The Power of Deep Reasoning with Large Graph Data

Benjamin Grosof and Janine Bloomfield
Coherent Knowledge*

Presentation (45-min.) at
SmartData Conference**
to be held August 18-20, San Jose, California, USA

Version of Aug 17, 2015

* Web: http://coherentknowledge.com
Email: *firstname.lastname*@coherentknowledge.com

** Web: http://smartdata2015.dataversity.net/

Coherent Knowledge: Company Overview

- Leverages over a decade of major government and privately funded research advances in artificial intelligence (AI) and semantic technologies. Founded 7/2013.
- Company offers: platform software product Ergo Suite™ + custom dev / services
  - Apps: policy/regulatory compliance for finance, defense intelligence analysis, e-commerce
- World-class founder team: created many industry-leading logic systems & standards
  - XSB Prolog, RuleML, W3C RIF, W3C OWL-RL, IBM Common Rules, SWRL, SweetRules
  - Extensive experience applying logic systems to numerous domains in govt. and biz.

Michael Kifer, PhD
Principal Engineer
Creator, Flora.
Co-Architect, W3C RIF.
Prof., Stonybrook Univ.

Benjamin Grosof, PhD
CTO & CEO
Creator, IBM Common Rules.
Co-Architect, RuleML.
Prof., MIT.
Advanced AI Prog. Mger. for Paul Allen.

Theresa Swift, PhD
Principal Engineer
Co-Architect, XSB Prolog.
Consultant, US CBP.

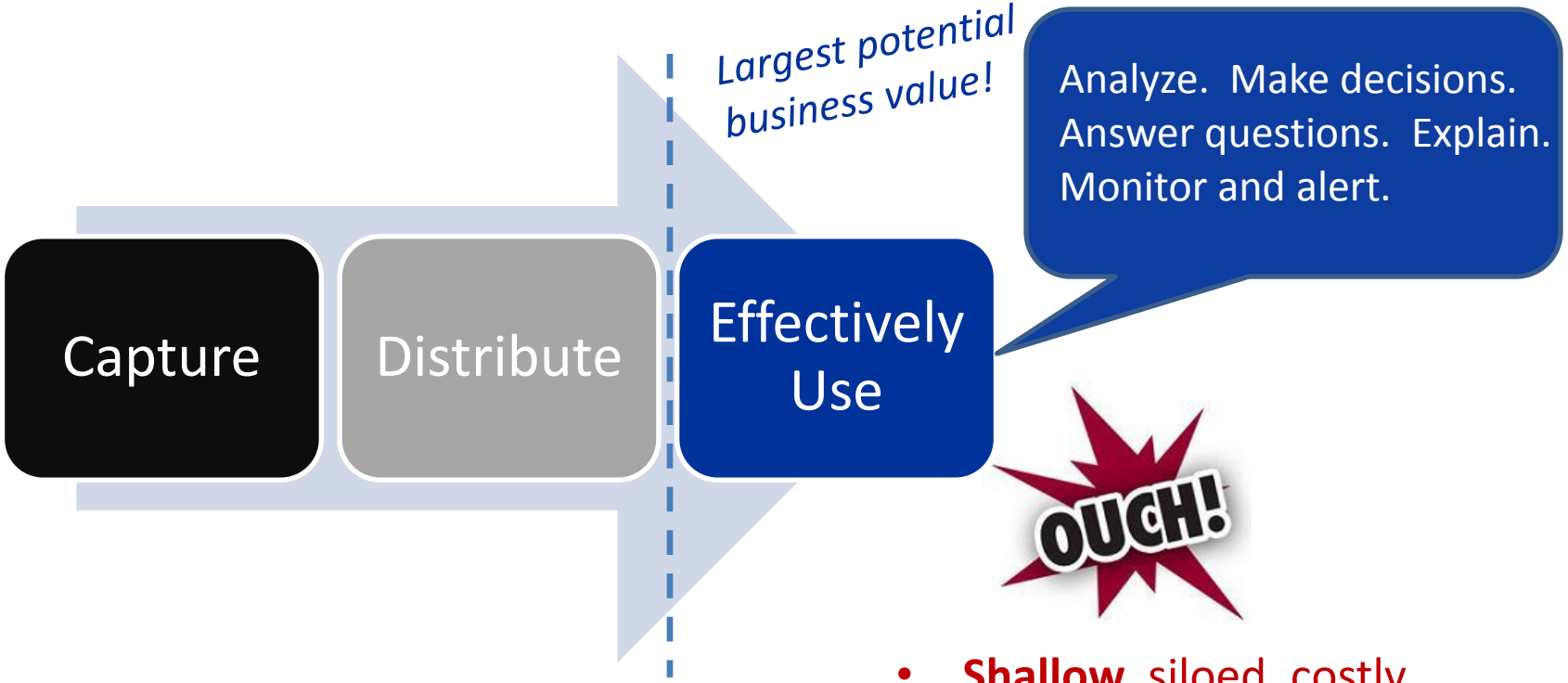
Paul Fodor, PhD
Senior Engineer
IBM Watson team.
Prof., Stonybrook Univ.

Janine Bloomfield, PhD
Dir., Marketing & Operations
Sr. Scientist, Climate Change, Environmental Defense Fund.
Mindexplorekids.org.

