Automated Decision Support for Financial Regulatory/Policy Compliance, using Textual Rulelog

Benjamin Grosof[1], Janine Bloomfield[1], Paul Fodor[1], Michael Kifer[1], Isaac Grosof[1], Miguel Calejo[2], and Terrance Swift[1]

[1] firstName.lastName@coherentknowledge.com; Coherent Knowledge Systems, USA
[2] mc@interprolog.com; InterProlog Consulting, Portugal

1 Business Case

A complex set of regulations and associated policies govern a wide range of operations and activities that financial institutions, such as banks and investment firms, engage in every day. Compliance, and *proof* of compliance, are essential: for both external regulators and internal management. The complexity and amount of regulations and related policies are ever-growing. The stakes are high: the global financial crisis of 2008 cost the overall economy trillions of dollars[3] while individual institutions spent over 100 billion dollars on fines, penalties, and legal settlements[4]. To mitigate the probability and severity of another financial crisis, the US and many other advanced industrial countries enacted a broad range of new regulations and policies. While these measures have improved somewhat the systemic risk of the global financial system, much more remains to be done to improve it further. Additional regulations, along with associated policies, are ongoingly being developed and issued, and likely will continue to be for many years in the foreseeable future.

Automated support is needed for compliance decisions and associated analysis, but currently deployed (i.e., previous) methods are expensive, unwieldy, and often quite inaccurate. These current methods often take the form of stand-alone or loosely connected components/services which review transactions or other activities, answer queries, and issue alerts. Typically, these methods require humans in the loop at run time, often because the scope of automation only captures part of the substance of the relevant regulations and policies. Overall, compliance is best viewed as mostly an aspect of operations, whose automation should be woven tightly into operational systems as a whole rather than as a separate step that sits architecturally only loosely coupled to other operational activities.

The market landscape in automated financial/regulatory compliance includes several kinds of vendors. One is providers of outsourced operations services; they often

[3] 'Financial Regulatory Reform: Financial Crisis Losses and Potential Impacts of the Dodd-Frank Act', GAO-13-180, 2013. `http://www.gao.gov/products/GAO-13-180` Accessed 28 May 2015.

[4] Banks pay out $100bn in US fines by R. McGregor and A. Stanley, Financial Times, March 25, 2014. `http://www.ft.com/cms/s/0/802ae15c-9b50-11e3-946b-00144feab7de.html#axzz3a5MXr6vx` Accessed 28 May 2015.

use IT automation. A second kind of provider furnishes IT development services. A third kind of provider furnishes software products (or software as a service) that facilitate building compliance systems. Others provide software solutions or data that are part of implementing particular regulations or policies. Even when these solutions are more, rather than less, complete for a particular regulation area, the solutions usually must accommodate extension to, and integration of, institution-specific policies, especially for larger institutional customers. In addition to utilizing vendors, larger institutions tend to implement quite a bit in-house as well. As with software applications generally, there is a momentum towards subscription and frequent updating, rather than traditional (infrequent-major-release) licensing, in both the software and the related data.

An example of a recent regulation issued after Dodd-Frank is the US Federal Reserve Act's Regulation W ("RegW").[5] It concerns activities/transactions between a bank and its counterparties (mainly, companies) that are defined as "affiliates" which share ownership, control, or advisory relationships. It is designed to limit concentrations of risks to an individual institution, and the banking system as a whole. Compliance requires avoiding transactions, or reporting transactions, under certain conditions.

2 Technological Challenges

There are a number of factors that make it difficult for previous methods to solve the problem well. The regulations (and associated policies) are frequently very complicated in both their logical/semantic substance and their English syntax ("legalese" is notorious). The regulations are full of meta information and rife with important exception cases requiring defeasibility. The body of regulations is voluminous, continually increasing, and ever changing. For these often high-stakes compliance decisions, accuracy and reasoning efficiency (near real time) are necessary, as are provenance and audit trails to demonstrate (often as potential legal evidence) why decisions were taken. The complicated English rules (and definitions) must be encoded into automated form suitable for reasoning. And a variety of enterprise data, not just transactions, needs to be integrated with the implemented regulations in order to perform that reasoning. Subject matter experts without extensive training in logic or programming (SME's) need to be involved closely in developing, testing, and debugging the encoded implementation. Yet it's hard for them to understand, much less contribute to, the implemented form.

