Automating Financial Regulatory Compliance Using Ontology+Rules and Sunflower

Reginald Ford
SRI International
Menlo Park, California
reginald.ford@sri.com

Grit Denker
SRI International
Menlo Park, California
grit.denker@sri.com

Daniel Elenius
SRI International
Menlo Park, California
daniel.elenius@sri.com

Wesley Moore
Quarule Inc.
Charlotte, North Carolina
wesley.moore@quarule.com

Elie Abi-Lahoud
Quarule Inc.
Dublin, Ireland
elie.abilahoud@quarule.com

ABSTRACT

Compliance departments in the international finance industry are struggling to use traditional methods to keep up with the demands of new and more stringent regulatory and policy requirements. One initiative supported by many institutions is definition of a common Financial Industry Business Ontology (FIBO). We regard a common ontology as an important step, but in order to support real-world uses cases, the ontology needs to be augmented, and further supplemented by *rules* that encode the meaning of regulations and policies. We use Sunflower, which is built on top of the Flora-2 knowledge representation languages and reasoner, to add automation to the compliance lifecycle. Sunflower is domain-agnostic, and financial regulatory compliance is one of its many application areas.

CCS Concepts

•Information systems → Expert systems; •Applied computing → Business rules; Enterprise ontologies, taxonomies and vocabularies; •Computing methodologies → *Logic programming and answer set programming;*

Keywords

Ontologies, Rules, Sunflower, Integrated Development Environment, Regulatory Compliance, Reasoning, Explanation

1. INTRODUCTION

Compliance departments in the financial industry are hard-pressed to keep up with both government regulations and internal controls that are intended to improve the transparency and safety of enterprise business practices. Although supported by corporate IT systems and vendor services, traditionally these departments have depended heavily on manual and cognitively intensive operations (e.g., using spreadsheets) to answer, for themselves and for auditors, this question: "Do we know that we are following the rules?" Given the financial crisis of 2007-08 and the size of fines levied since then, failure to provide an affirmative answer can yield severe consequences.

One of the challenges faced by regulators and regulatees alike is creating a common and well-understood vocabulary of finance concepts. To this end, the Enterprise Data Management (EDM) Council and the Object Management Group (OMG) is developing an industry-wide Financial Industry Business Ontology (FIBO), which is expressed using Web Ontology Language (OWL) [3].

The approach to regulatory compliance described in this paper uses ontologies and semantic rules to support reasoning over financial transaction data. We participated in a working group to show that an approach based on rules and ontology could leverage FIBO for developing automated analysis and control systems that are more comprehensible to financial industry professionals, and more flexible and adaptable for developers, than traditional procedural software applications [9, 7]. SRI International and Quarule have continued to build on this foundation [14, 19].

2. THE CORE WORK OF COMPLIANCE

Companies in regulated industries have a duty to develop a program of compliance. Examples in the financial industry include the US Securities and Exchange Commission final rule on Compliance Programs of Investment Companies and Investment Advisers [16] or the Federal Reserve Board policy letters on Compliance Risk Management Programs and Oversight at Large Banking Organizations with Complex Compliance Profiles [6]. Figure 1 shows main activities in a compliance program.

Through the program of compliance, a company assures both itself and its regulators that the company is:

- Following all regulations that apply;
- Identifying regulatory violations if they occur;
- Promptly correcting violations and their root causes; and
- Validating the sufficiency and effectiveness of policies, procedures, and controls.

SEMANTiCS 2016 September 12 -15, 2016, Leipzig, Germany

DOI: 10.1145/1235

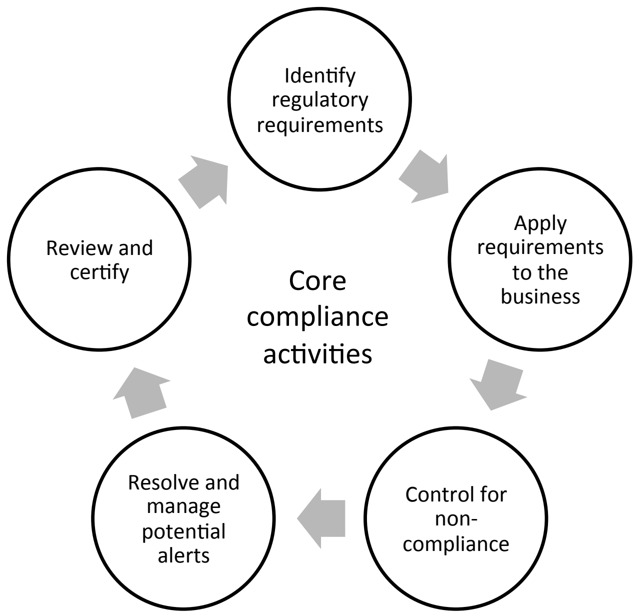

Figure 1: Core compliance activities

The first step towards achieving a valid compliance program is to identify all regulatory requirements applicable to the business. This is a function of business footprint, including its activities, products, markets, and jurisdictions.

