A Model-driven Regulatory Compliance Framework

Deepali Kholkar, Sagar Sunkle, Suman Roychoudhury and Vinay Kulkarni
Tata Consultancy Services Research, India

1 Introduction

Modern enterprises operate in an unprecedented regulatory environment [1]. Increasing regulation and heavy penalties on non-compliance have placed regulatory compliance among the topmost concerns of enterprises worldwide. Enterprises are increasingly looking to technology to aid their overall compliance process and efforts.

Industry uses GRC frameworks[1] for compliance management and tracking. These are document-oriented systems that help human experts maintain traceability between various artefacts in the compliance life-cycle. Documents such as legal text of regulations, compliance process descriptions, audit reports, etc. can be linked using tagging mechanisms. Actual implementation of compliance to regulations happens through organizational processes and IT systems. Therefore, GRC frameworks lacks the necessary end-to-end mechanism for automated compliance checking.

In this demonstration, we show a model-driven approach, where the entire process of interpreting natural-language (NL) regulations using information extraction (IE) / machine learning (ML) techniques to its final representation in formal rule specification language (DR-Prolog) is semi-automated. The transformation of rules from natural language text to DR-Prolog goes through a series of steps (1-3) as shown in Fig. 1, i.e., from constructing a domain model dictionary using open IE techniques to model authoring using SBVR[2] Structured English [2] (controlled English like context-free language) to SBVR model generation using model transformation techniques. Finally, DR-Prolog rules are generated from the SBVR model representation and relevant data is extracted from the enterprise data sources, which is then checked for compliance.

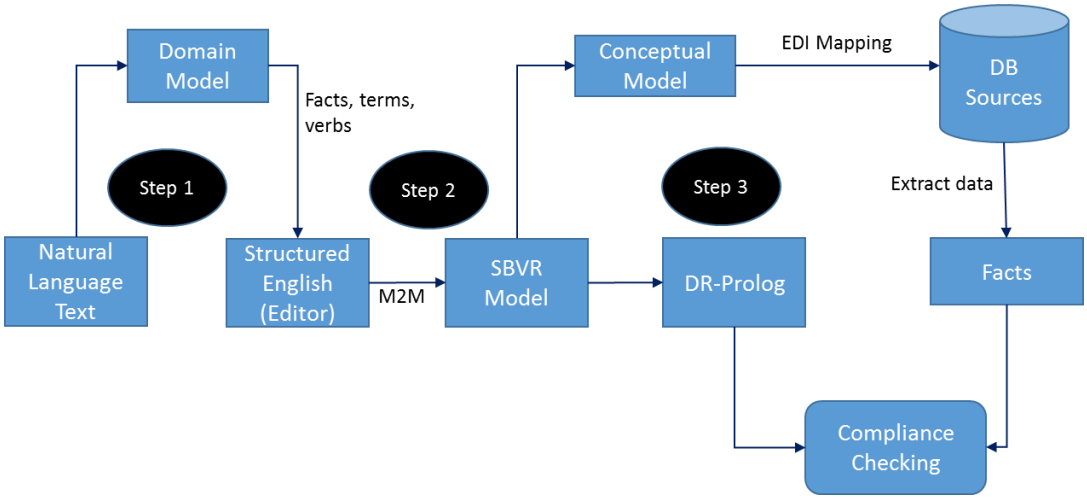

Fig 1. Block Diagram of End-to-End Model-Driven Regulatory Compliance Framework

2. Objective of this Demonstration

The main purpose of this demonstration is to: (1) Introduce our model-driven regulatory compliance framework. (2) Show how our framework can be applied to a real-world case study

[1] MetricStream, http://www.metricstream.com/
[2] Semantic Business Vocabulary and Rules

3. Model-Driven Regulatory Compliance Framework

In this section, we will explain the architecture in parts as per steps [1-3] as shown in Fig 1. First, we describe in Fig. 2, how relevant facts and domain vocabulary is extracted from natural language text of regulations using IE / ML techniques.