3 Rule-based Solution

We developed and applied an approach based on Textual Rulelog [3, 1, 2], implemented in our (Coherent Knowledge Systems') Ergo Suite™ platform ("Ergo")[6]. Ergo includes

[5] Part 223 Transactions between member banks and their affiliates (Regulation W), Electronic Code of Federal Regulations (e-CFR), GPO `http://www.ecfr.gov/cgi-bin/text-idx?tpl=/ecfrbrowse/Title12/12cfr223_main_02.tpl` Accessed 28 May 2015.

[6] `http://coherentknowledge.com`

both a Reasoner and an integrated development environment (IDE), Studio. Ergo provides: very high logical expressiveness, including for higher-order, defeasibility, quantifiers, head disjunction, and meta knowledge; and efficient dynamic automated reasoning capabilities, including fully detailed user-navigable explanations (for each answer) that are quite understandable by SME's. Ergo integrates tightly some English natural language capabilities both for authoring (i.e., developing) rules – going from English phrases to logical expressions ("text interpretation") – and for generating answers with explanations – which leverages mapping from logical expressions to English phrases ("text generation"). Ergo also includes connectors that import knowledge from OWL/RDF and other forms (e.g., relational databases and spreadsheets), then tightly integrate the imported knowledge into overall reasoning.

Our methodology for rule development ("authoring") had several steps. Starting from English *source sentences* in published regulations and other documents, we *articulated* a set of English *encoding sentences* that were clearer and syntactically more self-contained, along with some additional *background* knowledge encoding sentences. Then we *encoded* each encoding sentence into a logical rule in Ergo syntax, using a *textual terminology* style in which each English phrase is mapped closely (nearly isomorphically) to and from a corresponding Ergo logical term (typically higher-order); English words were used as Hilog functors, then composed into *phrasal terms*. We encoded additional rules that specified *textual templates* for text generation (used heavily in explanation) and text interpretation (as an extension of the original work). We also developed *ontology mapping* rules that supported effective reasoning with imported OWL knowledge. We iteratively tested (queried) and debugged the rules, intensively using Ergo's explanations capability. Explanations helped to find both missing knowledge and incorrect knowledge.

Ergo is flexibly deployable. The implemented Ergo based component, including its knowledge bases, can be integrated into an overall compliance product, service, or enterprise application, in several different ways. It can be queried, started or stopped, loaded, and configured via its Java API. Under development are additional methods including a RESTful web service wrapper.

Ergo scales well computationally in several important regards. First, its fundamental logic (i.e., semantic knowledge representation), Rulelog, is an extension of database logic, i.e., of well-founded declarative logic programs, and is equipped with restraint, a form of bounded rationality that allows one to ensure that inferencing is worst-case polynomial-time. Second, its fundamental reasoning algorithms have many performance optimizations, including compilation, transformation, indexing, and subgoal reordering; they employ a form of LP *tabling* [4, 5] that caches inferences and thereby reuses previous computations on subgoals when computing an answer to a new goal. Ergo inferencing/computation is primarily in main memory; however, it hooks up to other components, such as databases and triple stores, that are disk-based. On an ordinary current laptop or desktop, it scales up to millions, but not billions, of complex inferences and associated dependencies and fact assertions, while occupying roughly 5-10 gigabytes of RAM. One can take reasoning to yet larger scales by using processors equipped with more RAM, and/or by distributing reasoning computations across multiple Ergo instances on multiple processors.

Ergo has special performance optimizations for RDF data in bulk to be queried in SPARQL from Ergo, translated into Ergo, and loaded into Ergo (for use in reasoning), at high speed. This import and tightly integrated reasoning scales to millions of RDF triples. Under development are additional methods for distributed processing methods in order to scale this up further.

