs to description logic knowledge bases occurred [14], [15]. These are simpler, decidable logics for which the decision procedures are tractable for a machine. However, this 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 [5]. Alongside them, XML standards were developed to express such description logic knowledge bases [4], [12].

All research above is focused on a single kind of 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 modeled legislation, which is crucial for an interactive decision enactment system.

Regarding the formalization of the legal domain, some interesting suggestions regarding the use of an intermediate model between the knowledge domain and the final knowledge base have been done by [2] and [15].

VI. 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 [11] in two ways. First, the knowledge base was updated to reflect substantial changes to the regulations. 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, 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 regarding 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.

Acknowledgment

This research was supported by the Swedish Research Council grant 2016-00782. Marjolein Deryck is supported by VLAIO, the Flemish Agency for Innovation and Entrepreneurship, as part of the TETRA-project "Decision Analytics".

References

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

[2] 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.

[3] T. J. M. Bench-Capon, G. O. Robinson, T. Routen, and M. 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.

[4] A. Boer, R. Winkels, and F. Vitali. Metalex XML and the legal knowledge interchange format. In *Computable Models of the Law, Languages, Dialogues, Games, Ontologies*, pages 21–41. Springer, 2008.

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

[6] I. Dasseville, L. Janssens, G. Janssens, J. Vanthienen, and M. Denecker. Combining DMN and the knowledge base paradigm for flexible decision enactment. In T. Athan, A. Giurca, R. Grütter, M. Proctor, K. Teymourian, and W. 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.

[7] B. De Cat, B. Bogaerts, M. Bruynooghe, G. Janssens, and M. Denecker. Predicate logic as a modelling language: The IDP system. *CoRR*, abs/1401.6312v2, 2016.

[8] M. Denecker, G. Brewka, and H. 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.

[9] M. Denecker and J. 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.

[10] M. Denecker and J. 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.

[11] M. Deryck, F. Hasic, J. Vanthienen, and J. 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.

[12] 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.

[13] 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.

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

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