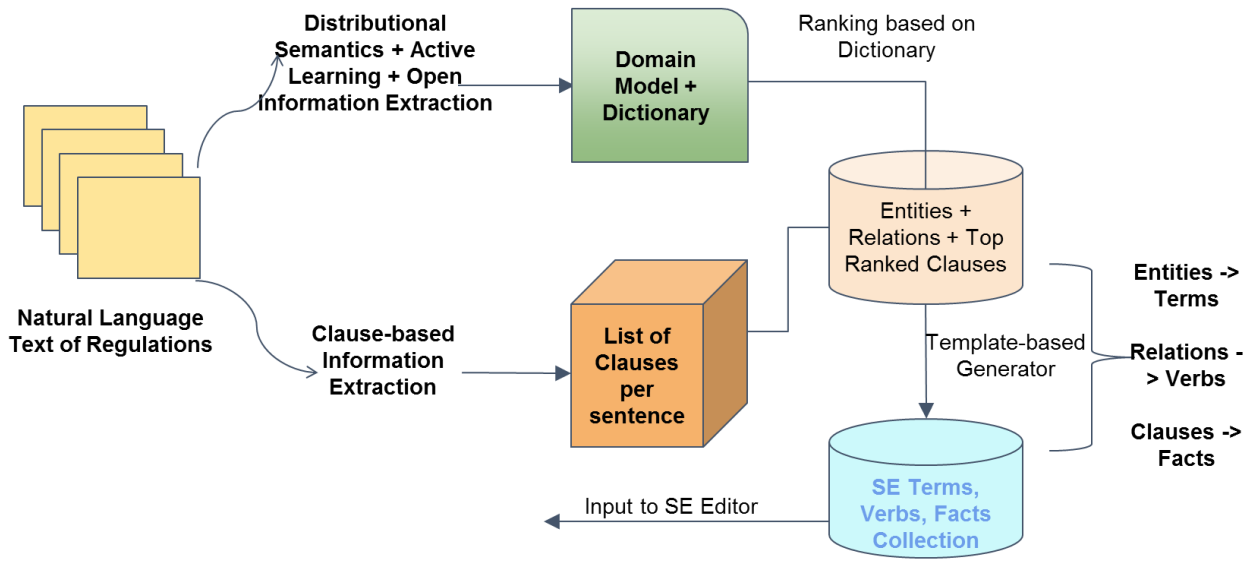

Fig 2. Natural Language to Structure English Construction using Information Extraction

Step 1

As the first step in the (semi-) automated regulatory compliance framework (see Fig 2), we enable the domain expert in modelling the regulatory domain (for a specific business domain) by providing an interactive domain model and dictionary generator. Additionally, we also provide an active learning based rule classifier which learns to separate the sentences in the regulatory text that contribute to a statement of regulatory rule from those that do not [3].

Since the syntax of the Structured English involves terms (domain model concepts and their mentions), verbs (relations extracted from the text), and facts (both ground facts and fact type/verb concept instances), we focus on generating possible facts for a given block of text using the terms and verbs. To this end, we use both the domain model and dictionary built earlier, as well as the rule classifier. The domain expert selects a block of text from the regulatory text which is to be translated to its SE counterpart. The rule classifier weeds out part of the block of text that does not contribute to regulatory rule. The remaining text is processed sentence by sentence using Clausie [4]. Clausie provides a domain independent, unsupervised triple (subject, relation, arguments) extraction using the syntactic information contained in the dependency parse of a sentence. From the triples that Clausie suggest, we choose those which conform to established relations from previous step. We use the domain dictionary to determine whether a triple matches previously found relation between domain concepts based on the occurrence of mentions of the concepts in the components of the triple. The triples so found are presented to the domain expert in the syntax of facts as used in the SE editor. We leave the specification or alteration to the suggested quantification and qualification to the domain expert.

