

From Legal Contracts to Formal Specifications: A Progress Report

Michele Soavi^a, Nicola Zeni^a, John Mylopoulos^b and Luisa Mich^a

^a University of Trento, Via Sommarive 14, 38123 Povo - TN, Italy

^b University of Ottawa, 800 King Edward Ave, Ottawa ON, Canada K1N 6N5

Abstract

Smart contracts are software systems that partially automate, monitor and control the execution of legal contracts. The requirements of such systems consist of a formal specification of the legal contract whose execution is to be monitored and controlled. Legal contracts are always available as text expressed in natural language. We have been working on the translation of such text documents into formal specifications. Our translation process consists of four steps that (a) Semantic annotation of text identifying obligations, powers, contracting parties and assets, (b) Identification of relationships among the concepts identified in (a), (c) Generation of a domain model for terms used in the contract, as well as identification of parameters and local variables for the contract, (d) Generation of formal expressions that formalize the constituents of obligations and powers. This paper reports on the status of the project and the results that have been achieved.

Keywords

Legal contract, smart contract, semantic annotation, domain model, formal specification

1. Introduction

The advent of software that partially monitors, automates and controls the execution of legal contracts has gained increasing interest in Academia, Government and Industry, thanks to the development of smart contracts. Such software systems have become possible thanks to blockchain and IoT technologies [1]. A first step towards a systematic software engineering (SE) process for building such software systems is to translate legal contract text expressed in natural language (NL) into a formal specification that defines precisely the terms and conditions (requirements) that a legal contract needs to fulfill and a corresponding smart contract needs to monitor.

We are interested in making the translation process systematic and tool-supported. To achieve this goal, we have decomposed the translation process into four steps: (a) Semantic and structural annotation of legal text; (b) Discovery of relationships for concepts discovered in step (a); (c) Formalization of terms used in the NL text and elements of the specification such as parameters and local variables; (d) Generation of formal expressions that use variables and parameters and capture the conditions of the contract-at-hand. This refinement of the problem is intended to make the translation process systematic, and also opens the door to tool support for each step of the process. The target language for legal contracts is Symboleo [2], a formal specification language for legal contracts. The ontology offered by Symboleo for describing legal contracts is built around the concepts of obligation and power, role and asset, event and situation.

In: F.B. Aydemir, C. Gralha, S. Abualhaija, T. Breaux, M. Daneva, N. Ernst, A. Ferrari, X. Franch, S. Ghanavati, E. Groen, R. Guizzardi, J. Guo, A. Herrmann, J. Horkoff, P. Mennig, E. Paja, A. Perini, N. Seyff, A. Susi, A. Vogelsang (eds.): *Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany, 12-04-2021*

EMAIL: michele.soavi@unitn.it (M. Soavi); nicola.zeni@unitn.it (N. Zeni); jmylopou@uottawa.ca (J. Mylopoulos); luisa.mich@unitn.it (L. Mich)

ORCID: 0000-0001-8519-1034 (M. Soavi); 0000-0003-1296-0140 (N. Zeni); 0000-0002-8698-3292 (J. Mylopoulos); 0000-0002-0018-6883 (L. Mich)



© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

The main research contribution of our project so far is the definition of a solution to the translation problem in the form of this 4-step process. The contribution is grounded on our past work with semantic annotation of legal text [3], [4], as well as the discovery of semantic relationships in legal text [5].

This progress report presents the four-step process in detail through an example, then reports on what has been accomplished so far and closes with future plans.

2. The Translation process

We illustrate the 4-step process with an example. Consider a simple sales contract between a meat producer and a supermarket chain that has clauses as shown in Table 1.

Table 1

A Meat Sale contract

- O1: Seller shall deliver a given quantity of meat of AAA quality to the warehouse of Buyer before the delivery date (Delivery obligation);
 - O2: Seller shall ensure that the meat is transported in accordance with meat transportation standards (Transportation obligation);
 - O3: Buyer shall pay agreed upon price amount within a week from delivery date (Payment obligation);
 - P1: If delivery is late, Buyer can charge Seller \$1,000 as late fee for each day of delay (LateF power).
-

The semantic and structural annotation step identifies structurally three obligations and a power. Powers are rights contracting parties have to cancel, suspend obligations, or create new ones. Moreover, this step identifies roles of parties, namely Buyer and Seller, as well as asset meat. Accordingly, the output of the first step would be something like what is shown in Table 2.

