From Regulatory Texts to BRMS: How to Guide the Acquisition of Business Rules?*

Abdooulaye Guissé, François Lévy, and Adeline Nazarenko

Université Paris 13, Sorbonne Paris Cité
Laboratoire d'Informatique de Paris-Nord (LIPN), CNRS (UMR 7030)
F-93430, Villetaneuse, France

Abstract. This paper tackles the problem of rule acquisition, which is critical for the development of BRMS. The proposed approach assumes that regulations written in natural language (NL) are an important source of knowledge but that turning them into formal statements is a complex task that cannot be fully automated. The present paper focuses on the first phase of this acquisition process, the normalization phase that aims at transforming NL statements into controlled language (CL), rather than on their formalization into an operational rule base. We show that turning a NL text into a set of self-sufficient and independent CL rules is itself a complex task that involves some lexical and syntactic normalizations but also the restoration of contextual information and of implicit semantic entities to get a set of self-sufficient and unambiguous rule statements. We also present the SemEx tool that supports the proposed acquisition methodology based on the selection of the relevant text fragments and their progressive and interactive transformation into CL rule statements.

1 Introduction

Checking the conformance of a process with respect to regulations is a growing domain of application for business rule management systems (BRMS). For instance, in order to export cars in various countries, car manufacturers have to satisfy safety and quality tests described in UNO regulations (*e.g.* 50 pages without annexes for the sole safety belts) and others constraints from the European and national authorities. Moreover, these regulations evolve over time (the UNO text has been modified 10 times between 2005 and 2009). Even if efficient rule systems are now able to exploit and maintain large rule bases, rule acquisition remains a bottleneck, and text-based rule acquisition is an important challenge.

We propose a method for the acquisition of rules and a tool, SemEx, which supports that method and guides the acquisition process. This approach relies

* This work was realized as part of the FP7 231875 ONTORULE project (`http://ontorule-project.eu`). We thank our partners for the fruitful discussions, especially to John Hall (Model Systems) for introducing us to the SBVR world and to Audi for the collaboration on their use case. We are also grateful to American Airline who is the owner of one of our working corpora.

A. Bikakis and A. Giurca (Eds.): RuleML 2012, LNCS 7438, pp. 77–91, 2012.

on two strong assumptions. First of all, we believe that acquiring conformance rules from regulations cannot be fully automated. This is due to the complexity of human natural language (NL). The acquisition strategy that we propose relies on the cooperation of a domain expert and local automated processes. The expert controls the transformation of the regulation but automatic processes ease up the expert work. Second, we consider that it is difficult if not impossible to translate directly NL regulations into an operational rule bases expressed in formal language. We rather propose to decompose the formalization work into two main phases and to use controlled languages (CL) such as SBVR structured English (SBVR-SE[1]) as an intermediate language. A domain expert designs a set of CL rules from the source NL regulations and this set of candidate rules is then passed on to a specialist of information technologies (IT) that formalizes the candidate rule base taking the characteristics of the final application and the constraints of the rule engine into account. We focus here on the first phase, the normalization of the source regulation into a set of candidate rules written in CL, rather than on the second phase, the transformation of the CL statements into rules, which has been more studied. We show how this normalization process can be divided into intermediary steps, which allows to decompose the expert work and to guide each step with specific helping tools.

This rule acquisition method and the associated SemEx tool have been developed as part of the ONTORULE project, which aim was to define an integrated platform for acquisition, maintenance and execution of business-oriented knowledge bases combining ontologies and rules. The work has been tested on two industrial use cases. The *AAdvantage use case* aimed at developing a classification application to determine the benefits that an airline customer retention program member has earned over a given period. The business rule model had been designed from the documentation downloaded from the American Airlines (AA) web site, in particular the Terms and Conditions (5,744 words), which describes the membership statuses and the associated benefits. In the *Audi use case*, a rule application has been being defined to certify the conformance of Audi procedures with vehicle safety international regulation. The present experiment is based on the aforementioned UNO regulation. In each use case, the normalization process has been guided by a domain ontology that had been built beforehand.

Section 2 presents the state of the art. The normalization methodology and the architecture of the SemEx tool are described in Section 3. Section 4 details the framework that we propose for the progressive translation of texts into CL, showing the complexity of the involved linguistic and semantic transformations. Section 5 presents the results of that normalization processes in our use cases.