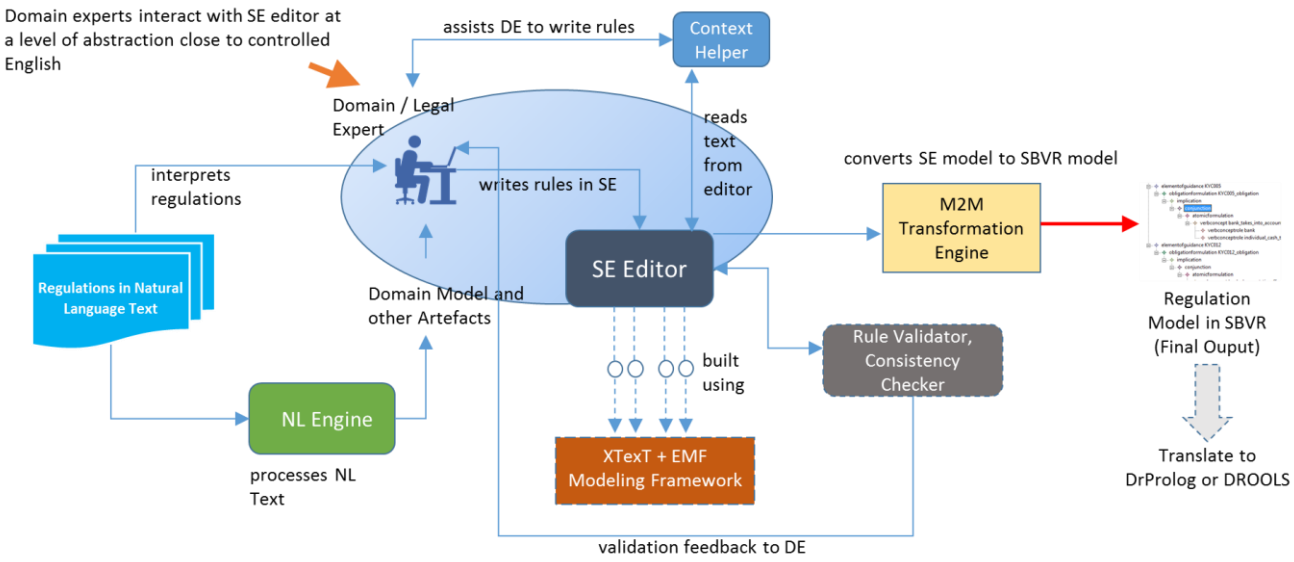

Fig 3. SE Model Authoring and SBVR Model Generation

Step 2

In second step of the framework (see Fig 3), we introduce Structured English (SE), which is a domain-specific language that enable domain experts to specify regulatory rules in a controlled English like language with inputs from step 1. SE is adapted from OMG SBVR SE specification (Appendix C) [2] and we have provided a context-free representation that captures key concepts like terms, facts, verbs, quantification, modality, quantifiers and rules. The language is developed in XTexT and internally uses ANTLR grammar specification with additional features like cross referencing provided by XTexT. The English like semantics of SE makes it amenable for domain experts to specify rules at a level of abstraction appropriate to them.

In addition to rule specification in SE, we also provide a text-to-model / model-to-model transformation facility that automatically transforms all the SE rules to an equivalent SBVR model representation as shown in Fig 3. In case of inconsistency in rules specified by the domain expert, a rule validator or consistency checker provides interactive feedback to her so as to catch errors at an early stage. Therefore the critical output of step 2 is the automatic creation of SBVR regulatory model that acts as input to step 3. This model can act as intermediate representation for not just translation to DR-Prolog but also to other rule systems like Drools, SWRL etc. Since SBVR is an industry standard one can use the generated regulatory model for integrating with other SBVR based systems or create purpose specific models (e.g., conceptual model) from it, which is discussed in the following step.

Step 3

In step 3 of our framework (see Fig. 4), we use the SBVR model from step 2 to generate both the rule base in logical form as well as for extraction of relevant enterprise data, on which the rules are to be checked.

We choose DR-Prolog as the language for creation of our rule base. Firstly, we create the metamodel map between SBVR and DR-Prolog meta-models, which is then used to translate to rules in DR-Prolog syntax.

The SBVR rule model being fact-oriented, captures the dependence of rules on fact types. These fact types denote the propositions whose truth value must be determined in order to evaluate whether the rule holds. The set of fact types therefore constitutes the necessary and sufficient *model* of information needed from the enterprise, for determining compliance, and is actually the *conceptual data model* of the regulation (see Fig 4). We programmatically extract the leaf level fact types from the SBVR model of rules to obtain the conceptual data model of the regulation.