Table 2

Annotated Meat Sale contract

- <\obl <\role Seller> shall deliver a given quantity of <\asset meat> of AAA quality to the warehouse of the <\role Buyer> before delivery deadline>
 - <\obl <\role Seller> shall ensure that the meat is transported in accordance with meat transportation standards>
 - <\obl <\role Buyer> shall pay agreed upon price amount within a week from delivery date>
 - <\powr If delivery is late, <\role Buyer> can charge <\role Seller> \$1,000 as late fee for each day of delay>
-

A semantic annotation requires a common vocabulary (ontology) for legal contracts. A structural annotation requires a grammar for the syntactic structure of legal contracts [4]. Also, for each concept to be annotated in the text the annotation tool requires a lexical pattern defined as a BNF grammar.

The second step identifies relationships for each one of the concepts identified in step one. For example, each obligation must have a debtor who is obliged to fulfill it, and a creditor (beneficiary). The debtors and creditors of O1 and O2 are obviously Seller and Buyer respectively, while in O3 roles are reversed. Note that in O2 and O3 there is no mention of a creditor, so this has to be inferred by any tool addressing this step. Finally, for the power the creditor is the Buyer and the debtor the Seller. In addition, each obligation/power can have a trigger that initiates it, an antecedent that serves as precondition, and a consequent that signals successful completion of the obligation/power. P1 has a trigger 'If delivery is late', while others take a trigger 'true', indicated by T, and are initiated when contract execution starts. O1 has antecedent 'true' and consequent 'deliver given quantity of meat of AAA quality to the warehouse of the <\role [Buyer]> before delivery deadline', while O2 has antecedent 'while meat is transported' and consequent 'meat will be transported in accordance with meat transportation standards'. The output of the second step is a conceptual model of concepts and

relationships identified in the text, where for each obligation we list trigger, debtor, creditor, antecedent, consequent, and for each power we list trigger, creditor, debtor, antecedent, consequent. The result of this step has as follows:

Table 3

A conceptual model for the Meat Sale contract

O1: trigger: T, debtor: Seller, creditor: Buyer, antecedent: T, consequent: ‘deliver given quantity of meat of AAA quality to the warehouse of the Buyer before delivery deadline’;
 O2: trigger: T, debtor: [Seller], creditor: [Buyer], antecedent: ‘meat is transported’, consequent: ‘meat is transported in accordance with meat transportation standards’;
 O3: trigger: T, debtor: [Buyer], creditor: [Seller], antecedent: T, consequent: ‘pay given amount within a week from delivery date’;
 P1: trigger: ‘If delivery is late’, creditor: [Buyer], debtor: [Seller], antecedent: T, consequent: ‘pay \$1,000 as late fee for each day of delay’.

The reader can think of this as the backbone of the specification. Now we need to embellish the backbone with expressions for each trigger, antecedent and consequent. The third step concerns formalizing terms used in the contract as extensions/specializations of the legal contract ontology used in step (a). In particular, ‘meat’ is an instance of class Beef, with instances portions of beef, while Beef itself is a specialization of Meat, which has as instances quantities of meat, including beef, chicken etc. In turn, Meat is a specialization of Food, which is a specialization of Asset. Asset is one of the concepts in the contract ontology. Along the same lines, ‘warehouse’ refers to an address which is the target of the delivery, so it can be formalized as attribute delAddr of class Buyer, which is a specialization of Role, another class in the contract ontology. The result of this step is a domain model for the contract being formalized.