2 Related Works

BRMS are useful for propagating automatically the changes in the business of organizations into their information systems [1]. According to [11], the different forms of business rules can be seen as a continuous flow of models: the rules

[1] Semantic Business Vocabulary and Rules `http://www.omg.org/spec/SBVR/1.0/`

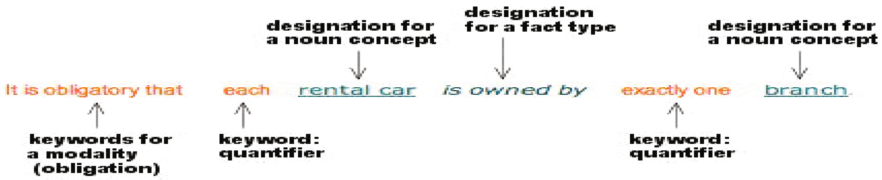

designation for a noun concept
designation for a fact type
designation for a noun concept
It is obligatory that
each
rental car
is owned by
exactly one
branch
keywords for a modality (obligation)
keyword: quantifier
keyword: quantifier

Fig. 1. SVBR SE business rule example (www.brcommunity.com)

evolve from an initial state (the rules are included in documents specifying the system), to a final state (they are formalized and integrated in information systems). However, some problems remain unsolved regarding 1) the acquisition of rules from specification documents, 2) their modeling in a formal language that enable their automation, 3) their integration in BRMS (storage, exploitation and maintenance).

The present work focuses on the first two points. This question, already raised by [3,11,9], concerns the transformation from informal to formal knowledge and the translation of text fragments written in NL into formal rules. This translation is difficult to automate, due to the complexity of NL and reduced expressivity of formal languages. Even the translation into SPARQL of LN queries, which are much simpler than texts, is acknowledged as a complex problem. To the best of our knowledge, only [8] considers a direct translation of legal texts but, after a parsing step, the abstract syntax trees are translated by hand into CTL[2].

Controlled languages have been proposed as intermediate languages in this translation process [18]. They allow to reformulate rules in a way that is still readable for the user and is easier to formalize than NL. In the Business Rules domain, CLs are used in Oracle Policy Modeling Suite[3], in IBM SPARCLE policy workbench [4]. Other in use controlled languages have been described by RuleSpeak [16], by Atempto Controlled English (http://attempto.ifi.uzh.ch/site/docs/).

SBVR (Semantics of Business Vocabulary and Rules) can be seen as a synthesis of several efforts and a standard independent of English or any natural language. It has been accepted by the OMG (Object Management Group)[4]. We refer to the English version of SBVR CL, namely SBVR Structured English (SBVR-SE). SBVR relies on formulas (Figure 1) combining linguistic basic templates with logical, modal or quantification operators.

The NL to CL translation is a complex task, the automation of which has been rarely considered. Recently, [2] has proposed NL2SBVR[5], a tool to automatically translate NL into SBVR-SE. According to the reported experiments, the complexity of the translation depends on the number of clauses that compose a sentence. Only so-called simple rules, composed of at most two clauses, are

[2] CTL is a modal temporal logic.

[3] www.oracle.com/technology/products/applications/policy-automation

[4] http://www.omg.org

[5] http://www.cs.bham.ac.uk/~isb855/nl2ocl/projects.html

translated with a 80% success rate. This is the reason why, in SemEx, translation into CL and simplification go along for complex and long NL rules.

NL simplification has been studied to ease translation [17], human understanding (esp. in case of understanding disorders [13]), text summarization [10,7], foreign language learning (see [6] for a general presentation). Methods give a significant role to the lexical part, trying to stick to a privileged vocabulary, and to the brevity of sentences. Our transformation process (see Section 4) relies on the same methods but differs in its goal. The above works aims at preserving the discursive structure of texts while sometimes simplifying the information content, whereas, in business rule acquisition, the simplification of text should give a set of independent rule statements but preserve their meaning.

3 Normalization Process

3.1 Overall Approach

The normalization process takes a regulatory text as input and outputs a new "text" composed of a list of independent, self-sufficient rules. The rules are written in a language that is as controlled as possible and the set of rule forms a draft of a business rule model (the basis of a formal rule model). The proposed approach relies on the *selection* of the relevant text fragments (sentences or sequences of sentences) that convey rule information, and on their *normalization*, *i.e.* on their translation into CL. Both steps are difficult to handle automatically. The selection step calls for browsing facilities, since missing some important passages leads to a partial BR model. Only simple natural language statements can be automatically and reliability translated in CL. An interactive approach is therefore adopted, which consists in transforming step-by-step a regulatory text fragment into one or several independent rules written in controlled language.

