

# CALCULEMUS: Towards a Formal Language for the Interpretation of Normative Systems

Robert van Doesburg and Tijs van der Storm and Tom van Engers<sup>1</sup>

**Abstract.** In this paper we will introduce a method for creating computational models of normative systems deduced from sources of norms in natural language.

The authors show how to make formal interpretations of normative sources in natural language that result in a computational model, which includes explicit references to all sentences of sources of norms that are considered relevant by the interpreters to constitute a computational model. The models produced can easily be held in sync with these sources.

The method presented is focused on the translation of laws in natural language into an executable computational form that can be easily validated by legal experts that have to decide on the desired interpretation of the source text. The model is tested in a prototype of a reasoner build in a newly developed domain specific language: FLINT. The model is based on Hohfeld's fundamental legal concepts.

## 1 INTRODUCTION

Organizations that handle thousands or even millions of cases a year depend on a form of computational law to be able to use supporting IT-systems. These organizations are accountable for building and maintaining such systems in compliance to the norms they are submitted to. This is the work of knowledge engineers that use experts' knowledge elicitation processes to incorporate these experts' interpretations of the normative sources of their organizations.

The two primary sources of norms are: legislation, i.e. bills and operational policy documents that all typically describe how generic abstract cases are to be treated, and case decisions in judicial procedures, from which we may learn how a specific individual case is to be treated, and that might have an impact on future cases too.

Knowledge engineers are typically intermediating between the (legal) experts and technical IT staff. They lack a method to formally link the knowledge of the elicited domain experts to the normative sources in natural language that these domain experts use to acquire their knowledge. This is especially problematic in case of changes in normative sources. Organizations need to quickly understand the impact of such changes and adapt their supporting IT-systems accordingly.

In the early nineties and the first decade of the twenty-first century solutions for this problem were presented [1][16][17], but none of these methods are presently being used on production scale within governmental organizations or industries. In this paper we will shortly describe the difference between our approach and early work. An elaborated overview of the various earlier approaches and the relation to our work will be published as a separate paper, this paper is too short for that exposé.

In this paper we present our approach, called CALCULEMUS, after the ideas of Leibniz who was the first that aimed at solving legal problems by means of calculation. We will demonstrate how it can be applied on actual legal sources with an example from Dutch Immigration Law. The resulting model is expressed in a domain specific language (DSL), FLINT (Formal Language for the Interpretation of Normative Theories). This DSL is specific in so far that it is targeted towards the specific way we express norms. We will illustrate this by giving an example of FLINT expressions.

The CALCULEMUS method and the FLINT prototype result from a co-operation between the Dutch Immigration and Naturalisation Service (IND), the Dutch Tax and Customs Administration (DCTA) and the Leibniz Center for Law. This paper is a report on the progress made on this subject since the NWO Workshop ICT with Industry in December 7-11, 2015 [11].

## 2 RELATED WORK

Our approach is based on the work of Wesley Newcomb Hohfeld and the fundamental legal concepts he introduced in 1913 [6]. Hohfeld's motive to introduce these legal concepts was his opinion that one of the greatest hindrances to the clear understanding of legal problems is the explicit or tacit assumption that all legal relations may be reduced to "rights" and "duties". Hohfeld proofed this was not the case by describing the ambiguities in the meaning of these concepts and went on to introduce a smallest set of legal conceptions to which, according to him, any and all 'legal quantities' could be reduced.

Hohfeld distinguished four Legal Relations: 'Power-Liability relations' (1), 'Immunity-Disability relations' (2), 'Duty-Claimright relations' (3), and 'Privilege-Noright relations' (4). Some scholars prefer 'Liberty-Noright relations' instead of 'Privilege-Noright relations'. We also use the first term.