Formalization of terms is intended to eliminate ambiguity. For example, ‘meat’ is multiply ambiguous (could it be chicken or lamb?) Likewise, does ‘delivery’ in “If delivery is late” could refer to the delivery action of transporting the meat to its destination, or more likely to the event ‘delivered’ that happens at the end of the delivery. The formalization is based on an ontology for contracts that includes as primitives, concepts such as Role, Obligation, Power, Asset, etc. and the relationships mentioned above, such as debtor, creditor, antecedent and consequent.

This step also identifies parameters for a contract, the ‘givens’ for each contract execution, as well as local variables that refer to instances of classes in the domain model. For the meat sale contract, parameters and some local variables are presented in Table 4.

Table 4

Parameters and local variables

O1: [Seller] shall deliver [qnt] quantity of meat of AAA quality to the warehouse of the [Buyer] before [delD] date (Delivery obligation);
 O2: [Seller] shall ensure that the meat is transported in accordance with meat transportation standards (Transportation obligation);
 O3: [Buyer] shall pay [price] amount within a week from the date of delivery (Payment obligation);
 P1: If delivery is late, [Buyer] can charge [Seller] {delay-in-days} * \$1,000 as late fee (LateF power).

Here, terms in square brackets [...] identify parameters of the contract, to be determined for each contract execution, while wiggly brackets {...} define variables whose values are to be determined during contract execution time.

The fourth step concerns translating NL expressions such as ‘meat is transported’ into expressions in a formal specification language. Symbaleo is such a language where true/false statements are defined in terms of events, instantaneous happenings, and situations, states-of-affairs that occur over a period of time. For example, the trigger of P1 “If delivery is late” is translated to “happens(delivered(meat, Buyer.delAddr), t) and t after delD” where delivered(...) is an event that happens at time t. The antecedent of O2 is translated to “occurs(transport(meat), int)”, which says that the situation

transport(meat) has occurred during interval int, while the consequent is translated into “occurs(MTS(meat), int)”, which says that the transportation occurs in accordance with transportation standards (MTS) during int, the interval of the transportation situation.

The final output of the translation process is the Symboleo specification shown in Table 5. Note that the specification also includes input parameters as well as declarations for local variables that take as values instances of classes defined in the domain model.

Table 5

Formal specification for the Meat Sale contract

Domain meatSaleD

Seller isA Role with returnAddr: String;

Buyer isA Role with delAddr: String;

Beef isA Meat with quant: Number; Meat isA Food; Food isA Asset;

Delivered isA Event with item: Meat, delAddr: String, delD: Date;

Paid isA Event with amount: Number;

end Domain;

Contract meatSaleC(buyer: Buyer, seller: Seller, meat:

Beef; qnt: Number-kg; price: CAN\$, delD: Date)

Declarations

delivered: Delivered with item := meat, deliveryAddr := buyer.delAddr;

paid : Paid **with** amount := price;

Obligations

O1: O(seller, buyer, T, happens(delivered(meat, buyer.delAddr), delDate)

and meat.quant = qnt and delDate before delD);

O2: O(seller, buyer, occurs(transport(meat), int), occurs(MTS(meat), int));

O3: O(buyer, seller, T, happens(paid(price), t) and t before delDate + 7days);

Powers

P1: happens(delivered(meat, buyer.delAddr), delDate) and delDate after delD \Rightarrow

P(buyer, seller, T, happens(paid((delDate – delD)* \$1,000), t'))

end Contract

3. Progress Report

We are developing an environment that will support each step of the process by proposing an output for a given input. For example, for the input of Table 1 the tool will (hopefully) propose the annotated text of Table 2. In our earlier work, [4] we found that such a tool can generate annotations of very good quality for legal text, and can generate good quality conceptual models [5] by discovering relationships among the entities identified by the annotation. In both cases the tool is not intended to automate steps 1 and 2, but rather to support a human carrying out each step by (a) Improving the quality of the output, and (b) reducing the manual effort required to carry out each step. The tool presented in [4], intended to help a human generate a conceptual model from a legal text was found to reduce the manual effort required by 80%.