The regulations are applied to business activities, products, people, legal entities, and business units through policies and procedures. Policies and procedures reflect regulatory requirements, how they apply to the business, and how the business fulfills the requirements. They filter and summarize the regulation as it applies to the business and the work of people in different aspects of the business. Companies are held accountable by regulators to have adequate policies and procedures, and evidence that they comply with them.

For every key requirement of policies and procedures, controls are developed and executed to identify non-compliance. Controls may take the form of monitoring, surveillance, or testing, but also include training, ad hoc investigations, and early warning systems. The choice of control type will depend on best regulatory practices, specific guidance from regulations or policies, and available resources.

Controls identify likely compliance violations as alerts. When alerts are discovered, critical tasks include managing their resolution, monitoring the time needed to resolve an alert, and increasing the rate of successful resolutions to problems.

High performing compliance programs focus on identifying and fixing as many problems as possible before any regulators or customers encounter them. Resolving the root cause of compliance alerts may be simple or very complex and costly. Alerts may identify one-time lapses in compliance that can be resolved easily with a correcting transaction, or they may identify systemic non-compliance. Fixing problems takes resources. Prompt identification and resolution of compliance problems save the enormous expense of enforcement actions and litigation. Finding a problem but failing to fix it increases potential fines and erodes the compliance culture within an organization. The rate of resolution of problems is a key performance metric of a compliance program.

Finally, regulations or policies typically require that a compliance program be validated and then certified by senior management for regulators. The process of validation and certification depends on a complete end-to-end review of the entire compliance program to periodically confirm that the program covers all regulatory requirements, that the program is adequate, and that the controls within the program are effective.

3. SUITABILITY AND BENEFITS OF SEMANTIC TECHNOLOGIES

Complex concepts and relationships in the finance industry are mirrored by complex regulations in the financial compliance domain. For example, the status of one bank as an affiliate of another bank is a highly complex relationship defined through many other concepts such as subsidiary and majority ownership of stocks. Regulations define complicated rule sets and constraints that cover detailed cases, and also the exemptions when these regulations do not apply. The difficulty of assessing compliance is compounded by the huge amount of data that needs to be checked. Depending on the type of transactions, banks may have tens of thousands to millions of transactions per day. Semantic technology approaches provide automated reasoning and inferencing tools that can infer new facts from existing facts and data. This inferential capability proves particularly powerful when the amount of data or the relationships and constraints among data are so complex that it is impossible for humans to understand and reliably reason about them. Therefore, semantic technology is a very good fit for applications in the financial compliance domain.

In the financial domain, while there are many sources of structured data (e.g., spreadsheets, SQL databases, and so on) that need to be integrated in an analysis tool, unstructured data is also relevant for some regulations (e.g., phone call recordings, email, and so on). Because solutions must adapt to changes in needs and regulations, the data models used in analysis and compliance reasoning must be extensible. More generally speaking, applications that evolve over time need solutions for flexible and transparent data management. This requirement applies to applications where it is impossible to identify and define *all* kinds of data ahead of time, such as virtually any knowledge management system, systems supporting research, or applications that use a large amount of unstructured and unpredictable data. Financial compliance applications fall under this category.

Semantic technologies are a great fit for evolving applications because they provide an abstraction mechanism that shelters the applications from changes in underlying storage formats (e.g., changes in relational databases or adding new types of data storages). Solutions based on semantic technology make these applications more robust against changes in the sense that such changes do not require reengineering the system from scratch, but rather extension of the underlying ontologies and rules. Analysis queries will, if properly designed, be forward compatible with the extended semantic model. Ontologies also provide the abstraction layer required to make the overall solution storage agnostic and, thus, provide the basis for achieving better scalability when there is a need to adapt to changes in storage formats. Rather than replacing legacy data storage formats, semantic technology approaches work in unison with them.

Since semantic technology approaches can operate with various datatypes and formats and their associated tools, they facilitate interoperability and overcome information silos. The abstraction layer of semantic model representations can be understood as a kind of canonical form at an abstraction level that is higher than that of the specific data storage format. Converters can move canonical information to other external forms for use by existing external tools. Similarly, information can be readily converted from external into internal canonical forms. Semantic technology approaches aim to keep the internal canonical form as generic as possible, so that tools for manipulation and reasoning about the information are reusable and more robust to changes in the underlying specifics of the data.

The expressiveness of semantic technology implies that there is inference and reasoning built into the language. Many semantic tools provide appropriate reasoning engines. For example, rather than explicitly stating relationships between different data elements, one could apply the concepts of inheritance and transitivity of inheritance to a small set of inheritance relationships defined on classes to infer relationships among an arbitrarily large number of data items. Using the built-in semantic reasoning over concepts such as inheritance and cardinality constraints among others means that rich capabilities for analyzing are readily available for applications that are built using a semantic technology approach. As a result, analysis, reporting or querying across multiple data sources can be implemented faster and more efficiently with solutions that are based on semantic technology than reimplementing basic reasoning and inferencing capabilities redundantly in procedural languages for each application.