The Hohfeldian legal conceptions can only exist in pairs and describe relations between two people, each holding one of the rights in a pair. 'Power-Liability relations' and 'Immunity-Disability relations' are generative: they can generate new 'Legal Relations'. The 'Duty-Claimright relations' and 'Privilege-Noright

---

<sup>1</sup> Leibniz Center for Law, University of Amsterdam, Netherlands, email: [RobertvanDoesburg@uva.nl](mailto:RobertvanDoesburg@uva.nl);  
CWI, Netherlands, email: [Storm@cw.nl](mailto:Storm@cw.nl)  
Leibniz Center for Law, University of Amsterdam, Netherlands, email: [vanEngers@uva.nl](mailto:vanEngers@uva.nl)

relations' are situational: they can only be created and terminated by a generative 'Legal Relation'.

To make our interpretations maximum traceable to the normative sources they are based upon, we strive for isomorphism. Trevor Bench-Capon [1][2] and Van Engers [12][15] are amongst the people that have stressed the importance of creating isomorphism between the formal models that represent sources of law and those sources.

Compared to the method presented in 1991 by Bench-Capon the CALCULEMUS approach is more precise in the explicit notation of references between sentences in normative sources. The fact that we have good and (inter)nationally accepted mechanisms for representing references and standards for identifying building block in sources of law, such as the Dutch Juriconnect standard and the European MetaLex standard [3] helps enormously.

In addition to that in CALCULEMUS a method for the formal interpretation of norms is used that represents the *rules of the game*.

Those rules can be used to actually *play the game*, but making models that describe *games*, i.e. models that include agency, intent and the dynamics of social interaction, are a separate issue. In this paper we will restrict ourselves to making formal interpretations of the *rules of the game*.

The method for interpreting norms used in CALCULEMUS, has similarities with Van Kralingen's norm-framework approach [16]. Van Kralingen, like us, uses Hohfeld's fundamental legal concepts as a foundation. However he chose to change the names of these concepts and dropped Hohfeld's focus on legal relations. His approach mixes up the description of *the rules of the game* and those of *playing the game*, although the latter does not include some important aspects of social interaction, e.g. agency needed to reason about the impact of norms on society. In our opinion this weakens the usefulness of his frame-based conceptual models, and resulted in an approach that is less attractive for legal experts.

In the nineties knowledge engineers focused on abstract legal ontologies and different of these legal ontology frameworks were developed [16]. The main focus of the research on such abstract formal conceptual models was on their computational abilities. How to actually make concrete conceptualizations, or legal domain ontologies from a jurisprudential, or legal perspective was listed as future work [16]. The method presented in this paper aims to fill this gap.

### 3 THE CALCULEMUS APPROACH

The CALCULEMUS approach is a normative system in three layers: sources of norms in natural language (1), the formal interpretation of norms in a 'Institutional Reality' (2), and the use of a formal interpretation of norms in 'Social Reality' (3) (see figure 1).

This model is an extended version of the three layers of reality model presented in [13] and was based upon the work of Searle [9]:

#### 1. Sources of Norms

This layer describes the components, structure and referential mechanisms that allow us to refer to the natural language sources describing the norms we want to 'translate' into formal computational models.

#### 2. 'Institutional Reality'

This layer describes the interpretation of the sources of norms in the previous layer, using: states representing situations; legal positions; and acts regulated by norms.

#### 3. 'Social Reality'

The 'Social Reality' layer describes agents, agent-roles, collaboration of agents, coordination, message passing, and other behavioral aspects of agents. This layer is used to describe and simulate behavior in societies regulated by norms. These norms can be used, e.g., to test (non-) compliance scenarios, and to predict effectiveness.