We have adopted the ideas and the tool proposed in [4] to develop a tool called Contract 1.0 that conducts structural and semantic annotation for legal contract text [6]. Moreover, we have evaluated empirically Contract 1.0 by having human subjects that generate annotations for a given contract manually, while others simply correct the output of Contract 1.0 to produce annotated text. The results of the experiment suggest that the use of Contract 1.0 can improve significantly the performance of human annotators both with respect to precision and recall. Lexical patterns have been refined in Contract 2.0 to improve the ability to recognize powers and obligations as results in the experimental evaluation were mixed [7].

Finally, a Systematic Literature Review has been compiled concerning the 4-step process [8]. The aim of the review is to understand the most relevant research efforts undertaken to complete each step and to identify best practices, development opportunities and existing gaps in approached solutions to the problem.

5. Work-To-Be-Done

Our next steps in this project include:

- a) Development of an environment that helps a user to manually translate legal contract text into a Symboleo formal specification through the 4-step process presented in this report.
- b) Development of a tool that supports mining of semantic relationships from annotated contract text (step 2). The difficulty with automatically mining relationships is that unlike concept annotation that exploits lexical indicators to determine what to annotate, relationships are often identified in the text positionally; for example, for the obligations of the Meat Sale contract, the respective debtor is the role that precedes ‘shall, while for the power of the contract the trigger expression appears at the beginning of the clause.
- c) Development of a tool that exploits WordNet [9] to support the generation of a domain model for a given contract; the tool will also identify parameters and local variables that refer to instances of classes in the domain model (step 3); Our expectation is that the identification of parameters and local variables will be mostly manual.
- d) The development of a tool that exploits FrameNet [10] to generate formal event and situation expressions from text (step 4).
- e) The empirical evaluation of the whole environment to determine the degree to which it reduces manual effort and improves the quality of output specifications.

6. References

- [1] Christidis K, Devetsikiotis M (2016) Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* 4: 2292-2303. <https://doi.org/10.1109/ACCESS.2016.2566339>
- [2] Sharifi S., Parvizimosaed A., Amyot D., Logrippo L., Mylopoulos J., “Symboleo: A Specification Language for Smart Contracts”, 28th IEEE Requirements Engineering Conference, RE@Next track, Zurich, September 2020.
- [3] Kiyavitskaya N, Zeni N, Breaux TD, Antón AI, Cordy JR, Mich L, Mylopoulos J, “Automating the Extraction of Rights and Obligations for Regulatory Compliance”, 27th International Conference on Conceptual Modelling (ER’08), November 2008, 154-168.
- [4] Zeni N., Kiyavitskaya N., Mich L., Cordy J., Mylopoulos J., “GaiusT: Supporting the Extraction of Rights and Obligations for Regulatory Compliance”, *Requirements Engineering* 20, Springer, 2015, 1-22.
- [5] Zeni N., Seid E., Engiels P., Ingolfo S., Mylopoulos J., “NomosT: Building Large Models of Law through a Tool-Supported Process”, *Data and Knowledge Engineering (DKE)*. vol.117, Elsevier, September 2018, 407-418.
- [6] Soavi M., Zeni N., Mylopoulos J., Mich L., “Contract – From Legal Contracts to Formal Specifications: Preliminary Results”, 13th International Working Conference on the Practice of Enterprise Modelling (PoEM’20), Virtual Conference, November 2020.
- [7] Soavi M., Zeni N., Mylopoulos J., Mich L., “Semantic Annotation of Legal Contracts with the Contract Tool”, *International Journal on Software and Systems Modeling (SoSyM)*. To appear.
- [8] Soavi M., Zeni N., Mylopoulos J., Mich L., “From Legal Contracts to Formal Specifications: A Systematic Literature Review”, *Universal Access in the Information Society*. To appear.
- [9] Miller G., Beckwith R., Fellbaum C., Gross D., Miller K., “Introduction to WordNet: An On-line Lexical Database”, *International Journal of Lexicography* 3(4), Winter 1990, 235–244.
- [10] Narayanan S., Fillmore C., Baker C., and Petruck M., “FrameNet meets the semantic web: A DAML+OIL frame representation”, 18th National Conference on Artificial Intelligence, Edmonton, Alberta. AAAI, 2002.