Semantic technology provides more declarative representation and reasoning and a higher abstraction level than conventional procedural programming languages. In semantic technology, reasoning is done over declarative knowledge (the "what") such as ontologies with classes and properties, rules (including mathematical ones) and a knowledge base of instances of these classes. Conventional programming with procedural or object-oriented programming languages is done at the lower abstraction level of imperative solutions (the "how").

In summary, semantic technologies provide a conceptual and logical foundation to the modeling and interrelation of information. As such, semantic technologies provide an infrastructural means on which to build applications. The challenges of financial compliance applications are particularly well-suited to semantic technology approaches. The benefits of such approaches will result in applications that can be more easily adapted and extended as more regulations are covered or new data sources are integrated into the reasoning and decision processes.

4. SUNFLOWER OVERVIEW

Ontologies provide a conceptual framework for modeling the financial domain, but when it comes to modeling complex policies and regulations, system dynamics, and behaviors, we need more expressiveness in the form of *rules*. Existing languages like OWL, SWRL, and RIF, and associated editing and reasoning systems, do not support many of the features required for modeling the financial domain. For example, SWRL does not not support n-ary predicates, aggregation or higher order expressions, structured output (such as CSV or XML), database interfaces, or tracing or debugging of reasoning with rules.

The Sunflower [19] suite is intended to fill this gap. Figure 4 shows the main components of the tool suite. Sunflower is a set of libraries and tools based on the Flora-2 language[1], which in turn is implemented as a layer on top of XSB[2].

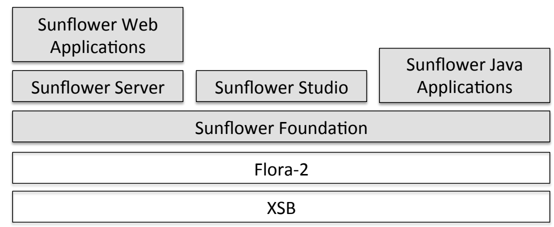

Figure 2: The main components of the Sunflower stack. Components developed at SRI are shaded.

Flora-2 is a knowledge representation language and associated reasoning engine developed and maintained primarily by Michael Kifer at Coherent Knowledge Systems. While Flora-2 has its origins in the *logic programming* research community, OWL has its root in *description logics*. Flora-2 is highly expressive. It supports, among other things, n-ary formulas, negation-as-failure, aggregation, higher-order predicates, functions, frame syntax for classes and instances, infix mathematical expressions, prioritized or default rules, and knowledge base update operators. Flora-2 integrates ontologies and rules in a powerful way.

On top of Flora-2, *Sunflower Foundation* is a library, implemented mostly in Java and partially in C/C++ and Flora itself, which provides many features that are essential to building applications based on Flora rules and ontologies. These features include a Flora parser that generates a detailed syntactic representation of Flora content in Java, syntactic manipulation of that representation, a higher-level ontology model, importers and exporters for other languages (RDF, OWL, SWRL, CSV, SQL, etc), an interface to the Flora reasoner, a live RDF triple store connector, an explanation module that produces structured explanations of reasoning results to the user, and a natural language module that produces English paraphrases of reasoning results and explanations.

Sunflower Studio is an Eclipse-based IDE for working with Flora-2 content. It uses Sunflower Foundation, and exposes most of its functionality through various user interfaces. It provides a tightly integrated and synchronized suite of tools for ontology and rule understanding, editing, and validation. The tools include editors for ontologies, properties and instances and rules; search capabilities; graph views showing rule dependencies, class hierarchies, knowledge base relations, and import hierarchies; query user interfaces that allow editing and executing queries; query results interfaces; and a test framework.

[1]http://flora.sourceforge.net/
[2]http://xsb.sourceforge.net/

Sunflower Server is a Web server that exposes much of the Sunflower Foundation functionalities over HTTP using REST APIs. The API is documented using Swagger[3], which allows the automatic generation of client libraries for many different programming languages[4]. We have used this to generate both Java and JavaScript clients for different applications. The JavaScript client is used in a proof-of-concept Web front-end for Sunflower that allows editing and querying of Flora ontologies.

We have used Sunflower as a basis for automated reasoning applications for systems-of-systems interoperability for military training and testing, military acquisition systems, and design tradeoff analysis for space system architectures. This paper describes how we use it to represent and reason about financial regulatory compliance.

Sunflower capabilities that are particularly salient and attractive for the finance domain include: (a) the ability to provide explanations in English language of analysis results and (b) the ability to import data from various structured data sources.