Outline

- What
- How
- Case Studies
- Windup

Problem: Knowledge Management

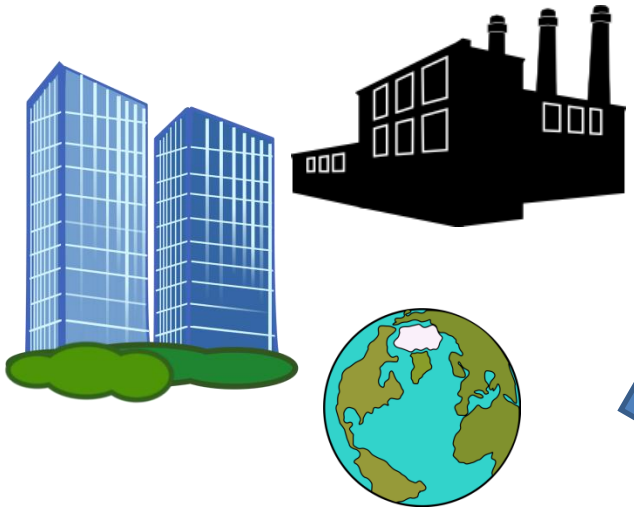


Existing Technologies

- Content Management
- Search
- Business Rules
- Databases

- **Shallow**, siloed, costly
- Not actionable
- Lacks automation, accuracy
- Lacks transparency
- End users and subject matter experts are not empowered

Evolution of Enterprise Knowledge Management

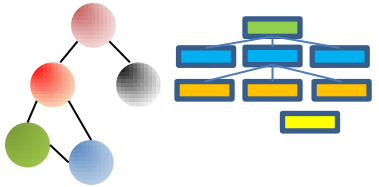


Data integration with flexible schemas,
more meta-data. E.g., RDF, SPARQL, OWL.

Augment relational/traditional data stores.

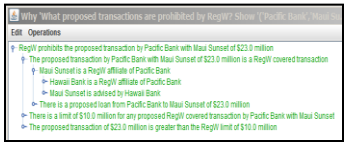
Smart Data

Graph/Linked Data bases



Smart Rules

Decisions & Analysis



Deep reasoning, more complex knowledge.
E.g., Textual Rulelog.

Leverage Smart Data investment.

Smart Rules with Deep Reasoning

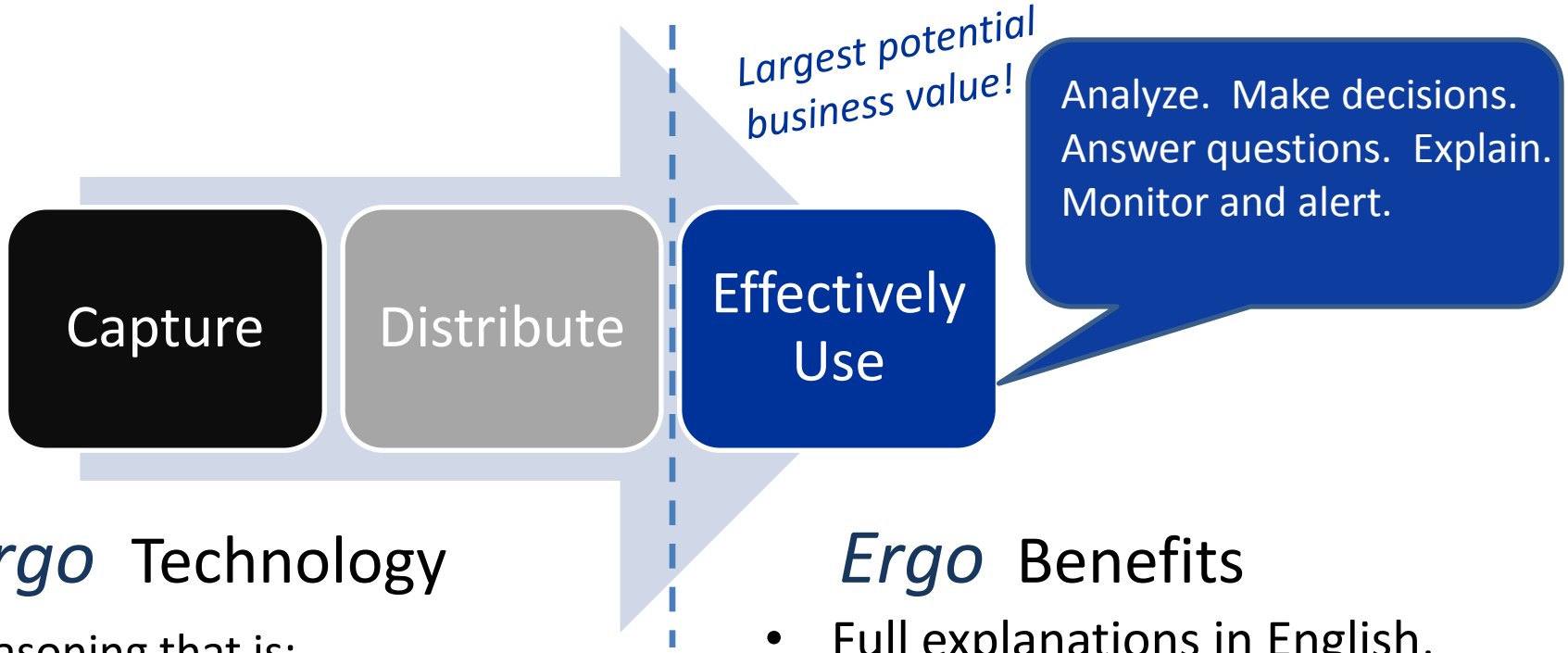
- Provide critical technical capabilities to unlock business value
  - More complex analytics
  - Context and mappings for data and system integration
  - Represent more complex knowledge: policies, regulations, science, …
  - Capture subject matter experts' (SMEs') insights directly, semantically