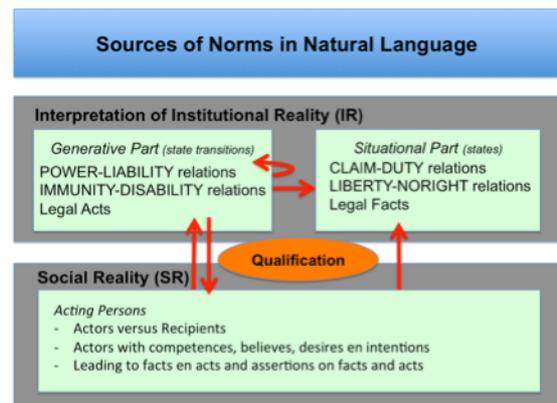


Figure 1. Three layers of reality

The second layer, 'Institutional Reality', is constructed to enable computational law. Concepts, or 'institutional facts' are derived from sources of norms, and are put in an explicit structure. Van Engers and Van Doesburg have introduced 'Institutional Reality' based on Hohfeld's fundamental legal concepts [14].

The third layer is the 'Social Reality' that can contain any brute or social fact. To qualify a social fact as a 'institutional fact' a qualified official is needed. This can be the administrator deciding on an application or objection, or it can be a judge ruling on an appeal.

The CALCULEMUS approach results in 'institutional facts' that can be used by our DSL-based reasoner build in DSL FLINT, to calculate normative positions.

'Social Reality' is modeled using agent-role modeling, see for example [10]. This paper focuses on the second layer the formal interpretation of norms, using the newly developed Formal Language for the Interpretation of Normative Theories: FLINT.

### 3.1 'Institutional Reality' for normative systems

'Institutional Reality' is build out of normative relations. A normative relation (NR) is an extension of the concept of a 'legal relation', as defined by Hohfeld [6]. Legal relations are based on legal sources in natural language. A normative relation can have any normative source. The elements of a 'Institutional Reality' for normative systems are described individually below.

## Facts and Acts

Institutional facts (iFACTs) in FLINT are facts that can be traced back to a normative source. Facts in ‘Social Reality’ can be qualified as iFACTs.

Institutional acts (iACTs) are acts that can be traced back to a normative source. Acts in ‘Social Reality’ can be qualified as iACTs.

## ‘Normative Relations’

Situational Normative Relations (‘situational NRs’) exist in two types: ‘Claimright-Duty relations’ and ‘Liberty-Noright relations’. They exist of the following elements: the ‘holder’ of a ‘Claimright’ or ‘Liberty’; the holder of a ‘Duty’ or ‘Noright’; the ‘Object of the normative relation’; and the ‘Duty’ or ‘Noright’ itself. For every element of the ‘situational NR’ references to normative sources are registered.

‘Situational NRs’ are created and terminated by ‘generative Normative Relations’.

Generative Normative Relations (‘generative NR’) also exist in two types: ‘Power-Liability relations’ and ‘Immunity-disability relations’. They exist of the following elements: an ‘Actor’ (a person); a ‘Recipient’ (another person); an ‘institutional act (iACT)’; the ‘Object of the normative relation’; a ‘precondition’ and a ‘postcondition’. The references to normative sources of the elements of a ‘generative NR’ are also registered.

The precondition of a ‘generative NR’ is a iFACT or a set of iFACTs and ‘situational NRs’ combined using Boolean connections.

The postcondition is a set of iFACTs and ‘situational NRs’ that are created and/or terminated by a ‘generative NR’. The ‘postcondition’ can only be reached when the ‘precondition’ is met and an act in ‘Social Reality’ is qualified as the iACT belonging to the ‘generative NR’ that the ‘postcondition’ is a part of.

The ‘postcondition’ describes the transition of the initial state that fulfills the ‘precondition’ to an end state.

## 3.2 An example

The case study in this paper is on the admission of international students to the Netherlands. The case study is described in more detail in [14]. In this paper an example of a ‘generative NR’ relevant for the case study is used to present the Domain Specific Language ‘FLINT’. The sources of law used in this example are English translations of the original Dutch text published on [overheid.wetten.nl](http://overheid.wetten.nl) by the Dutch Formal Publications Office.