The explanation functionality is especially important since compliance analysis often results in false positives. Current black-box compliance software does not allow compliance officers to drill down into the reasons why a transaction is evaluated as compliant or non-compliant. One of the main requirements for financial compliance applications is not only automatic determination of a transaction's compliance, but also to reveal the reasons why a transaction showed up in the alert list of possible non-compliant transactions, or similarly to check why certain transactions are not showing up as non-compliant. Flora-2's meta-programming features facilitated the implementation of a tracing capability and an English paraphrasing functionality within Sunflower that allow users to drill into the details of analysis results. The functionality even paraphrases the proof logic of the analysis in English so that it is easier to understand for those who are not experts in semantics.

The ability to ingest data from various other structured data sources, including MySQL databases, CSV files, spreadsheets, and RDF triple stores is also important for financial applications because much of the data is currently stored in spreadsheets or relational databases. Sunflower also provides means to export query results or knowledge base (KB) content to formats such as tables, spreadsheets or OWL ontologies.

The breadth of tools and functionalities in Sunflower combined with the flexibility of deploying Sunflower as standalone tool or via its Web API as part of enterprise applications makes it unique among existing semantic technology tools.

5. AUTOMATING THE COMPLIANCE LIFE CYCLE

Figure 3 depicts our approach to automating compliance. At the center is the Sunflower-based analysis engine that will reason about whether transactions and related activities performed by a bank are compliant with regulations and policies. Two main inputs are necessary for the analysis engine:

[3]http://swagger.io

[4]e.g., using https://github.com/swagger-api/swagger-codegen

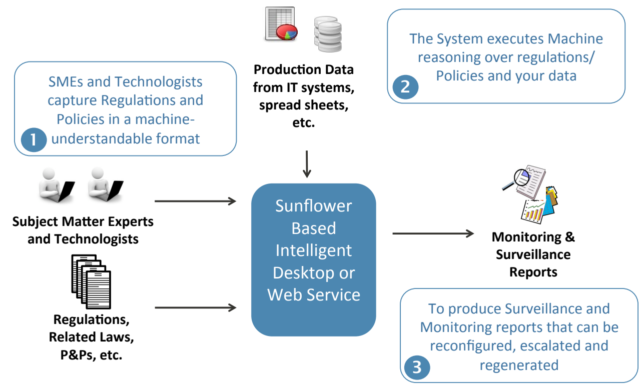

Figure 3: An Approach to Automating Compliance

1. machine-readable encodings of the regulations and policies, and

2. machine-readable transaction data that is represented in a vocabulary that is shared with item 1.

Semantic Technology Experts (STEs) must work together with Subject Matter Experts (SMEs) to capture the regulations and policies as the machine-readable rules in item 1. The rules are expressed over a domain-specific vocabulary that is represented as a set of ontologies. For the domain of regulatory compliance, we started with the FIBO ontology and extended it with concepts needed to express use cases such as Regulation W [21]. Extensions included concepts to represent detailed transaction properties (e.g., date and amount of transaction), classes to capture organization and their properties, and assets and their properties.

The task of capturing regulations as rules is currently performed manually, but we envision that template-based rule editors will support this task in the future. A rule editor for financial compliance would also make use of a financial ontology as a means to auto-populate menus so that users can choose among the defined concepts when filling out rule templates. Section 6 provides examples of rules and ontological concepts for the Regulation W use case.

Existing data sources and their specific format must be connected to the vocabulary defined in the ontologies to provide the second input in the list above. For example, assume that transaction data is stored in one big relational database table named "Transaction", and assume that this table has columns for "transaction id, date, and amount" as well as columns for "asset name, asset properties ...." and so on. In the ontology, the database model is mirrored by the definition of classes for `Transaction` and `Asset`.

If the analysis engine needs to reason about existing data, we need to ingest or translate that data in a format that is compliant with the concepts in the ontology. Using the example above, part of the existing database table "Transaction" will be translated into instances of the `Transaction` class, and other parts will be translated into instances of the `Asset` class. This mapping from a data source format to the ontology needs to be performed only once. Once the mapping is known, it is easy to write software to ingest the instance data into the KB.

It is important to note that Sunflower does not require the data to be ingested into the KB. Instead, it can provide connectors to existing data stores such as SQL databases or RDF triple stores. Sunflower then queries the stores to access the data it needs for reasoning. This is important for application domains like financial regulatory compliance that involve a huge amount of data. The hundreds of thousands to millions of transactions performed by banks each day are a volume of data that cannot be feasibly ingested into a semantic technology reasoner. For example, we have worked with a bank to apply our Regulation W analysis technology to a large volume of data stored in a relational database. Using the connectors to MySQL, we performed automated analysis on several days' real data that amounted to about 1.5M transactions.

The Sunflower-based analysis engine takes the regulations and policies expressed as rules, and either ingests data or connects to existing data stores and provides various analysis queries for the end user. A typical query for a given regulation would ask: "What transactions are compliant with the regulation?" Such a query can be specialized to look only at transactions in a certain time frame, or at specific (sets of) transactions, at a certain (set of) regulations, and so on.

The output of our reasoner, a list of transactions that are and are not compliant together with explanations for every compliance decision, can be used as a basis for further alert systems or reports.