- Case studies in this presentation
  - Financial services regulatory/policy compliance
  - National intelligence analysis
  - E-commerce marketing

  - *Many other applications: health treatment guidance, info access, tutoring, …*

Smart Rules for Smart Data™

Solution: Deep Reasoning in Ergo™



Ergo Technology

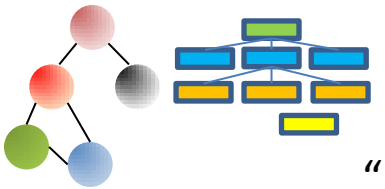
- Reasoning that is: deep, accurate, transparent
- Knowledge that is: flexible, complex
- Scalable. Draft standard.
- Recent research breakthrough in theory & algorithms

Ergo Benefits

- Full explanations in English, navigable in detail
- End users and subject matter experts are empowered
- Lower labor & cost. More agile.
- More automated, accurate
- Greater integration

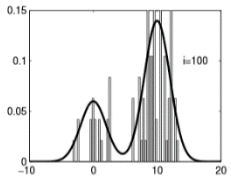
Actively Reason over Today's Gamut of Knowledge

Graph DB
& Semantic tech



Relational DB

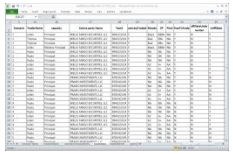
Machine
Learning



$Prob_N(x) = \Sigma_{i=1..N} \delta_{si}(x) / N$

Domain Apps
& Legacy

Spreadsheets



"Business Rules"

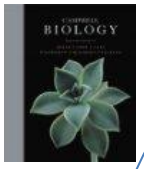
$tier(X,1) \wedge supply(Y,X) \Rightarrow tier(Y,2)$

Probabilistic engines

$p(H|C) = \Pi\, p(x_i|Y_j) / \Sigma_C \Pi\, p(x_i|Y_j)$

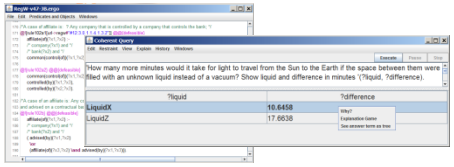
External Info
& Services

Text & Natural
Language processing



Application
Actions

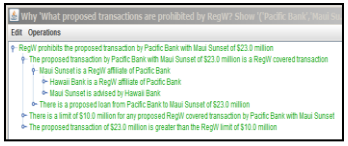
Queries
Assertions
Edits



ERGO

KB
Libraries

Answers, Views
Decisions, Alerts
Explanations



Outline

- What
- How
- Case Studies
- Windup

Technological Approach

- Problem: high logical expressiveness is required for
  - Complex knowledge: policies, regulations, science
    - Often initially stated in natural language text
  - Data/knowledge integration mappings

- Solution: techniques in Coherent's Ergo platform for smart rules
  - *Rulelog* – fundamental logical knowledge representation
    - Full Meta expressiveness: higher-order, defeasible, ...
  - *Textual* Rulelog – adds close relationship between text (English) and logic
  - *Explanations* – fully detailed, interactively navigable, in text (English)
    - Understandable to SMEs
  - *Optimizations* for large amounts of changing data

➢Address the 3 V's of big data, in combination with deep reasoning
  - Volume, velocity, variety

Capabilities: Focus in this presentation

- Import RDF data in bulk into the rule engine's knowledge base
- Externally query triple stores via SPARQL, dynamically during overall rule reasoning
- Handle frequent updates to asserted rules/facts, while maintaining validity of *inferred* facts
- Facilitate SMEs' understanding of rules and contributions to rule authoring, by supporting close relationship between

Text: English natural language

Logic: rules, queries, answers, explanations

Ergo Architecture

Optional Custom Solutions

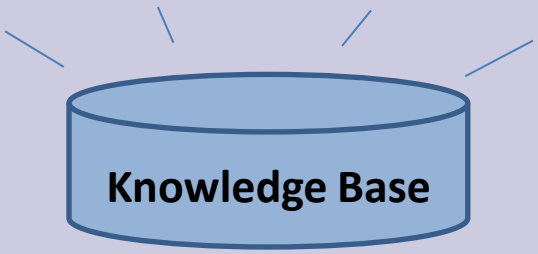
Ergo Suite

Ergo Studio

Rule Editor and Query UI

(Integrated Development Environment)

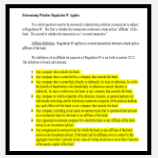
Ergo Reasoner



queries, assertions

answers, explanations

Complex Information
- English Doc.'s etc.
- Policies, Regulations
- Financial, Legal, Science



External Info
(multi-source)