This approach is illustrated on Figure 2. The underlying methodology is supported by the SemEx tool[6], which offers two main acquisition functionalities for the selection and normalization of rules. Those functionalities and the corresponding perspectives of SemEx are presented in sections 3.3 and 3.4[7].

For traceability, diagnosis and revision purposes, the whole set of transformation results needs to be stored and mined. SemEx therefore relies on a rich annotation scheme to encode the rule base under construction and the source text from which it derives. Once it is built, the annotation structure can be explored through dedicated search functionalities. This annotation scheme is described in Section 3.2.

SemEx is built on W3C standards and technologies, which enables the reuse of resources (OWL ontologies) and components (SPARQL search engine).

[6] `http://www-lipn.univ-paris13.fr/~guisse/index.php?n=Semex.Semex`

[7] Additional functionalities are accessible *via* SemEx for text annotation and for mining the text and the resulting business model, but they are not described in detail here. See [12]. for an overall description of the tool.

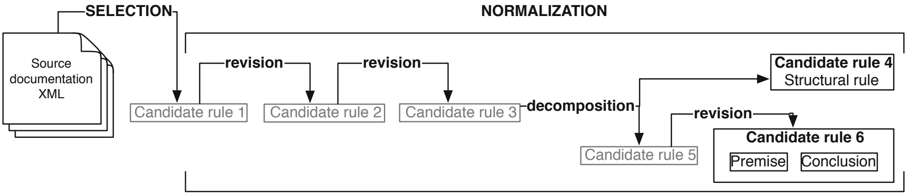


Fig. 2. Rule acquisition overall approach. Two final candidate rules (4 and 6) have been derived from the same initial candidate rule (1) extracted from the source text.

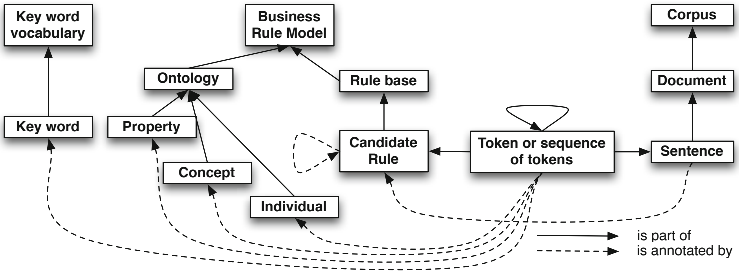


Fig. 3. Annotation scheme

3.2 Annotation Scheme

The annotation scheme describes the data structure in which the source document and the business rules that derive from it are encoded. The basic elements of this scheme are the textual units that are either tokens or sequences of tokens (elementary character strings resulting from a segmentation process). The source document is actually represented as a corpus which is composed of one or several documents. A document is a list of sentences, which are themselves represented as sequences of tokens. The rule base is composed of a set of candidate rules which are also sequences of tokens.

Two types of annotation relations are defined. The high-level ones relate sentences and candidate rules. The rule base is similar to a document but the candidate rules that compose it are partially ordered by an annotation relation. A sentence as a whole can be annotated by a candidate rule which results from its selection and a candidate rule as such can be annotated by one or several candidate rules that derive from it. Low-level annotation relations link the textual units that compose the sentences and the candidate rules to elements of the conceptual (or lexical) and logical (or grammatical) vocabularies: the ontological elements (concepts, properties or individuals) that compose the domain ontology chosen to interpret the source document, and the keywords that serve as grammatical words in the controlled language.

In technical terms, this data structure is encoded as a RDF graph. Annotation links are encoded in RDFa: the RDF annotations are anchored in textual units of XML documents and refer to resources that are OWL entities or candidate rules.

For visualization, low-level annotations are usually represented in a SBVR-style, where the annotated units are colored in blue, red, green and orange according to the type of element (concept, instance, property or known key word) that is referred to. High-level annotations are encoded as explicit references in the source sentence or candidate rule that point to the target candidate rules.

3.3 Selection of Regulatory Fragments

The first challenge for the knowledge engineer who develops a business rule model is to identify the relevant parts of the document and to select the fragments that convey regulatory information. He/she mainly has to read through the source text but the relevant information is often scattered in large and complex documents. At this step, precision must be favored over recall, as it is harder to recover missed fragments than to drop irrelevant ones.

SemEx proposes several devices to help that selection task. The first one is the low level of annotations. When browsing the annotated text in which all the recognized textual units are colored, the expert can focus on passages in which many textual units have been identified. The keywords are especially useful: sentences with a lot of annotated keywords are likely to be relevant.