4 Results

Our approach was originally developed in coordination with, and in support of, an overall application-piloting proof-of-concept (PoC) effort, called "FIBO Rules", conducted by the Enterprise Data Management Council (EDMC), a leading financial-sector international industry-government consortium. The PoC was conceived at the FIBO Summit, held in June 2013 at the SemTechBiz SJ conference. RegW was identified by banking participants as a challenge regulation that had significant complexity and industry urgency. One of us (B. Grosof) acted as technical lead for the PoC. Besides Coherent, other participants in the PoC included Wells Fargo Bank, SRI International, and GRCTC (Ireland). The PoC very successfully reached its goal of demonstrating how a representative subset of RegW could be effectively automated using Textual Rulelog while also leveraging the EDMC/OMG Financial Industry Business Ontology (FIBO). The business benefits of our rule-based solution were dramatic, including: higher accuracy; solution development that has lower cost, greater agility, higher reusability, and more SME participation; and greater scope of automation including SME-understandable explanations/provenance. The PoC was presented to industry audiences in 2014 at the OMG Technical Meeting plenary session on March 26, an EDMC webinar on June 26, and multiple sessions of the SemTechBiz SJ industry conference on August 21.[7,8,9,10]

The technical benefits of our approach, that enable the business benefits, revolve especially around expressiveness, authoring, and explanations. Textual Rulelog provides very high expressiveness of rules and answers, by combining: defeasible higher-order logic formulas; justification graphs that are selectively expandable/collapsible; and text

---

[7] Semantics - Crossing the Chasm Workshop, OMG Technical Meeting Special Event, March 26, 2014, Reston, VA, `http://www.omg.org/news/meetings/tc/va-14/special-events/Sem-Fin_Day.htm` Accessed 28 May 2015.

[8] EDM Council Industry Webinar 26 June 2014, `https://www.youtube.com/watch?v=8c9WmMW16t8&feature=youtu.be#t=11m00s`Accessed 28 May 2015.

[9] 'Leading Financial Industry Consortium Proof-Of-Concept Features Coherents Ergo Suite', Coherent Knowledge Systems website, `http://coherentknowledge.com/leading-financial-industry-consortium-proof-of-concept-features-coherents-episto/` Accessed 28 May 2015.

[10] Semantic Technology and Business Conference, August 19-21, 2014, San Jose, CA, Sessions titled *Detailed Explanations in English of Rich Reasoning for E-Learning and Compliance*, *Semantics in Finance: Addressing Looming Train Wreck in Risk Management, Regulatory Compliance and Reporting*, *Semantics: the Technical Lens for Financial Transparency and Risk Management* and *Grill the experts on FIBO and its underlying technology*, `http://semtechbizsj2014.semanticweb.com/agenda.cfm?confid=82&scheduleDay=08/21/14` Accessed 28 May 2015.

generation/interpretation. This reduces the cost/time, and logical skill, needed for authoring and increases the understandability of explanations.

Next, we give some illustrative examples of rules and explanations for this RegW case study. An example query in Ergo is:

```
What proposed transactions are prohibited by RegW?
      Show (?Bank,?Company,?Amount).
```

Here, the prefix "?" before "Bank" indicates a logical variable; likewise, before "Company" and before "Amount". The set of answers for a query can be displayed in Ergo Studio as a table of answer tuples, where each tuple has one element per variable in the query. An example answer tuple for this particular query is: `'Pacific Bank', 'Maui Sunset', 23.0`. This answer tuple corresponds to the variable tuple "`?Bank, ?Company, ?Amount`". Here the "23.0" indicates millions of dollars (for brevity's sake).

The user can then ask "Why" by right-clicking on that answer, and Ergo in response will automatically generate and present an explanation. The explanation is a justification graph, presented as a sideways tree, in which each line corresponds to one step of rule inferencing and is analogous to a line in a natural-deduction style proof (remember high school geometry?). Each explanation line has a handle at its left. By clicking on the handle of a line, the user can selectively expand a portion of the explanation. This can be done repeatedly to drill down into full detail of the explanation (leaf nodes of the justification graph correspond to asserted facts). After the user has inspected that portion of the explanation, the user can then collapse the portion (i.e., its subtree) by again clicking on the same handle. Figure 1 is an example of a detailed such explanation, for the example answer tuple above.