- Data
- Views, Rules
- Schemas & Ontologies
- Results of ML

API's

Java
WS
C

External Services & Frameworks

- Relational DB
- RDF/Graph DB
- Other Sem. Tech
- Machine Learning
- Apps, Docker, …

events, decisions

App Actions

WS = Web Services. Sem. = Semantic. ML = Machine Learning

Ergo Reasoner & Ergo Studio (IDE/UI)

- Textual Rulelog: Implementation of major research advances in logic (Rulelog) and how to map between logic and English (Textual Logic)
  - **Ergo is the most complete & highly optimized implementation available**
  - Rulelog significantly extends Datalog, the logic of databases, business rule systems (production/ECA/Prolog), semantic web ontologies, and earlier-generation semantic web rules cf. SWRL/RIF/RuleML
- Ergo Reasoner component – with sophisticated algorithms
  - Reordering, caching, transformation, compilation, indexing, modularization
- Ergo Studio component – User Interface with array of advanced techniques
  - Integrated Development Environment (IDE). Visualizations of knowledge.
  - Fast edit-test loop with award-winning toolset
- Knowledge interchange with leading and legacy systems
  - SQL, SPARQL, RDF, RDF-Schema, OWL. Others in dev or easy to add. Fully automatic.
- Open, standards-based approach. Builds on open source components.
  - Rulelog is draft industry standard from RuleML (submission to W3C & Oasis)

Ergo Suite – Coherent Knowledge Management Platform

- Unprecedented flexibility in the kinds of complex info that can be stated as assertions, queries, and conclusions (highly expressive "knowledge" statements)
  - Almost anything you can say in English – concisely and directly
  - Just-in-time introduction of terminology
  - Statements about statements (meta knowledge) – contextualizes knowledge
  - State and view info at as fine a grain size as desired
- Probabilistic info combined in principled fashion, tightly combined with logical
  - Tears down the wall between probabilistic and non-probabilistic
- Unprecedented ease in updating knowledge
  - Map between terminologies as needed, including from multiple sources
- Conflict between statements is robustly handled (often arises during integration)
  - Resolved based on priority (e.g., authority), weighting, or else tolerated as an impasse
- Scalable and computationally well-behaved

KRR Features Comparison: Rulelog Shines

Feature \ System	Rulelog Rules - e.g., Ergo	Datalog Rules - e.g., Jena, SWRL, Ontobroker, SPIN	Production Rules - e.g., IBM, Oracle, Red Hat	Prolog - e.g., SICStus, SWI, XSB	FOL & OWL-DL - e.g., Vampire, Pellet, Prover9	ASP Solvers - e.g., DLV, CLASP
Semantic & on standardization path	✓	✓	restricted case	restricted case	✓	✓
Basic expressiveness						
• Datalog LP	✓	✓	✓	✓	✓	✓
• Logical functions	✓	✗	✗	✓	✓	✓
• General formulas	✓	✗	✗	✗	✓	✗
Full Meta expressiveness						
• Higher-order syntax, provenance	✓	✗	✗	✗	✗	✗
• Defeasibility & well founded negation	✓	✗	✗	✗	✗	✗
• Restraint bounded rationality	✓	✗	✗	✗	✗	✗
• Probabilistic	✓	✗	✗	✗	✗	✗
Efficiency						
• Goal-directed	✓	✗ (except Jena)	✗	✓	✓	✓
• Full LP tabling with dependency-aware updating	✓	✗	✗	✗ (except XSB)	✗	✗
• Polynomial time complexity	✓	✓	✓	✗	✗	✗

Notes on KRR Features Comparison

- "System" means system type / approach of logical knowledge representation and reasoning (KRR).
- "Semantic" means in the sense of KRR, i.e., fully declarative and having a model theory in the logical sense.
- "FOL" means First Order Logic. "ASP" means Answer Set Programs.
  - ASP is recently emerging. The tasks for which it's suitable are more similar to FOL than to the other systems here.
- "Standardization" here means industry standardization. "On path to" means in process of being, or already, standardized.
- "Restricted case" means for a syntactic/expressive subset.
- Event-condition-action rules in this context are similar to, and lumped in with, production rules.
- "LP" means declarative logic programs.
- Datalog means LP without logical functions. Usually this is restricted to Horn. But here we permit negation(-as-failure).
- OWL-RL is pretty much a restricted case of Datalog LP.
- "Higher-order syntax" means Hilog, which enables probabilistic – and also 1) fuzzy and 2) frame syntax cf. F-Logic.
- "Provenance" means provenance info about assertions, via properties of rule id's that are within the logical language / KRR.
- "Full" applies to all four of the meta expressiveness features.
- Defeasibility includes flexible argumentation theories.
- "General formulas" means classical-logic-like formulas, including with head existentials and with head disjunction.
- "LP tabling" includes sophisticated: cacheing of intermediate reasoning results, inference control, and indexing.
- "Dependency-aware updating" means that when assertions are added or deleted, saved inferences are only recomputed if they depend on the changes to the assertions.
- Polynomial time "complexity" means worst-case computational complexity, with constant-bounded number of variables per rule. Polynomial-time is similar to database querying, and is a.k.a. "tractable".

- Datalog X defeasibility: Ontobroker has full well founded negation.
- Prolog X defeasibility: XSB has full well founded negation.
- ASP X defeasibility: ASP has restricted defeasibility & well founded negation.
- Datalog X goal-directed: Jena has a backward engine as well as a forward engine.
- ASP X general formulas: ASP has head disjunction.
- FOL X full LP tabling with dependency-aware updating: Some FOL theorem-provers cache intermediate results in a way that is analogous to LP tabling, and some do dependency tracking but we're not sure how analogous or sophisticated.