6. REGULATION-W EXAMPLE

The US Federal Reserve Act Regulation W (RegW) established requirements for transactions between banks and their affiliates. This section illustrates how Sunflower can automate efforts to analyze whether particular financial transactions are compliant with RegW. Figure 4 summarizes RegW and the questions that need to be answered. For example, is the transaction reportable (i.e., "covered")? Is it permitted (i.e., up to 10% of capital stock and surplus worth with one affiliate, and 20% combined with all affiliates)? Answering these questions requires some other computations, such as the identification of affiliates, and whether or not the asset to be traded is "'investment grade."

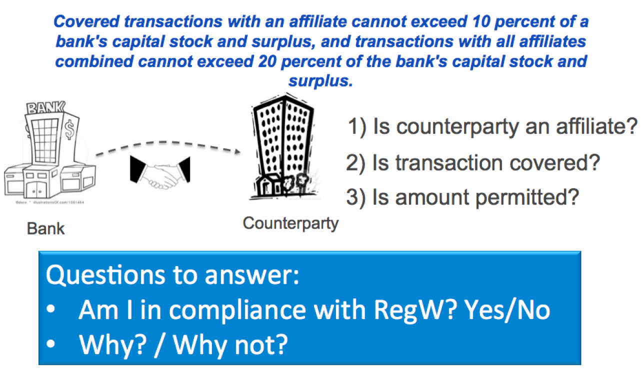

Figure 4: Main Regulation-W concepts

Under the auspices of a FIBO working group, we worked together with financial industry subject matter experts to capture the regulations initially as a semi-formal specification of concepts, relationships, and use cases. Table 1 shows two of the seventeen "scenarios," which define reasons why a transaction might be reportable under RegW or is exempt from reporting.

We then formalized the specification of scenarios as a system of Flora-2 rules. Figure 5 shows dependencies of the selected `reportableAndWhy` rule predicate on other predicates. For example, the `issuedByAffiliate` predicate determines whether an asset was issued by an affiliate, a fact which is relevant for scenarios 1 and 4 in Table 1.

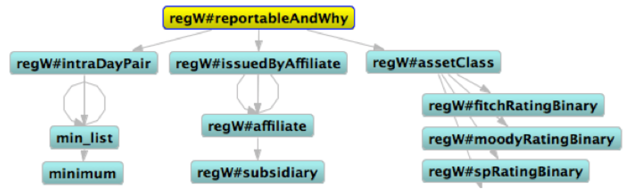

Figure 5: "Reportable and Why" rule dependencies.

The `reportableAndWhy` predicate is associated with several rule bodies that represent different use case scenarios. For example, the rule below defines a case where a transaction is not reportable because it satisfies the `Ready Market` exemption criteria. It is `Ready Market` because the buyer is an affiliate, and the issuer is not; the asset is on the secondary market (i.e., `free to trade`), and is `actively traded`; and the evaluations assigned by rating agencies classify the asset as `corporate debt`.

```
@!{ReportableRule_Scenario1_ReadyMarket}
regW#reportableAndWhy
(?Transaction, 'N', 'Ready Market', regW#Scenario1) :-
  ?Transaction [
    regW#firm -> ?Firm,
    regW#asset -> ?Asset,
    regW#capacity -> regW#Principal,
    regW#counterParty -> ?Counterparty ],
    ?Firm [ regW#affiliate -> ?Counterparty ],
   \+(regW#issuedByAffiliate(?Asset,?Firm)),
  ?Asset [
    regW#freeToTrade -> 'Y',
    regW#activelyTraded -> 'Y',
    regW#assetClass -> regW#CORPDBT ].
```

For the initial demonstration of reasoning over RegW rules, we imported data on transactions, assets, and firms from spreadsheets. To scale up to analyzing millions of transactions, a subsequent demonstration applied the rules to data stored in a MySQL database. The spreadsheet import and SQL connector operations associated the data with FIBO classes and properties, or extensions to FIBO that we defined to capture concepts at the level of granularity required to support reasoning about RegW compliance. Note that the example in this paper uses fictionary data rather than real data from our banking partner.

To find out which transactions are reportable, and why, we can directly query the `reportableAndWhy` predicate. This produces query results as in Figure 6 (as shown in the Sunflower query results UI). For example, Regulation W permits transaction `B5634_0117_02` and does not require it to be reported because it is `Ready Market`.