The second device is a small information extraction engine that allows to design extraction patterns in an interactive mode and look for matching fragments. For instance, a sentence that follows the structure "if... [then]... must..." is likely to convey relevant information. Only a rudimentary version of the extraction engine is integrated in SemEx for now but it could be extended. The goal is to store the patterns so that the most generic and reliable ones can be reused from one application to another.

The third device is a semantic search engine that returns a list of sentences or candidate rules in answer to a semantic query. For instance, if one looks for all the sentences that mention the concept participant, one gets all the sentences that contain the word "participant" but also in which the participant is designated as "a participating company" or even as a "member of the program". This ensures a high recall level.

3.4 Normalization

Once the relevant fragment are identified, they must be normalized into CL. The goal is to get rid of ambiguities, to homogenize the lexical and syntactic turns and to make explicit all useful information. This is a complex process that cannot be fully automated. SemEx methodology supports an interactive and progressive process that transforms the initial sentence extracted from the source document into a standardized one, which is written in CL or as close as possible.

In technical terms, the expert can derive a new candidate rule from an existing one. The derivation is encoded as an annotation link and any intermediary step can be restored at any moment. The next section details the transformations that can be applied during the normalization of a rule base. Figure 4 on page 88 gives an example of a derivation tree.

4 Guiding the Transformation Process

This section presents the types of elementary transformations that are needed to translate a NL rule into controlled language. Each of them is explained and illustrated on our use cases. Transformations aim at clarifying the text provided to IT specialists in charge of implementing the rules. This involves reformulating ambiguous or tricky sentences, while preserving the meaning of the underlying rules. Each rule must also be formulated in a self sufficient way, so that its operative meaning can be determined without referring to the source text or to other rules.

Our target language is close to SBVR-SE. The main difference is that we exploit a lexicalized ontology [15] to represent the domain and its conceptual vocabulary. This vocabulary has a narrower scope than that of SBVR, which often includes specific and general-purpose dictionaries[8] whereas general purpose vocabulary remain mostly out of the scope of our transformations.

We identified four types of transformation that are presented below. The *lexical normalizations* replaces the terms of a candidate rule so as to stick to the domain vocabulary. The *decontextualization* makes explicit the contextual elements of meaning, so that the resulting rules be understandable independently of the source text and other rules. The *syntactic normalization* simplifies the syntactic structure of the sentence so that it is unambiguous and easy to understand. The *semantic normalization* operates at the semantic level, where discourse entities not explicitly referred in the text must often be introduced.

4.1 Lexical Normalization

The operation of lexical normalization aims at checking the business vocabulary of a candidate rule and at replacing all the mentioned terms by their preferred forms. This transformation process takes as input a candidate rule and a lexicalized ontology, which specifies not only the relevant concepts and properties for the field of application but also the preferred and alternative terms to refer to them [15]. The goal is that the candidate rule conform to that vocabulary. The rule terms must be disambiguated and made as specific as possible with respect to the terminology associated to the domain ontology.

The lexical normalization is based on the recognition of the terms of a candidate rule. This operation relies on the annotation of the candidate rule with respect to the ontology: a semantic annotator is integrated in SemEx [12]. Terms recognized as preferred terms in the ontology are left *as is* but alternate terms are replaced by their associated preferred ones. These annotation and replacement processes is based on a lemmatized version of the text as output by a part-of-speech tagger[9] to ensure the linguistic correctness of the resulting rule.

[8] In the SBVR-SE specification, vocabularies can use 'Authoritative dictionaries for the relevant natural languages' [14, p.133]. The EU-rent example incorporates Merriam-Webster Unabridged [14, p.275].

[9] We rely on the TreeTagger (`http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`)

The normalization process also requires that the ambiguity of terms which can get several annotations be solved. The expert has to identify the relevant term meaning in the context and select the proper unambiguous preferred terms.

For instance in

> *Two **belts** or **restraint systems** are required for the **buckle inspection** and the **low-temperature buckle test**.*

The concepts BuckleInspection and LowTemperatureBuckleTest are represented by their preferred terms "buckle inspection" and "low-temperature buckle test", but "belts" and "restraint systems", which stand for the concepts SeatBelt and ChildRestraintSystem, are replaced by the preferred forms of these concepts ("seat belt" and "child restraint system"). This lead to the following transformed candidate rule:

> *Two **seat belts** or **child restraint systems** are required for the **buckle inspection** and the **low-temperature buckle test**.*

Lexical normalization also involves nominalizations when a domain concept or entity is mentioned through a verbal phrase. For instance, "be tested for strength" should be replaced by "undergo a strength test" in the following rule:

> *All the adjustment devices shall **be tested for strength** as prescribed in paragraph 7.5.1.*

4.2 Decontextualization

Decontextualization extends lexical normalization in that it tracks references to business concepts which are not made by the specific business vocabulary stored in the lexicalized ontology, but by a word or phrase (the referent) which co-refer to a pre-mentioned word or phrase (the antecedent) and whose meaning depends on the antecedent's one. The co-reference link must be broken and the actual meaning of the referent must be made explicit so that the rule can be understood independently of its context. Various types of referent can be found.