The ‘generative NR’ with code NR.AA.16.1.b is a Power-Liability relation based on article 16, paragraph 1, point b, AA. This article describes the Power of Our Minister of Justice to reject an application for a residence permit if the alien does not possess a valid border crossing document. The alien is the ‘Recipient’ in this NR. The ‘precondition’ of NR.AA.16.1.b is the existence of a application (iFACT ApplicationExists) and the existence of the

iFACT that the alien does not possess a valid border crossing document (iFACT NoValidBorderCrossingDoc).

The ‘precondition’ also exist of the absence of three exceptions for the Power to reject an application on the ground that the alien does not possess a valid border-crossing document. These exceptions are: the alien proves that he can not (any longer) be put in possession of a valid border-crossing document due to the government of his country (1), the alien is citizen of Somalia and the Netherlands do not recognize the Somalian authorities and Somalian documents (2), and the alien is a child born in this country born who apply for stay with their parents, provided it meets the conditions for residence with its parents (3).

The first exception is described in article 3.72 of the Aliens Decree (AD). The second and third exception in chapter B1/4.2, sentence 4 of the Aliens Act Implementation Guidelines (AAIG).

## 4 FLINT: A Domain-Specific Language for Specifying Norms

### 4.1 Introduction

Domain-specific languages (DSLs) are software languages tailored toward a particular problem domain. Well-designed DSLs provide custom notation, which is closer to the concepts and vocabulary of domain experts. This improves productivity (shorter programs), quality assurance (better error message through domain-specific checks), and stakeholder communication (programs are expressed in terms of the problem domain). DSLs have been successfully applied in areas such as financial services, hardware description, and web development (for related work on DSLs, see [8]).

Although DSLs provide a bridge between a problem domain and its technical realization in software, DSL development requires both language engineering expertise and domain knowledge. Recent developments in the area of language workbenches provide integrated tool support for significantly lowering the complexity of this task [5]. Language workbenches take traditional compiler tools to the next level, by also providing support for defining editor services, such as syntax highlighting, outline views,

cross-referencing, static analyses, and documentation generation. Although often overlooked, user-level tool support is essential for adopting formal languages in non-technical domains.

In this section we present FLINT, a DSL for describing and executing norms. The current prototype of FLINT is designed as a textual language using the meta programming system Rascal [7].

### 4.2 FLINT

FLINT is a domain-specific formal language (DSL) that is targeted towards describing our models of ‘Institutional Reality’. We will illustrate FLINT by using an example which formalizes the “reject” relation introduced in 3.3.2, see figure 2 for an example of the source code.

```

iFact ApplicationExists
source: Article 14, paragraph 1, point a, AA
{The application to grant a temporary residence permit}

iFact NoValidBorderCrossingDoc
source: Article 16, paragraph 1, point b, AA
{The alien does not possess a valid border-crossing document.}

iFact CannotHaveBorderCrossingDoc
source: Article 3.72, AD
{The alien proves that he can not (any longer) be put in
possession of a valid border-crossing document due to
the government of his country.}

iFact CitizenOfSomalia
source: B1/4.2 sentence 4 AAIG
{The alien is citizen of Somalia.}

iFact ChildrenBornInNL
source: B1/4.2 sentence 4 AAIG
{Children born in this country born who apply for stay with their parents,
provided they meet the conditions.}

iFact RejectedBecauseNoValidBorderCrossingDoc
source: Article 16, paragraph 1, introduction and point b, AA
{The application to grant a temporary residence permit is rejected because the
alien does not possess a valid border-crossing document.}

relation NR.AA.16.1.b: [Our Minister] has the power towards [the alien]
to [reject] [the application to grant a temporary residence permit]
source: Article 16, paragraph 1, introduction and under point b, AA
when
ApplicationExists AND NoValidBorderCrossingDoc
AND NOT (CannotHaveBorderCrossingDoc OR CitizenOfSomalia OR ChildrenBornInNL)
action:
+ RejectedBecauseNoValidBorderCrossingDoc
{An application to grant a temporary residence permit as referred to in Article
14 may be rejected if: the alien does not possess a valid border-crossing docu

```

Figure 2. Source text FLINT

The first six declarations capture the ‘institutional facts’ that may hold or not. Each iFACT has an intuitive name (e.g., “NoValidBorderCrossingDoc”), a reference to the legal source, and the relevant fragment of the actual text of the law. Additional meta-data, such as Juriconnect identifiers that serve as references to sources of law, are normally included, but for presentation purposes have been omitted from this example.