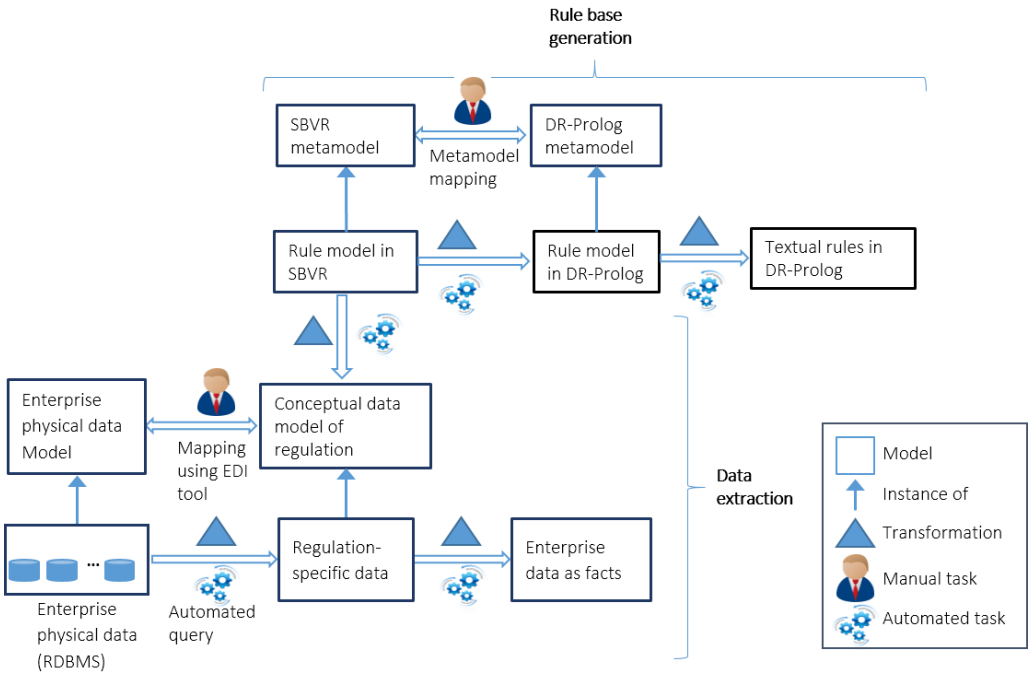

Fig 4. SBVR to Dr-Prolog generation, Conceptual Model Creation and Compliance Checking

Data relevant to a regulation is typically distributed across several enterprise systems. Therefore, we need an integrated view of relevant enterprise data [5] for data extraction. We use an in-house enterprise data integration (EDI) tool [6] that allows mapping of multiple physical database schemas to a single conceptual schema. Our conceptual schema of the regulation is imported into the EDI tool. Domain experts then map it to the distributed physical database schemas of the enterprise. We then generate queries on the conceptual data model, in an automated manner, that are translated by the EDI tool to queries on enterprise physical tables using the schema mapping, as depicted in Figure 4. Lastly, the translated queries on execution fetch the required data from enterprise databases, and checks for compliance against the generated DR-Prolog rules.

4. Case Study – Know Your Customer (KYC)

We will demonstrate the framework using Reserve Bank of India's KYC regulations. KYC regulations aim to prevent money laundering and financing of terrorism. They require the financial institutions like banks to take new customers following strict identity and address checks while transactions of existing customers need to be monitored based on their risk profiles. The various steps in the framework will be demonstrated using representative customer types identified in the KYC regulations. Fig. 5 shows the interactive domain model and dictionary generator [3].

The top-left box shows current set of clusters of sentences where known mentions of domain concepts are found. The top-right box shows the current dictionary, the middle-right box the current domain model, and the bottom-right box shows the clusters of sentences where no mentions of existing domain concepts are found. We also show these, so that the domain expert may choose to enter any of concept mentions pairs from this box, which she earlier missed out. The middle-left box is where she enters the mentions of existing concepts, new concepts, and unary/binary/ternary relations. The bottom-left box is used to show instructions and cautions such as a repeated attempt to add pre-existing concept or mention.

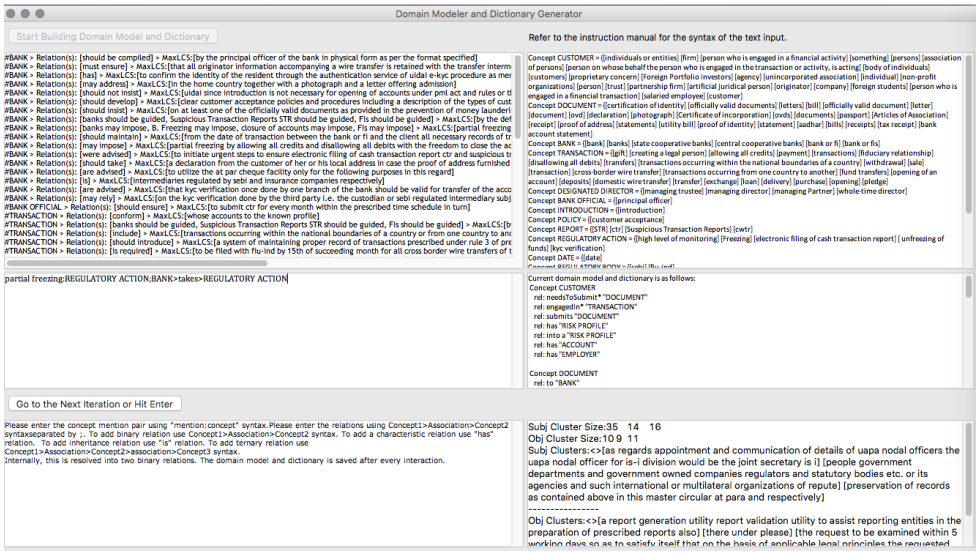

Fig 5. Domain Model and Dictionary Generator GUI

The domain model and dictionary, once the domain expert assesses them to be sufficiently exhaustive, are translated to SE syntax and along with the Clausie-generated and user (domain expert)-accepted facts. The following shows how the SE editor then comes into play.