Textual terminology

- Leverage Rulelog to much more simply and closely map between natural language and logic

- English phrase ↔ logical term in Rulelog
- English word ↔ logical functor in Rulelog

- Basis for textual templates

Outline

- What
- How
- Case Studies
- Windup

Case Study 1: Automated Decision Support for Financial Regulatory/Policy Compliance

Problem: Current methods are expensive and unwieldy, often inaccurate

Solution Approach – using Textual Rulelog software technology:

- Encode regulations and related info as semantic rules and ontologies
- Fully, robustly automate run-time decisions and related querying
- Provide understandable full explanations in English
  - *Proof*: Electronic audit trail, with provenance
- Handles increasing complexity of real-world challenges
  - Data integration, system integration
  - Conflicting policies, special cases, exceptions
  - What-if scenarios to analyze impact of new regulations and policies

Business Benefits – compared to currently deployed methods:

- More Accurate
- More Cost Effective – less labor; subject matter experts in closer loop
- More Agile – faster to update
- More Overall Effectiveness: less exposure to risk of non-compliance

Demo of Ergo Suite for Compliance Automation: US Federal Reserve Regulation W

- EDM Council Financial Industry Consortium Proof of Concept – ***successful and touted pilot***
  - Enterprise Data Management Council (Trade Assoc.)
  - Coherent Knowledge Systems (USA, Technology)
  - SRI International (USA, Technology)
  - Wells Fargo (Financial Services)
  - Governance, Risk and Compliance Technology Centre (Ireland, Technology)
- Reg W regulates and limits $ amount of transactions that can occur between banks and their affiliates. Designed to limit risks to each bank and to financial system.
- Must answer 3 key aspects:

1. *Is the transaction's counterparty an affiliate of the bank?*
2. *Is the transaction contemplated a covered transaction?*
3. *Is the amount of the transaction permitted ?*

Determining Whether Regulation W Applies

Two initial questions need to be answered in determining whether a transaction is subject to Regulation W. The first is whether the transaction is between a bank and an "affiliate" of the bank. The second is whether the transaction is a "covered transaction."

Affiliate Definition. Regulation W applies to covered transactions between a bank and an affiliate of the bank.

The definition of an affiliate for purposes of Regulation W is set forth in section 223.2. The definition is broad, and includes:

- Any company that controls the bank;
- Any company that is controlled by a company that controls the bank;
- Any company that is controlled, directly or indirectly, by trust or otherwise, by or for the benefit of shareholders who beneficially or otherwise control, directly or indirectly, by trust or otherwise, the bank or any company that controls the bank;
- Any company in which a majority of its directors, trustees, or general partners (or individuals exercising similar functions) constitute a majority of the persons holding any such office with the bank or any company that controls the bank;
- Any company, including a real estate investment trust, that is sponsored and advised on a contractual basis by the bank or an affiliate of the bank;
- Any registered investment company for which the bank or any affiliate of the bank serves as an investment adviser;
- Any unregistered investment fund for which the bank or any affiliate of the bank serves as an investment adviser, if the bank and its affiliates own or control in the aggregate more than 5 percent of any class of voting securities or more than 5 percent of the equity capital of the fund[1];

The Starting Point - Text of Regulation W

Demo goes here

- Note: The demo actually covers the next 4 slides that show executable assertions in Textual Rulelog. The deck includes those slides in order to be more self-contained.

Examples of the Underlying Textual Rulelog Executable **Fact** Assertions

- 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))).

Executable Assertions: non-fact **Rules**

```
/* 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)
 :- /* if */
   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))).
```

Executable Assertions: **Exception** Rule

```
@!{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, specified 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')).
```