The Sunflower tracing capability allows compliance officers and auditors to understand the reasons *why* a particular result was returned. The explanation for any query result is produced by clicking the '?' next to an individual result. The trace for the second result in Figure 6 is shown in Figure 7. Traces can be displayed as graph, a tree expressed in Flora-2 syntax, or a tree expressed in English. The traces are

Scenario #	1	4
Description	Bank purchases 3rd party issued securities directly from an affiliate in the secondary market	Bank purchases affiliate issued securities from an affiliate in the primary market
Type of covered transaction	Asset purchase from an affiliate	Purchase of, or an investment in securities issued by an affiliate
Possible exemptions	Intraday / Ready Market / Riskless Principal	Intraday
Against capital limit/ collateral required	Capital	Capital / Collateral

Table 1: Sample scenarios for reportable/exempt transactions

?Transaction	?Reportable	?Reason	
regW#B5634_0117_01	Y	'Asset Purchase from an Affiliate'	?
regW#B5634_0117_02	N	'Ready Market'	?
regW#B5634_0117_03	N	IntraDay	?
regW#B5634_0117_04	N	IntraDay	?
regW#B5634_0117_05	N	'Riskless Principal'	?
regW#B5634_0117_06	Y	'Issued by an affiliate'	?
regW#B5634_0117_07	N	IntraDay	?

Figure 6: Results for query "Which transactions are reportable and why?"

proof trees. Each node in tree contains a Flora-2 expression, a justification (to the right of the vertical bar), and optionally one or more children. The most common justifications are "fact", i.e., the Flora-2 expression was directly derived from a fact (the node will have no children in this case), or it will mention a specific rule that was used to derive the expression. In this case, there will be one or more child nodes, which corresponds to clauses in the body of the rule. These, in turn, may be derived through other facts or rules, and so on. Clicking on a rule name in the trace causes the definition of the rule to be shown. It is also to include hyperlinks to online descriptions of the rules in the justification (e.g., to the authoritative definition of the regulation itself). The tracing functionality is implemented as a meta-interpreter (i.e., in the Flora-2 language itself), along the lines described in classical Prolog text books [18], and succeeds our previous work in [4].

The English trace is the most understandable to ordinary users. It is produced through a combination of default rules for different syntactic forms, and user annotations, all written in the Flora-2 language itself. For example, for a Flora-2 expression,

```
regW#B5634_0117_02 [ regW#firm -> "ABC Bank"^^\string ]
```

we get a default English version, "the firm of B5634_0117_02 is ABC Bank". This is improved by adding a `english` annotation on the `regW#firm` property:

```
regW#firm [ english ->
  'the firm engaged in transaction "%1" is "%2"' ].
```

The resulting English text is shown on the third line from the top in Figure 7. The arguments of the property are slotted in at the locations indicated by `%1` and `%2`. Other types of user annotations are possible, and they are combined with each other, and with the default rules, in a recursive way, which allows us to produce reasonable English traces for a relatively small amount of work.

The example trace shows the complexity of the reasoning required for determining why Regulation W does not require transaction `B5634_0117_2` to be reported. The second line of the trace shows that the result is obtained by applying `ReportableRule_Scenario1_ReadyMarket`. The two parties in the transaction (i.e., the firm is `ABC Bank` and counterparty is `Affiliate 1 of ABC Bank`), the traded asset, and properties of the asset (such as `free to trade` and `actively traded`) are facts found in the KB. Application of the `AssetClass-Rule_IG` (i.e., "investment grade") identifies the asset class of the traded asset as `corporate debt`. An analyst who wants to know why the asset was found to be investment grade can click the expansion triangle and follow the trace into details of the rating agency evaluations. The counterparty is determined to be an affiliate of the firm by applying the rule for `SubsidiaryOfAffiliateIsAffiliate`. An analyst who is curious about the details of subsidiary relationships can click the expansion triangle and follow the trace. Expanding the last line shows why the issuer of the asset is identified as a non-affiliate of the `ABC Bank`.

The results of a query can be saved and subsequently compared with the new results that are obtained after a fact or rule is changed. Figure 8 shows what happens after changing the Moody rating of the `CVS_Care` asset from `Baa1` to `BBB+`. A new result is shown in boldface, and the removed result is grayed out. The transaction goes from not reportable to reportable.

?Transaction	?Reportable	?Reason	
regW#B5634_0117_01	Y	'Asset Purchase from an Affiliate'	?
regW#B5634_0117_02	**Y**	**'Asset Purchase from an Affiliate'**	?
regW#B5634_0117_02	*N*	*'Ready Market'*	?
regW#B5634_0117_03	N	IntraDay	?

Figure 8: Comparing results

We developed the example described in this section as a proof of concept that FIBO, Flora-2 and Sunflower could help automate essential regulatory compliance activities described in Section 2. This preliminary step, which explicitly captured the regulations in a business context as a set of scenario cases, was an exacting task that relied on compliance experts to explain their explicit and implicit knowledge in a way that semantic technology experts could understand and formulate as rules. Much of the implicit knowledge was teased out as part of the interaction. Similarly, incumbent on the semantic technology experts was the responsibility to make ontologies and rules that were intelligible to financial and regulatory domain experts. To this end, we benefited from the declaratory character of ontologies and Flora-2 rules, and the Sunflower suite of tools. In particular, tracing was used as a test tool to validate the rules, and also as a tutorial tool to explain the rules by example. It also helped