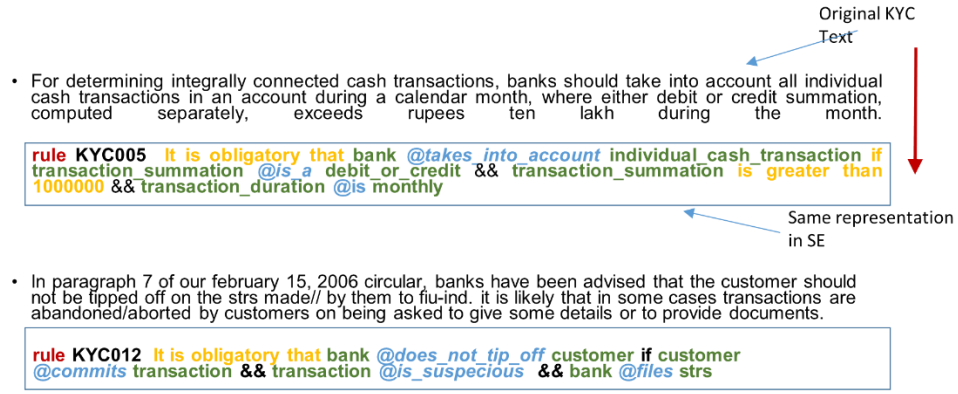

Fig. 6 Original KYC text and its equivalent representation in SE

In Fig 6, one can observe sample KYC regulations in plain English and its equivalent representation in Structured English. A snapshot of the SE editor is shown in Fig. 7. One may notice, how the SE language captures the modalities, quantifiers or implications in a controlled English like environment. Typically a rule in SE, consists of various facts (e.g., `bank @takes_into_account individual_cash_transactions`) that will be provided as input to the domain expert from the information extraction phase (as shown in Fig 5) as she gradually authors the rule base. Corresponding SBVR model (see Fig 8) will be automatically generated in the background that the domain expert is oblivious to but used by next part of the tool-chain to generate DR-Prolog (formal) rules and subsequent data extraction from Enterprise data sources.

Here, we provide some actual images (Figs. 7 & 8) of the SE Editor and corresponding SBVR models generated from SE using model transformation techniques.