Grammatical Words. Pronouns and possessive adjectives often embed a reference to a business entity. In the following rule, "They" should be replaced by "The adjustment devices".

> *All the adjustment devices shall undergo a strength test as prescribed in paragraph 7.5.1. **They** must not break or become detached under the tension set up by the prescribed load.*

This type of coreference can often be solved automatically using an anaphora solver, which identifies the referring items and their antecedent. We plan to integrate such an anaphora solver in SemEx and to test the benefit of this additional helping tool but we do not have such an experimental feedback yet.

Generic Business Terms referring to high-level concepts are often used to refer to more specific ones. This is a stylistic way to avoid repetitions when the context is clear enough, but those references must be made explicit in context-independent candidate rules. In the UNO regulation, "Test" is often used for the specific test under description. In the following example, "test" means "micro-slip test" and the generic term must be replaced by the specific one.

> *The samples to be submitted to the micro-slip test shall be kept for a minimum of 24 hours in an atmosphere having a temperature of 20 ±5 °C and a relative humidity of 65 ±5%.* ***The test*** *shall be carried out at a temperature between 15 and 30 °C*

Individual Constraints are often left implicit to skip straightforward details. For instance, in

> ***Mileage credit*** *will be credited only to the account of the AAdvantage member who flies, etc.*

"Mileage credit" does not refer to the plain general concept MileageCredit, but to a specific instance earned by the AAdvantage member who took the mentioned flight ("who flies"). Decontextualization yields to

> *Mileage credit* ***awarded for a flight*** *will be credited only to the account of the AAdvantage member who* ***takes that flight****, etc.*

Searching for implicit individual constraints is difficult. We plan to compare the concepts in the rule to configuration of roles in the ontology, as the triangle which links the mileage credit earned for a ticket, an AAdvantage member who buys a ticket and a flight for which the ticket is delivered.

Reference Keys are symbols or numbers which refer to a distant piece of text. We observed that in regulation texts, the accompanying text can take various forms but that reference keys are often used to introduce exceptions. Clarification is therefore both important and cumbersome.

In the following example, two load determination procedures must be defined depending on the fact that the buckle is part of the attachment or not. A complex reorganization of the candidate rule is therefore necessary.

> (7.5.1) *The buckle shall be connected to the tensile-testing apparatus and the load shall then build up to 980 daN . . . If the buckle is part of the attachment, the buckle shall be tested with the attachment, in conformity with paragraph 7.5.2. below,*
> (7.5.2) *The attachments shall be tested in the manner indicated in paragraph 7.5.1., but the load shall be 1,470 daN.*

Such a text reorganization cannot be made automatically, but navigation facilities can be proposed so that the expert can easily identify reference keys and get a quick access to the referred parts of text.

4.3 Syntactic Normalization

Syntactic normalization aims at giving a more standard phrasing of the simple candidate rules or at splitting them into several simpler ones that are easier to understand. This normalization is close to the text simplification operations that have been proposed for English [17,5], but some of these simplifications are not adequate in our business rules context and we propose more specific structures and transformations for our specialized texts.

Sentence Reordering reorganizes the sentence to stick to the order of a logical rule pattern. This often lead to exchange the main and subordinate clauses in a candidate rule. In the following example, the elliptic ”upgrade” is understood as a coreference and the clauses are reordered:

> *Upgrades **are void if** sold for cash or other consideration.*
> ***If** upgrades are sold for cash or other consideration, these upgrades **are void**.*

Splitting Enumerations. Enumerations are a well-known factor of sentence complexity and splitting enumerations leads to decompose candidate rules into simpler ones. Enumerations have various linguistic forms : pairs of connectors such as *either...or*, *neither...nor*, *not only...but also*, *whether...or*, etc., coordinating conjunctions (*and*, *or*) or plain juxtaposition. The enumerated list can be the subject, the object or even the verb of the clause. For instance, the following candidate rule should be split into three independent sentences:

> ***Neither** accrued mileage, **nor** award tickets, **nor** upgrades are transferable by the member upon death.*
> *Accrued mileage is not transferable by the member upon death. Award tickets are not transferable by the member upon death. Upgrades are not transferable by the member upon death.*

Enumerations are difficult to handle automatically. Coordination markers are easy to detect but the scopes of the enumerations are not. Their interpretation is sometimes difficult: splitting is correct only if the enumeration clusters independent conditions; otherwise, it may lead to errors, as in:

> *Mileage credit may not be combined among AAdvantage members, their estates, successors and assigns.*

Splitting Rules. Independently of enumerations, complex candidate rules often need to be split. Solving a coreference is a frequent cause that should be often handled automatically when an anaphora solver will be integrated into Semex. In the following example, the decontextualization of the pronouns ”which” and ”yours” leads to split the candidate rule into three independent ones:

> *The membership year, **which** is the period in **which** your elite benefits are available, runs from March 1 through the last day of February of the following year.*

> *The **membership year** is a period. **Member's** elite benefits are available in the **membership year**. The membership year runs from March 1 through the last day of February of the following year.*

4.4 Semantic Restoration

Semantic restoration is a fourth kind of transformation that is often implied by decontextualization or syntactic normalization : discourse entities, which are implicit in the source documents, often have to be restored during normalization.

Restoring an Entity to Solve a Reference. In some cases of decontextualization, there is no unambiguous designation available for solving a coreference and a new entity must be introduced.

In the following example, expliciting ”which” by ”two perpendicular axes” misses the coreference, using ”these perpendicular axes” does not solve it but introducing a reference to a SensitivityTestAxes concept enables the rule split.

> *When retractors are being tested for sensitivity to vehicle deceleration they shall be tested at the above extraction along two perpendicular axes, **which** are horizontal if the retractor is installed in a vehicle as specified by the safety-belt manufacturer.*

> *When retractors are being tested for sensitivity to vehicle deceleration they shall be tested at the above extraction along the **sensitivity test axes**. **Sensitivity test axes** are perpendicular. **Sensitivity test axes** are horizontal if the retractor is installed in a vehicle as specified by the safety-belt. manufacturer.*

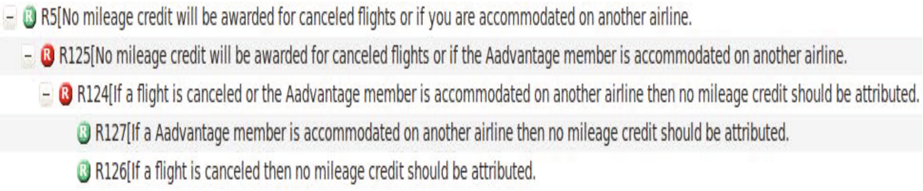
Restoring an Interval to Express Constraints. It often happens that constraints between entities are only expressible with the help of an interval that is not mentioned as such in the text.

In the following example, the transformation depends on the concepts available to refer to time entities and the proposed solution assumes there is none and defines them all:

> *The breaking load shall be determined within 5 minutes **after** the strap is removed from the conditioning atmosphere or from the receptacle.*

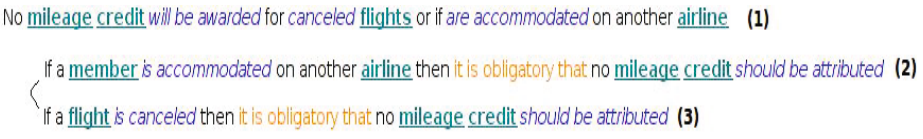
> *The determination time is the time when the breaking load is determined. The removing time is the time when the strap is removed from the conditioning atmosphere or from the receptacle. The delay between the removing time and the determination time will be less than 5 minutes*

These cases frequently occur for temporal and spatial constraints and keyword search (*e.g. after*, *until*, *since*) should help to detect the problematic rules.



- R5[No mileage credit will be awarded for canceled flights or if you are accommodated on another airline.
  - R125[No mileage credit will be awarded for canceled flights or if the Aadvantage member is accommodated on another airline.
    - R124[If a flight is canceled or the Aadvantage member is accommodated on another airline then no mileage credit should be attributed.
      - R127[If a Aadvantage member is accommodated on another airline then no mileage credit should be attributed.
      - R126[If a flight is canceled then no mileage credit should be attributed.