The final declaration describes the generative POWER relation between the ‘Actor’ “Our Minister” and the ‘Recipient’ “the alien”. In this example case, it describes the power to reject an application for a temporary residence permit, on the ground of not possessing a valid border-crossing document. The ‘precondition’ encodes the requirement of this document (after the “when” keyword), which also describes exceptions (e.g., being a citizen of Somalia). Whenever the ‘precondition’ holds, the ‘Actor’ (“Our Minister”) can enact the relation, which causes the action to be executed. In this, the action consists of adding the ‘institutional fact’ “RejectedBecauseNoValidBorderCrossingDoc”. Enforcing a generative relation represents a transition into a new (institutional) world, where additional facts are added or existing facts are withdrawn, comparable to belief revision that is a well-known mechanism in AI.

In addition to checking for iFACTs in the ‘precondition’, and adding or deleting iFACTs in the ‘postcondition’, a generative rule can also query and create or withdraw situational Normative Relations or generative Normative Relations.

### 4.3 Benefits of FLINT

FLINT is accompanied with an integrated development environment (IDE), which provides editor services such as automatic syntax checking, syntax highlighting, jump-to-definition (e.g., clicking on an iFACT in a ‘precondition’ or ‘postcondition’ moves the cursor to its definition), error marking, content completion, and hover documentation. Currently, the IDE displays errors when a references iFACT or relation is not defined. In the future, we will extend this with automated cross-referencing with legal texts, more advanced consistency checks (e.g., decidability,

reachability, etc.), and refactoring operations (e.g., consistently renaming an iFACT).

We have automatically imported an initial set of legal relations and iFACTs from Excel sheets, which immediately uncovered a number of mistakes due to typos or missing iFACT declarations. Automated tooling for engineering such legal specifications is thus useful, even if the current analysis support is still quite primitive.

FLINT specifications can be used for simulating cases. This involves defining an initial world by listing the iFACT and situational relations that hold. Given this initial world, some of the generative relations are enabled, because the ‘precondition’s are true. In the simulation, the enabled relations can be fired, to obtain a new state of the world, in which a (possibly different) set of relations is enabled.

Though the study case only includes the interpretation of one ‘Normative Relation’, it does show the approach to interpret these sources.

The case can be extended in two ways:

1. By collecting and interpreting normative sources of other actions than rejecting an application because the alien does not possess a valid border-crossing document: e.g. assessing other grounds for rejection, or assessing the acts of granting or disregarding an application.
2. By collecting and interpreting normative sources of normative statements that further specify the iFACTs used in the ‘precondition’: e.g. normative rules on establishing the iFACT that the alien does not possess a valid border-crossing document, on the definition of a border-crossing document, and on the iFACTs that determine the validity of a border-crossing document.

## 5 CONCLUSION AND FUTURE WORK

In this paper we made the case for formalization of normative positions and relations, separated from the formalization of agent behavior.

One of the reasons for separating the *rules of the game* from describing the *playing of the game* is that the latter is much more difficult and requires us to really understand the full complexity of intelligently operating agents in a complex adaptive systems context. While formalizing the rules of the game is a relatively easier task and the results thereof are already showed to be beneficial for practice, research on norm-guided social behavior in complex adaptive systems is still ongoing [4].

In this paper we presented the main ideas behind our CALCULEMUS method. We presented the semi-formal model of ‘Institutional Reality’ that is an interpretation of the sources, and we showed the formal model expressed in a DSL named FLINT (Formal Language for the Interpretation of Normative Theories).

We accept that there are still many open issues particularly in modeling ‘Social Reality’. We’re grateful to being able to work in spirit of the great philosopher Leibniz that initiated the idea. Calculemus!