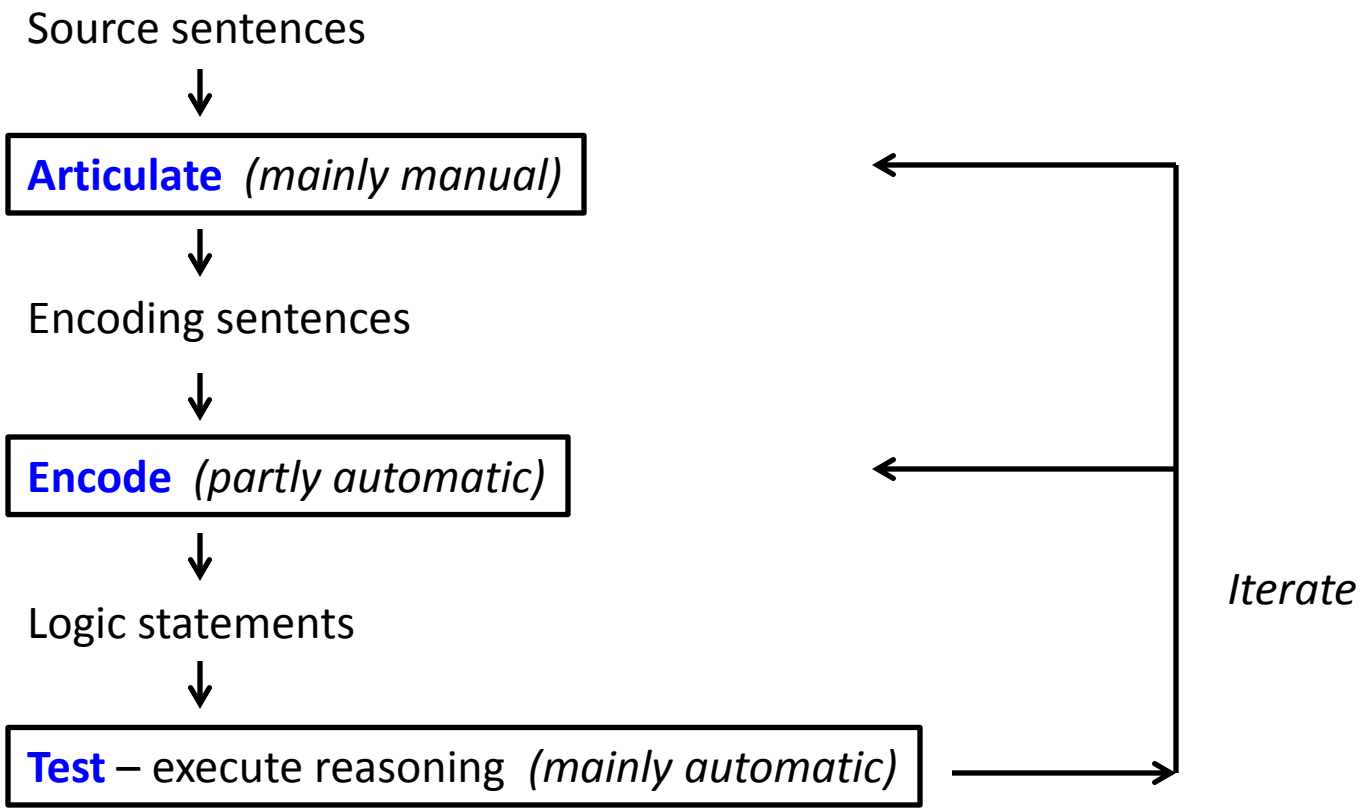
Executable Assertions: **Import of OWL**

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

/* Imported OWL knowledge:  from Financial Business Industry Ontology (FIBO) */
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).

/* Ontology Mappings between textual terminology and FIBO OWL vocabulary */
company(?co) :- fibob#BodyCorporate(?co).
fibob#whollyOwnedAndControlledBy(?sub,?parent) :- subsidiary(of)(?sub,?parent).