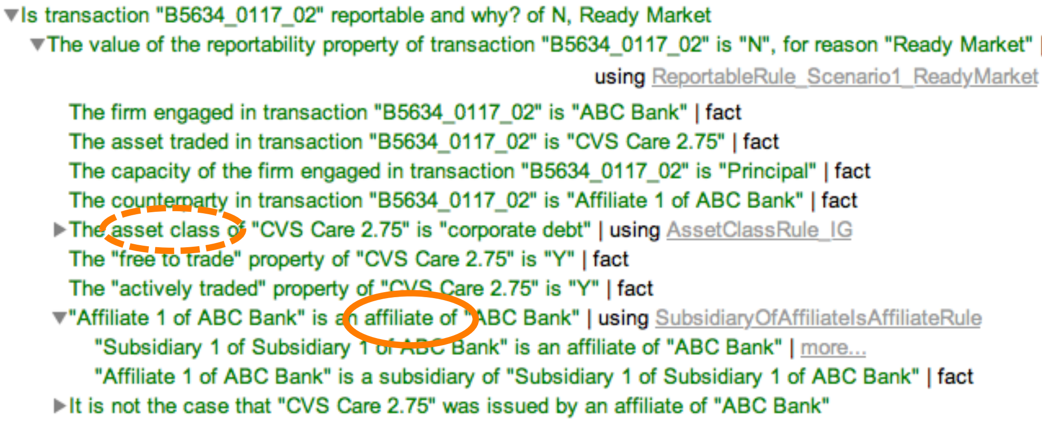

Figure 7: English explanation for the query "Why is a transaction permitted or reportable?"

that the Flora-2 syntax and expressiveness enabled creation of a rule base that we believe is more compact (approximately 750 lines), intelligible, and maintainable than could be achieved using a procedural programming language, a combination of OWL and SWRL, a combination of RDF and SPARQL, RIF, or the like.

We subsequently developed additional regulatory compliance rule sets to detect several types of undesirable trading practices. One such practice is brokers "churning" an account by engaging in trading that may be considered excessive relative to the investment objectives of the account and the risk tolerance of the client. Another is "wash trading," in which the same financial instruments are simultaneously (or near-simultaneously) bought and sold, for example, to generate broker fees without benefit to a customer or to inflate the perception that an instrument is in demand by artificially boosting trading volume.

7. RELATED WORK

Some of the research context for this paper has already been mentioned in previous sections. Section 1 cites research related to FIBO as a foundation for regulatory compliance applications. Section 4 cites semantic technology languages and software tools.

See [8] for an influential early paper on combining rules and ontologies, in particular the intersection of Description Logic Programs (DLP) and the Description Horn Logic (DHL) fragment of first-order logic (FOL). Prior to adopting the Flora-2 language in 2012, we based our ontology and rules work on a combination of OWL and SWRL, as exemplified in [17, 15, 5]. SWRL is a W3C member submission specified in [20]. See [4] for a description of SWRL-IQ, a tool that extends SWRL expressiveness.

Semantics of Business Vocabulary and Rules (SBVR) is an OMG standard used for natural language formalization of rules, as in [1] for example. Although we did not ultimately adopt SBVR as one of our core technologies, the FIBO working group experimented with using it as a language for capturing the knowledge of subject matter experts, a preliminary step to defining rules in Flora-2.

Other approaches to modeling regulatory compliance and applying ontologies and rules have been suggested. Compliance Representation Language (CoReL) is a domain specific language described in [11], and a survey of Frameworks for regulatory compliance management is presented in [10]. Application of the OntoBacen ontology, SBVR, OWL, and SWRL to financial system risk management is explored in [13]. An approach to context-adaptive rules for smart building configuration is described in [12]. A semantic methodology for specifying building construction regulatory compliance checking system, including SWRL rules, is described in [2].

8. CONCLUSIONS AND FUTURE WORK

This paper describes an approach that uses semantic technologies to automate much of the analysis work that finance companies must perform to be compliant with an increasing volume of regulations, and to demonstrate compliance to corporate executives and government auditors. Section 2 explains what is required of a high-performing compliance program. Section 3 summarizes the rationale for applying semantic technology in this domain. Section 4 gives an overview of the Sunflower semantic tool suite. Section 5 summarizes our approach to using ontology and rules to determine whether projected or actual trades are compliant with regulations and accepted business practice. Section 6 describes an extensive proof of concept demonstration. The success of the demonstration informed the decision of two of the authors to found Quarule, Inc., and the SRI and Quarule teams to continue their collaboration.

The current and future research interests of the authors include advanced tools and presentation methods to improve the understandability and maintainability of the rules, ontologies, and reasoning results. One of SRI's main challenges is to enhance Sunflower with general tools such as rule editors that make it possible for subject matter experts and end-users to do much of the work on their own, and depend less on semantic technology experts.

Quarule's complementary emphasis is to make semantic reasoning and explanation capabilities accessible to end-users in their everyday workflows, including integrated alerting and reporting functions. One of the main design and implementation challenges is to keep the learning curve for end-users within reasonable bounds.