## ACKNOWLEDGEMENTS

We thank IND, DCTA, NWO, and the participants of the ICT with Industry workshop 2015 for their support and contributions.

## REFERENCES

- [1] T. Bench-Capon and F.P. Coenen, 'Isomorphism and legal Knowledge Based Systems', *Artificial Intelligence and Law*, 1-1, 65-86 (1991)
- [2] T. Bench-Capon and T.F. Gordon, 'Isomorphism and Argumentation', in P. Casanovas (Ed.), *Proceedings of the 12th international conference on artificial intelligence and law*, 11-20, New York, NY, USA: ACM (2009).
- [3] CEN MetaLex. *Open XML Interchange Format for Legal and Legislative Resources*. [Online]. Available from: <http://www.metalex.eu>.
- [4] A. Deljoo, L.H.M. Gommans, T.M., Engers, and C.T.A.M. de Laat, 'An Agent-Based Framework for Multi-Domain service networks: Eduroam case study', in *8th International Conference ICAART 2016*, 275-280, Springer (2016).
- [5] S. Erdweg, et al., 'Evaluating and comparing language workbenches: Existing results and benchmarks for the future', *Computer Languages, Systems & Structures*, 44, Part A, 24-47, (2015).
- [6] W.N. Hohfeld and W.W. Cook, *Fundamental legal conceptions as applied in judicial reasoning, and other legal essays*, New Haven: Yale University Press, 1919.
- [7] P. Klint, T. van der Storm, and J. Vinju, 'EASY meta-programming with rascal', in J.M. Fernandes, et al. (eds.), *GTTSE III. LNCS*, 6491, 222-289, Springer, Heidelberg, (2011).
- [8] M. Mernik, J. Heering, A.M. Sloane, 'When and how to develop domain-specific languages', *ACM Computing Surveys (CSUR)*, 37-4, 316-344 (December 2005)
- [9] J.R. Searle, *The construction of social reality*, Penguin Books, London, 1996.
- [10] G. Sileno, A. Boer, and T. M. van Engers, 'Commitments, expectations, affordances and susceptibilities: Towards positional agent programming', *PRIMA 2015: Principles and Practice of Multi-Agent Systems Lecture Notes in Computer Science*, 687-696, (2015.)
- [11] R. van Doesburg et al., 'Towards a Method for a Formal Analysis of Law, Study Case Report ICT with Industry workshop 2015', NWO (2016). [Online]. Available from: <http://www.nwo.nl/over-nwo/organisatie/nwo-onderdelen/ew/bijeenkomsten/ict-with-industry+workshop/proceedings>
- [12] T.M. van Engers, 'Legal Engineering: A Structural Approach to Improving Legal Quality', in A. Macintosh, R. Ellis, R., and T. Allen (eds.), *Applications and Innovations in Intelligent Systems XIII, proceedings of AI-2005*, 3-10, Springer, (2005).
- [13] T. M. van Engers, A. Boer, 'Public Agility and Change in a Network Environment', in Judith Schossboeck, Noella Edelmann and Peter Parycek (Eds.), *JeDEM 3(1)*, 99-117, (2011).
- [14] T.M. van Engers and R. van Doesburg, 'Modeling the Interpretation of Sources of Norms', in *proceedings of eKNOW2016*, IARIA XPS Press, 41-50, (2016).
- [15] T.M. van Engers and E. Glassée 2001. 'Facilitating the Legislation Process Using a Shared Conceptual Model' *IEEE Intelligent Systems*, 16(1), 50-58, IEEE, (2001).
- [16] R.W. van Kralingen, *Frame-based conceptual models of statute law*, Kluwer, (1995)
- [17] P.R.S. Visser and T.J.M. Bench-Capon, 'A Comparison of Four Ontologies for the Design of Legal Knowledge Systems', *Artificial Intelligence and Law*, 6-1, 27-57, (1998).