/* Semantics of OWL - specified as general Rulelog axioms */
?r(?y) :- rdfs#range(?p,?r), ?p(?x,?y).
?p(?x,?y) :- owl#subPropertyOf(?q,?p), ?q(?x,?y).
```

Knowledge Authoring Process using Ergo Suite

- Start with source text in English – e.g., textbook or policy guide
  - A sentence/statement can be an assertion or a query
- Articulate: create encoding sentences (text) in English.
  As necessary:
  - Clarify & simplify – be prosaic and grammatical, explicit and self-contained
  - State relevant background knowledge – that's not stated directly in the source text
- Encode: create executable logic statements
  - Each encoding text sentence results in one executable logic statement ("rules")
  - Ergo Suite has tools and methodology
- Test and debug, iteratively
  - Execute reasoning to answer queries, get explanations, perform other actions
  - Find and enter missing knowledge
  - Find and fix incorrect knowledge
  - Optionally: further optimize reasoning performance, where critical

Knowledge Authoring Steps using Ergo Suite



In-development: methods to greatly increase the degree of automation in encoding

Case Study 2: E-commerce marketing: Configuring Complex Products

- Very large manufacturing company
- Product catalog: highly technical, complex
- Over 1 Million products, in over 50 countries
- Problem: Creating configurators and updating catalog too slow & costly. Thus siloed by sales regions.
- Solution (with smart data partner) on $multi-Billion segment
  - Developed product ontology: compatible with legacy and standards
  - Converted all product data into live triple store
  - Added configuration rules, in Rulelog, on top of that catalog
  - Leveraged Ergo expressiveness and RDF import capability
- Benefits: faster, cheaper, agile updating, reusable globally
  - Enable greatly enhanced up-sell and cross-sell, across product lines

Importing Large Amounts of Data

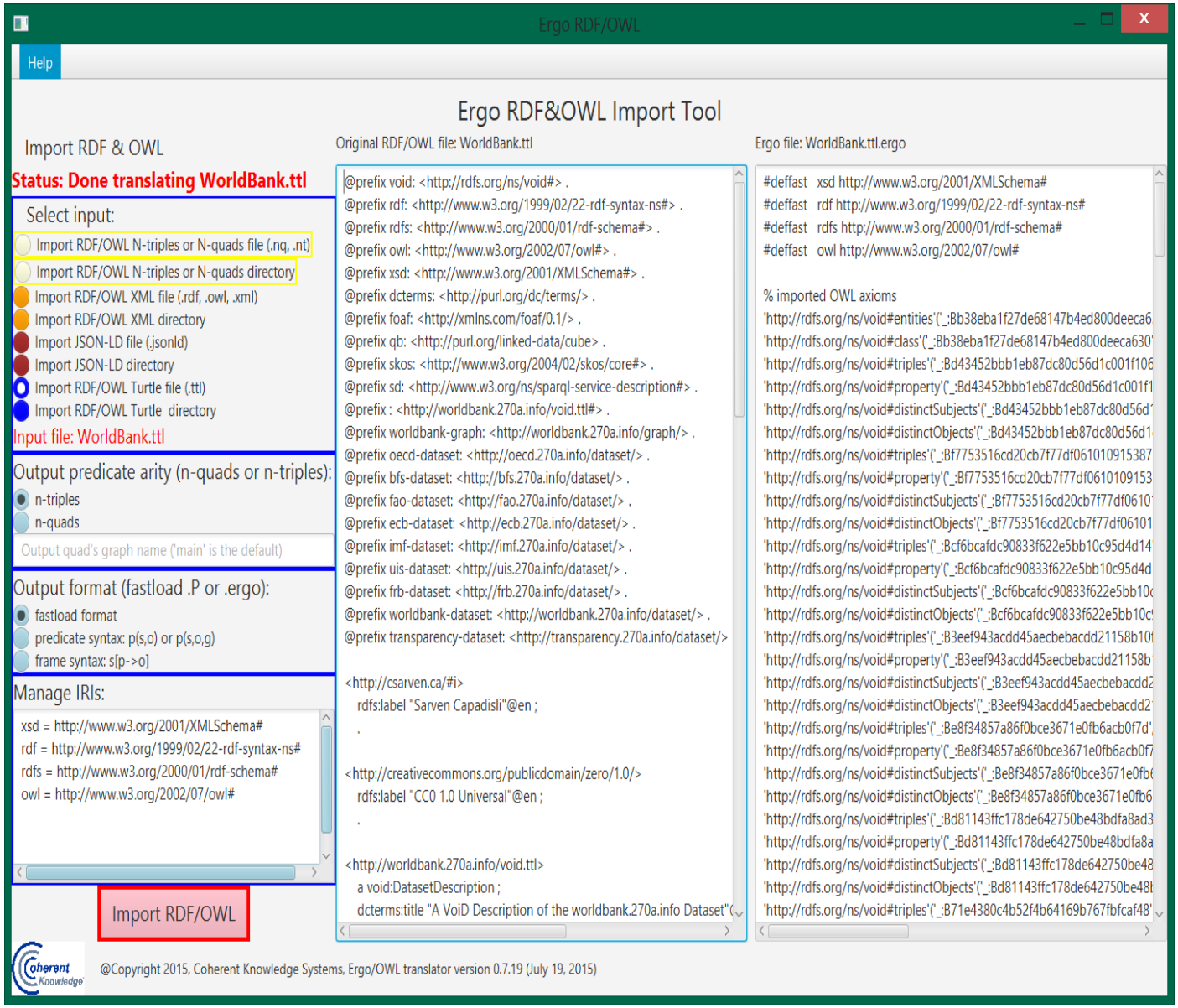
- Problem – long time (many minutes) taken to load into the Rulelog in-memory knowledge base for reasoning, when there are many rules
  - A fact is a special case of a rule
  - But often there are LOTS of facts: e.g., Millions of RDF triples
- Solution – fastloader optimization for scaling
  - Streamlines processing of facts
  - >50X speedup: seconds not minutes

Externally querying SPARQL, tightly integrating with endpoints

- Problem – How to leverage data and processing available from existing RDF triple stores, take advantage of their persistence and transactional etc. robustness
  - Often very large scale (e.g., Billions of triples)
- Solution: Ergo connector that
  - Goes out from Ergo to *dynamically* query via SPARQL to triple stores (interleaved during Ergo reasoning)
  - Translates results into Ergo and keeps reasoning
  - Uses Apache Jena libraries for translation, querying and integration of SPARQL endpoints (multiple distributed services that accept SPARQL queries and return results

Importing RDF & OWL knowledge into Ergo

Screenshot of Ergo OWL connector part of Ergo Studio



Translates RDF & OWL to Ergo

Define IRIs in Ergo Studio

N-triples and N-quads

RDF/OWL XML, JSON-LD, or Turtle as input. Predicate or Frame syntax output.

A SPARQL query

```
prefix : <http://dbpedia.org/ontology/>
prefix prop: <http://dbpedia.org/property/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