Fig. 4. Example of a derivation tree



No mileage credit will be awarded for canceled flights or if are accommodated on another airline **(1)**

If a member is accommodated on another airline then it is obligatory that no mileage credit should be attributed **(2)**

If a flight is canceled then it is obligatory that no mileage credit should be attributed **(3)**

Fig. 5. SBVR translation

5 Experiments and Results

This normalization method has been tested on the two ONTORULE industrial use cases for which two real BR models and rule applications had to be defined, exploiting available sources of information. In each case, a rule base has been designed by extracting a set of candidate rules from the source regulatory texts and by normalizing them, using as many transformation steps as necessary. Transformations which are not yet proposed by a tool incorporated in SemEx have been handled manually by the expert, so as to test the completeness and the correctness of the description.

Figure 4 shows the example of Rule R5, on which a decontextualization transformation (R125) and then a syntactic normalization (R124) are applied before it is decomposed into two elementary and independent sub-rules (R126 and R127). We tried to automatically translate the initial and the final candidate rules into SBVR, using the NL2SBVR tool [2].The result (Figure 5) shows that the meaning of the initial sentence is lost in the translation (statement 1), but that the translation works for the transformed final candidate rules giving valid SBVR statements (2 and 3).

The following tables present the types and size of the rule bases that have been designed out of the source regulatory texts for each use case. Four types of transformations are considered here: the normalization of the vocabulary; the restoration of contextual information, the syntactic transformation and the decomposition of rules. Since the syntactic and semantic transformations are often performed together, the semantic transformations are not isolated here.

Table 1 shows the results of the selection phase. It gives the size of the initial rule bases with respect to the size of the source texts. In the use cases, 1/4 at least of the sentences have been selected as relevant regulatory information.

This high rates are due to the fact that the source text are short but dense. This table also shows the initial structure of the rule base. More than 2/3 of the extracted rule are structural rules, some of them are operative rules and few of them are derivation rules.

Table 1. Results of the selection phase on the AAdvantage and Audi use cases (CR = candidate rule; SR = structural rule; OR = operative rule; DR = derivation rule)

Use Case	# of sentences	# of initial CR	Selection rate	# of SR	# of OR	# of DR
AAdvantage	245	74	30%	54	14	6
Audi	221	54	25%	45	9	0

Table 2 details the types of transformations that have been made for each use case. Four types of transformation are considered: the first three ones respectively affect the vocabulary, the context and the syntax; the last one is the decomposition of one candidate rule into several ones. In terms of number of transformations, the syntactic normalization is the most expensive one: it requires twice as much transformation steps than other types of transformations. All the initial rules have undergone a syntactic normalization (100%). The vocabulary transformation and decontextualization also affect more than half of the initial candidate rules, whereas decomposition is required in less numerous cases (resp. 35% and 20% for the AAdvantage and Audi use cases).

Table 2. Distribution of the different types of transformation (α = rate of a given type of transformation when considering all the transformations that have been made; β = rate of initial candidate rules that have undergone a given type of transformation)

Normalization types	AAdvantage α	AAdvantage β	Audi α	Audi β
Vocabulary	19%	65%	20%	61%
Context	18%	60%	19%	57%
Syntax	43%	100%	47%	100%
Decomposition	21%	35%	14%	20%

The last table (3) presents the structure of the resulting rule base, the number and types of the final candidate rules. As expected, there are more final rules than initial ones since some of them have undergone a decomposition. At the end of the normalization process, the regulatory information written in the AAdvantage text (245 sentences) has been reduced to a set of 104 candidate rules which are autonomous SBVR statements. The reduction is even higher for the Audi use case (225 sentences, 65 candidate rules), whose source text is more detailed.

Table 3. Structure of the final rule base

Use Case	SR	OR	DR	Final CR
AAdvantage	71	27	6	104
Audi	54	11	0	65

6 Conclusion

This paper tackles the rule acquisition problem, assuming that regulations written in NL are a rich source of knowledge but that turning NL into formal statements is a complex task than cannot be fully automated. We propose to decompose the acquisition process into two main phases: the translation of NL statements into CL and their formalization into an operational rule base.

The present paper focuses on the first "normalization" phase. It shows that transforming NL statements into CL is itself a complex task that involves some lexical and syntactic normalizations but also the restoration of contextual information and of implicit semantic entities to get a set of self-sufficient, unambiguous and easy to understand rule statement. We also present the SemEx tool that supports the proposed acquisition methodology based on the selection of the relevant text fragments and their normalization into a SBVR-like CL.