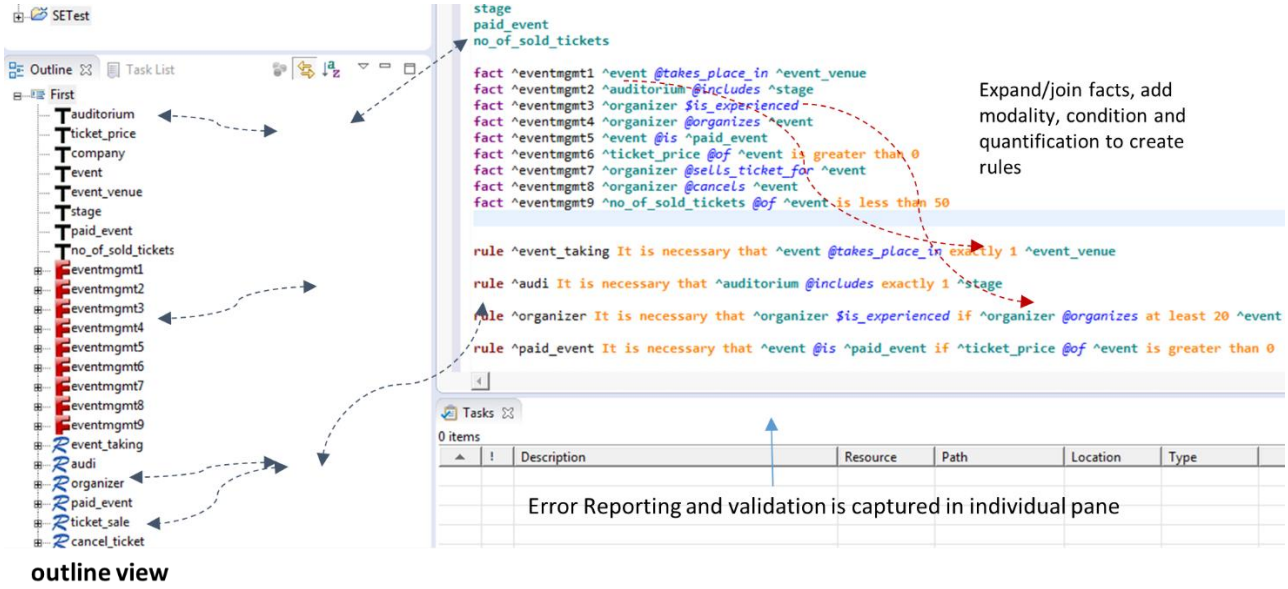

Fig 7. Structured English Model Authoring Editor

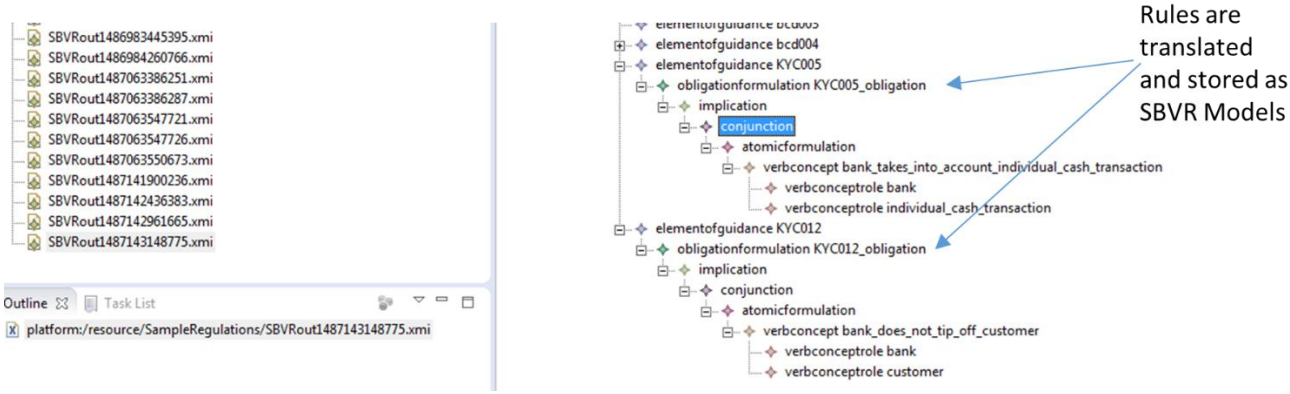

Fig 8. SBVR Model automatically generated from SE

The complete (semi-) automated regulatory compliance framework including DR-Prolog rule generation from SBVR model, conceptual model creation, data extraction from EDI and compliance checking will be demonstrated in the light of KYC regulations but omitted here for brevity.

5. Conclusion

In this demo, we will demonstrate our end-to-end (semi-) automated regulatory compliance framework and the complete tool-chain starting from natural language representation of rules to its corresponding representation in structured english, followed by translation of SE to SBVR models and finally to formal logic specification. Models were used in every stage of the framework, starting from domain dictionary model used in IE phase, SE SBVR model in model authoring phase and conceptual data model in formal rule generation /data extraction phase. All these models works in synergy and is represented at a level of abstraction suitable for a specific purpose to address the overall goal of creating an automated regulatory compliance framework.

6. References

1. Reuters, T.: State of Regulatory Reform 2016 - A Special Report. (2016)
2. OMG. 2008. Semantics of Business vocabulary and Rules (SBVR), OMG Standard, v. 1.0.
3. Sagar Sunkle, Deepali Kholkar, Vinay Kulkarni: Informed Active Learning to Aid Domain Experts in Modelling Compliance. EDOC 2016: 1-10
4. Luciano Del Corro, Rainer Gemulla: ClausIE: clause-based open information extraction. 355-366
5. Deepali Kholkar, Sagar Sunkle, Vinay Kulkarni: From Natural-language Regulations to Enterprise Data using Knowledge Representation and Model Transformations. ICSOFT-PT 2016: 60-71
6. Raghavendra Reddy Yeddula, Prasenjit Das, Sreedhar Reddy: A Model-Driven Approach to Enterprise Data Migration. CAiSE 2015: 230-243