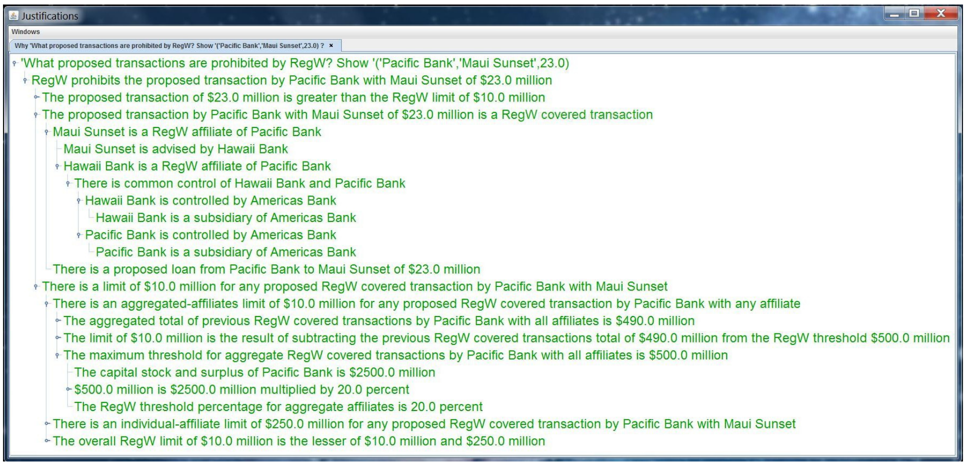

Fig. 1. An example of explanation

Ergo generates explanations by first creating a justification graph (JG) whose nodes are logical facts, then transforming each node in the JG. Text generation occurs during this transformation of a JG node. Text generation uses mappings of logical formulas to

English sentences; the mappings are specified via Ergo rules (textual templates, mentioned earlier).

Ergo's high expressiveness makes it easier to author rules by mapping from English to logic, as well as vice versa. Examples of Ergo facts are:

```
subsidiary(of)('Pacific Bank','Americas Bank').
advised(by)('Maui Sunset','Hawaii Bank').
bank('Hawaii Bank').
company('Maui Sunset').
capital(stock(and(surplus)))('Pacific Bank',2500.0).
proposed(loan) (from('Pacific Bank'))(to('Maui Sunset'))
  (of(amount(23.0))) (having(id(1101))).
previous(loan)(from('Pacific Bank'))(to('Hawaii Bank'))
  (of(amount(145.0))) (having(id(1001))).
proposed(asset(purchase))(by('Pacific Bank'))
  (of(asset(common(stock)(of('Flixado')))))(from('Maui Sunset'))
  (of(amount(90.0)))(having(id(1202))).
```

Examples of non-fact rule assertions are:

```
/* A company is controlled by another company when the first
 company is a subsidiary of a subsidiary of the second company.*/
@!{rule103b} /* declares rule id */
@@{defeasible} /* indicates the rule can have exceptions */
 controlled(by)(?x1,?x2)
   :-  /* the "if" symbol */
      subsidiary(of)(?x1,?x3) \and
      subsidiary(of)(?x3,?x2).
/*A case of an affiliate is: Any company that is advised on a
 contractual basis by the bank or an affiliate of the bank. */
@!{rule102b} @@{defeasible}
 affiliate(of)(?x1,?x2) :-
   ( advised(by)(?x1,?x2)
     \or
     (affiliate(of)(?x3,?x2) \and advised(by)(?x1,?x3))).
```

Here "`/*...*/`" encloses a comment; "`:-`" means "if"; "`@!{...}`" is meta info that indicates a rule id. "`@@{defeasible}`" means the rule can have exceptions. "`@{...}`" encloses a rule tag, a label used for specifying prioritization information in handling of conflicts between rules, for reasoning about exceptions. "$\backslash overrides$" is a predicate used to specify such prioritization-type precedence info. Example rules that specify an exception case are:

```
@!{rule104e}
@{ready market exemption case for covered transaction'}
   /* tag  for prioritizing */
\neg covered(transaction)(by(?x1))(with(?x2))
     (of(amount(?x3)))(having(id(?Id))) :-
   affiliate(of)(?x2,?x1) \and
   asset(purchase)(by(?x1))(of(asset(?x6)))(from(?x2))
```

```
      (of(amount(?x3)))(having(id(?Id))) \and
   asset(?x6)(has(ready(market))).
/* prioritization info, as one tag being higher than another */
\overrides(ready market exemption case for covered transaction',
    'general case of covered transaction').
/* If a company is listed on the New York Stock Exchange (NYSE),
  then the common stock of that company has a ready market. */
@!{rule201} @@{defeasible}
asset(common(stock)(of(?Company)))(has(ready(market))) :-
        exchange(listed(company))(?Company)(on('NYSE')).
```

IRI prefixes can be specified in Ergo syntax via a directive that declares them. An example is:

```
:- iriprefix fibof =  /* declares an abbreviation */
"http://www.omg.org/spec/FIBO/FIBO-Foundation/20120501/ontology/".
```