SemEx has been designed as an interactive rule acquisition tool. It guides the domain expert through a sequence of steps that produces elementary candidate rules, helps the detection of relevant keywords and controls the results. Some helping tools have already been plugged in SemEx: *e.g.* a semantic annotator that takes a lexicalized ontology as input and annotates a text with respect to that ontology and a keyword search that helps locating the most relevant text fragments or identifying the problematic features (*e.g.* anaphoric pronouns) in the selected fragments.

In the next future, we plan to exploit more intensively NL processing tools to guide and help the acquisition task. Part of the morphological and syntactic calculus could be automated using a parser. Some anaphora could be solved and syntactic transformation patterns could be exploited. We are currently testing these technologies to enrich SemEx with additional helping tools. We are also planning to integrate a SBVR parser to check the syntactic conformity of the final candidate rules with SBVR-SE. Successful checking would indicate that the transformation phase is achieved and failures would give some indication on how to complete it. The last and most challenging tool would be a semantic parser able to check the conformity of the final candidate rule to the underlying ontology. That would help to identify the semantic shortcuts that need to be made explicit in the candidate rules.

References

1. Bajec, M., Krisper, M.: Issues and challenges in business rule-based information systems development. In: ECIS (2005)
2. Bajwa, I.S., Lee, M.G., Bordbar, B.: Sbvr business rules generation from natural language specification. In: AAAI Spring Symposium 2011 Artificial Intelligence 4 Business Agility, pp. 541–545. AAAI Press, San Francisco (2011)

3. BRG: Defining business rules what are they really? The Business Rules Group : formerly, known as the GUIDE Business Rules Project - Final Report revision 1.3 (July 2000)
4. Brodie, C., Karat, C.-M., Karat, J.: An empirical study of natural language parsing of privacy policy rules using the sparcle policy workbench. In: SOUPS 2006 (2006)
5. Candido Jr., A., Maziero, E., Gasperin, C., Pardo, T.A.S., Specia, L., Aluisio, S.M.: Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In: Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications, EdAppsNLP 2009, pp. 34–42. Association for Computational Linguistics, Stroudsburg (2009)
6. Chandrasekar, R., Doran, C., Srinivas, B.: Motivations and methods for text simplification. In: Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING 1996), pp. 1041–1044 (1996)
7. Chandrasekar, R., Srinivas, B.: Automatic induction of rules for text simplification (1997)
8. Dinesh, N., Joshi, A., Lee, I., Sokolsky, O.: Reasoning about Conditions and Exceptions to Laws in Regulatory Conformance Checking. In: van der Meyden, R., van der Torre, L. (eds.) DEON 2008. LNCS (LNAI), vol. 5076, pp. 110–124. Springer, Heidelberg (2008)
9. Dubauskaite, R., Vasilecas, O.: An open issues in business rules based information system development. In: Innovative Infotechnologies for Science, Business and Education, vol. 1 (2009)
10. Gasperin, C., Specia, L., Pereira, T.F., Aluisio, S.M.: Learning when to simplify sentences for natural text simplification. In: ENIA 2009 (VII Encontro Nacional de Inteligência Artificial) (2009)
11. Halle, B., Goldberg, L., Zackman, J.: Business Rule Revolution: Running Business the Right Way. Happy About (2006), `http://books.google.com/books?id=I3mvAAAACAAJ`
12. Lévy, F., Nazarenko, A., Guissé, A., Omrane, N., Szulman, S.: An environment for the joint management of written policies and business rules. In: Proceedings of the International Conference on Tools with Artificial Intelligence (IEEE-ICTAI 2010), pp. 142–149 (2010)
13. Max, A.: Simplification interactive pour la production de textes adaptés aux personnes souffrant de troubles de la compréhension. In: Proceedings of TALN, poster session (2005)
14. OMG: Sbvr (2008), `http://www.omg.org/spec/SBVR/Current`
15. Omrane, N., Nazarenko, A., Rosina, P., Szulman, S., Westphal, C.: Lexicalized ontology for a business rules management platform: An automotive use case. In: Proceedings of the 5th International Symposium on Rules, International Business Rules Forum (RuleMF@BRF), Ft Lauderdale, Florida, USA (November 2011)
16. Ross, R.G.: Principles of the Business Rule Approach, ch. 8-12. Addison-Wesley, Boston (2003)
17. Siddharthan, A., Caius, G.: Syntactic simplification and text cohesion (2003)
18. Wagner, G., Lukichev, S., Fuchs, N.E., Spreeuwenberg, S.: First-version controlled english rule language. In: REWERSE IST 506779 Report I1-D2 (February 2005)