9. ACKNOWLEDGMENTS

The work described in this paper is the result of collaborative and individual research and development efforts by SRI International and Quarule, Inc. The RegW example incorporates information provided by members of a FIBO working group, including Wells Fargo experts in regulatory compliance.

Development of Sunflower was funded in part by the U.S. Office of the Assistant Secretary of Defense for Readiness under the Open Netcentric Interoperability for Training and Testing (ONISTT) project, and by the TRMC (Test Resource Management Center) T&E/S&T (Test and Evaluation/Science and Technology) Program under the NST Test Technology Area. We are also indebted to the research community for developing and maintaining the open source language and software components on which Sunflower depends, especially Flora-2 (a.k.a. Ergo Lite), XSB Prolog, and InterProlog.

10. REFERENCES

[1] E. Abi-Lahoud, J. Hall, T. Butler, and D. Chapin. Interpreting regulations with SBVR. In *Joint Proceedings of the 7th International Rule Challenge, the Special Track on Human Language Technology and the 3rd RuleML Doctoral Consortium, hosted at the 8th International Symposium on Rules (RuleML2013)*, Seattle, WA, July 2013.

[2] T. H. Beach, Y. Rezgui, H. Li, and T. Kasim. A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Syst. Appl.*, 42(12):5219–5231, 2015.

[3] M. Bennett, D. Allemang, E. Kendall, P. Rivett, M. Uschold, and et. al. Financial Industry Business Ontology, May 2016. `<www.edmcouncil.org/financialbusiness>`.

[4] D. Elenius. SWRL-IQ: A prolog-based query tool for OWL and SWRL. In *Proceedings of OWL: Experiences and Directions Workshop 2012, Heraklion, Crete, Greece, May 27-28, 2012*, 2012.

[5] D. Elenius, D. Martin, R. Ford, and G. Denker. Reasoning about Resources and Hierarchical Tasks Using OWL and SWRL. In A. Bernstein et al., editor, *8th International Semantic Web Conference, ISWC 2009*, Lecture Notes in Computer Science, pages 795–810. Springer, 2009.

[6] Federal Reserve. http://www.federalreserve.gov/bankinforeg/topics/compliance.htm.

[7] B. Grosof and J. Bloomfield. Detailed explanations in English of rich reasoning for e-learning and compliance. In *10th Annual Semantic Technology and Business Conference*, San Jose, CA, August 2014. Dataversity.

[8] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description Logic Programs: Combining Logic Programs with Description Logic. In *Proc. 2nd International Semantic Web Conference (ISWC2003)*, 2003.

[9] E. Kendall, W. Moore, E. Abi-Lahoud, G. Denker, R. Ford, S. Riehemann, and R. Beatch. Semantics in finance. In *10th Annual Semantic Technology and Business Conference*, San Jose, CA, August 2014. Dataversity.

[10] M. E. Kharbili. Business process regulatory compliance management solution frameworks: A comparative evaluation. In *Eighth Asia-Pacific Conference on Conceptual Modelling, APCCM 2012, Melbourne, Australia, January 2012*, pages 23–32, 2012.

[11] M. E. Kharbili, Q. Ma, P. Kelsen, and E. Pulvermüller. Corel: Policy-based and model-driven regulatory compliance management. In *Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2011, Helsinki, Finland, August 29 - September 2, 2011*, pages 247–256, 2011.

[12] V. Kumar, A. Fensel, and P. Fröhlich. Context based adaptation of semantic rules in smart buildings. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, IIWAS '13, pages 719:719–719:728, New York, NY, USA, 2013. ACM.

[13] F. Polizel, S. J. Casare, and J. S. Sichman. Ontobacen: A modular ontology for risk management in the brazilian financial system. In *JOWO@IJCAI*, volume 1517 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[14] Quarule: Risk Controls and Compliance Certification Technology, 2016. `<www.quarule.com>`.

[15] S. Riehemann and D. Elenius. Ontological analysis of terrain data. In *COM.Geo*, page 10, 2011.

[16] Securities and Exchange Commission. http://www.sec.gov/rules/final/ia-2204.htm.

[17] R. D. Shankar, S. B. Martins, M. J. O'Connor, D. B. Parrish, and A. K. Das. Epoch: An ontological framework to support clinical trials management. In *Proceedings of the International Workshop on Healthcare Information and Knowledge Management*, HIKM '06, pages 25–32, New York, NY, USA, 2006. ACM.

[18] L. Sterling and E. Shapiro. *The Art of Prolog*. The MIT Press, 1986.

[19] Sunflower: A Framework for Ontologies and Rules, 2016. `<sunflower.csl.sri.com>`.

[20] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. `<www.w3.org/Submission/SWRL>`.

[21] *Transactions Between Member Banks and their Affiliates (Regulation W), Electronic Code for Federal Regulations, Title 12, Chapter II, Subchapter A, Part 223*. US Government Publishing Office.