In Ergo syntax, "`foo#`" indicates that foo is being used as a shortname IRI prefix. Examples of imported knowledge from FIBO (facts translated from OWL/RDF into Ergo) are:

```
rdfs#subClassOf(fibob#BankingAffiliate,fibob#BodyCorporate).
rdfs#range(fibob#whollyOwnedAndControlledBy,
  fibob#FormalOrganization).
owl#disjointWith(
  edmc#Broad_Based_Index_Credit_Default_Swap_Contract,
  edmc#Narrow_Based_Index_Credit_Default_Swap_Contract).
```

In order to reason effectively with knowledge imported from OWL, Ergo includes some general axioms that specify the semantics of OWL. Examples of such axioms are:

```
?r(?y) :- rdfs#range(?p,?r), ?p(?x,?y).
?p(?x,?y) :- owl#subPropertyOf(?q,?p), ?q(?x,?y).
```

Ergo is also very capable at representing ontology mappings. To tightly integrate the knowledge imported from FIBO with the other knowledge in Ergo that represented the RegW regulations and information that originated in English, we developed rules that mapped between textual terminology and FIBO OWL vocabulary. Examples of such ontology mapping rules are:

```
company(?co) :- fibob#BodyCorporate(?co).
fibob#whollyOwnedAndControlledBy(?sub,?parent) :-
       subsidiary(of)(?sub,?parent).
```

5 Importance and Impact

Our rule-based approach offers the realistic promise to increase the productivity (i.e., reduce cost and risk) of financial regulatory/policy compliance by at least several percent. Such a productivity improvement will, over the next decade or two, be worth many billions of dollars to the global economy, and to the individual institutions that are players in banking and investment. In addition to the productivity advantages of our approach, increasing systemic stability has many non-economic benefits. Moreover, the radically increased transparency afforded by our approach has the potential to significantly improve the governance of the process of writing and enforcing regulations, thereby mitigating the critical cluster of problems arising from exploitative gaming of the regulation system by financial-institution players. These problems are long-studied and include over-complexity of regulation a.k.a. "overregulation", "regulatory capture", and "regulatory arbitrage"[11].

Acknowledgements

Thanks to all the FIBO Rules effort's participants for their insights and collaboration on RegW and financial regulatory compliance via semantic technology, particularly: Dennis Wisnosky, Michael Bennett, and Michael Atkin of EDMC; David Newman, Wesley Moore, Cheryl Maske, and Patrick Greenfield of Wells Fargo (David also wore an EDMC hat for the effort); Elie Abi-Lahoud of GRCTC; and Daniel Elenius, Grit Denker, Susanne Riehemann, Reg Ford, and John Shockley of SRI International.

References

1. Grosof, B., Dean, M., Kifer, M.: Semantic Web Rules: Fundamentals, Applications, and Standards (July 2013), tutorial presented at AAAI-2013; available at `http://silk.semwebcentral.org/`
2. Grosof, B., Kifer, M., Fodor, P.: The Power of Semantic Rules in Rulelog: Fundamentals and Recent Progress (August 2015), tutorial presented at RuleML-2015; available at `http://www.coherentknowledge.com/publications/`
3. Grosof, B.: Rapid Text-Based Authoring of Defeasible Higher-Order Logic Formulas, via Textual Logic and Rulelog. In: Morgenstern, L., Stefaneas, P., Lvy, F., Wyner, A., Paschke, A. (eds.) Theory, Practice, and Applications of Rules on the Web, Lecture Notes in Computer Science, vol. 8035, pp. 2–11. Springer Berlin Heidelberg (2013), `http://dx.doi.org/10.1007/978-3-642-39617-5_2`
4. Swift, T., Warren, D.: XSB: Extending the Power of Prolog using Tabling. Theory and Practice of Logic Programming (2011)
5. Swift, T.: Incremental Tabling in Support of Knowledge Representation and reasoning. Theory and Practice of Logic Programming 14(4-5), 553–567 (2014), `http://dx.doi.org/10.1017/S1471068414000209`

[11] 'Wall Street is Using the Power of Dodd-Frank Against Itself' by A. Davidson, NY Times (Magazine), May 27, 2015 `http://www.nytimes.com/2015/05/31/magazine/wall-street-is-using-the-power-of-dodd-frank-against-itself.html` Accessed 28 May 2015