select ?film ?title ?gross where {
        ?film a :Film;
        rdfs:label ?title;
        prop:gross ?gross
        filter(langMatches(lang(?title), "EN"))
        filter (datatype(?gross) =
'http://dbpedia.org/datatype/usDollar')
}
order by desc(?gross)
limit 100
```

film	title	gross
http://dbpedia.org/resource/Harry_Potter_(film_series)	"Harry Potter (film series)"@en	7723431572
http://dbpedia.org/resource/List_of_Marvel_Cinematic_Universe_films	"List of Marvel Cinematic Universe films"@en	6317676293
http://dbpedia.org/resource/Motifs_in_the_James_Bond_film_series	"Motifs in the James Bond film series"@en	4809157447
http://dbpedia.org/resource/Pirates_of_the_Caribbean_(film_series)	"Pirates of the Caribbean (film series)"@en	3729577967
http://dbpedia.org/resource/The_Twilight_Saga_(film_series)	"The Twilight Saga (film series)"@en	3345177904
http://dbpedia.org/resource/The_Lord_of_the_Rings_(film_series)	"The Lord of the Rings (film series)"@en	2917506956
http://dbpedia.org/resource/Avatar_(2009_film)	"Avatar (2009 film)"@en	2782275172
http://dbpedia.org/resource/Transformers_(film_series)	"Transformers (film series)"@en	2669807552
http://dbpedia.org/resource/The_Fast_and_the_Furious	"The Fast and the Furious"@en	2380084668
http://dbpedia.org/resource/X-Men_(film_series)	"X-Men (film series)"@en	2304925865
http://dbpedia.org/resource/Production_of_Harry_Potter_and_the_Deathly_Hallows	"Production of Harry Potter and the Deathly Hallows"@en	2301794524
http://dbpedia.org/resource/Titanic_(1997_film)	"Titanic (1997 film)"@en	2186772302
http://dbpedia.org/resource/Mission:_Impossible_(film_series)	"Mission: Impossible (film series)"@en	2096647856
http://dbpedia.org/resource/Star_Trek_(film_franchise)	"Star Trek (film franchise)"@en	2040000000
http://dbpedia.org/resource/Jurassic_Park	"Jurassic Park"@en	2016573690
http://dbpedia.org/resource/Indiana_Jones_(franchise)	"Indiana Jones (franchise)"@en	1978055564
http://dbpedia.org/resource/The_Hobbit_(film_series)	"The Hobbit (film series)"@en	1967410881
http://dbpedia.org/resource/Men_in_Black_(film_series)	"Men in Black (film series)"@en	1653165042
http://dbpedia.org/resource/The_Matrix_(franchise)	"The Matrix (franchise)"@en	1632989142
http://dbpedia.org/resource/The_Ultimate_Matrix_Collection	"The Ultimate Matrix Collection"@en	1632989142
http://dbpedia.org/resource/The_Chronicles_of_Narnia_(film_series)	"The Chronicles of Narnia (film series)"@en	1580364900
http://dbpedia.org/resource/The_Hunger_Games_(film_series)	"The Hunger Games (film series)"@en	1555813431
http://dbpedia.org/resource/The_Avengers_(2012_film)	"The Avengers (2012 film)"@en	1518594910
http://dbpedia.org/resource/The_Mummy_(franchises)	"The Mummy (franchises)"@en	1415408499

External SPARQL query within Ergo rule

```
db1(?Film, ?Title, ?Gross) :-
        sparqlOpen(?ConnectionID, 'http://dbpedia.org/sparql'),
        sparqlQuery(?ConnectionID, 'prefix : <http://dbpedia.org/ontology/>
                prefix prop: <http://dbpedia.org/property/>
                prefix xsd: <http://www.w3.org/2001/XMLSchema#>
                select ?film ?title ?gross where {
                        ?film a :Film;
                        rdfs:label ?title;
                        prop:gross ?gross
                        filter(langMatches(lang(?title), "EN"))
                        filter (datatype(?gross) = 'http://dbpedia.org/datatype/usDollar')
                        }
                        order by desc(?gross)
                        limit 100',
                        [?Film, ?Title, ?Gross]).

?- db1(?Film, ?Title, ?Gross).    /* Test query.  Ergo outputs the answer below */
```

?Film = 'https://en.wikipedia.org/wiki/Harry_Potter_(film_series)'
?Title = 'Harry Potter (film series)'
?Gross = $7,723,431,572

Updating of Inferred Data

- Problem: When the *asserted* rules or facts data change, stored previously *inferred* facts may no longer be valid. But recomputing a large set of inferred data may take a long time.
- Solution: dependency-aware updating via "incremental tabling" extension of tabled logic programming reasoning algorithm
  - Fast edit-test loop during knowledge authoring
  - Near-real-time decision automation

Case Study 3: Defense Intelligence Analysis

- Challenge: managing, accessing, integrating knowledge
  - Huge amounts of information as text, RDF, triple stores
- Current text extraction methods:
  - Noisy/inaccurate, shallow, patchy
  - Lacks contextualization, e.g., is a date past, present, or future?
  - Events are more complex than Entities, and are only shallowly treated

Textual terminology (repeat slide)

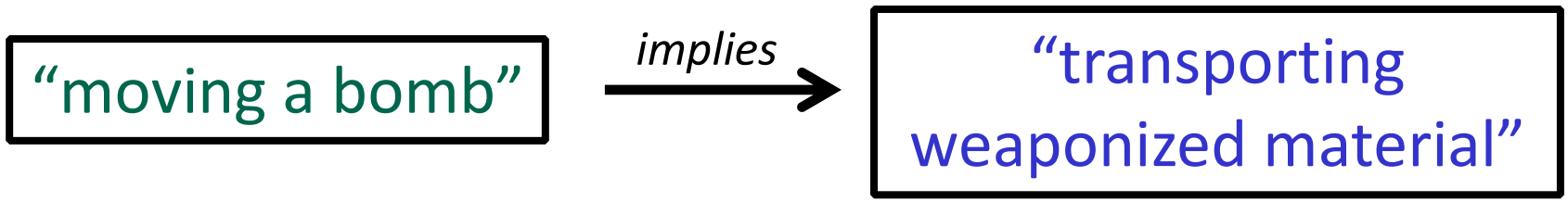
- Leverage Rulelog to much more simply and closely map between natural language and logic

- English phrase ↔ logical term in Rulelog
- English word ↔ logical functor in Rulelog

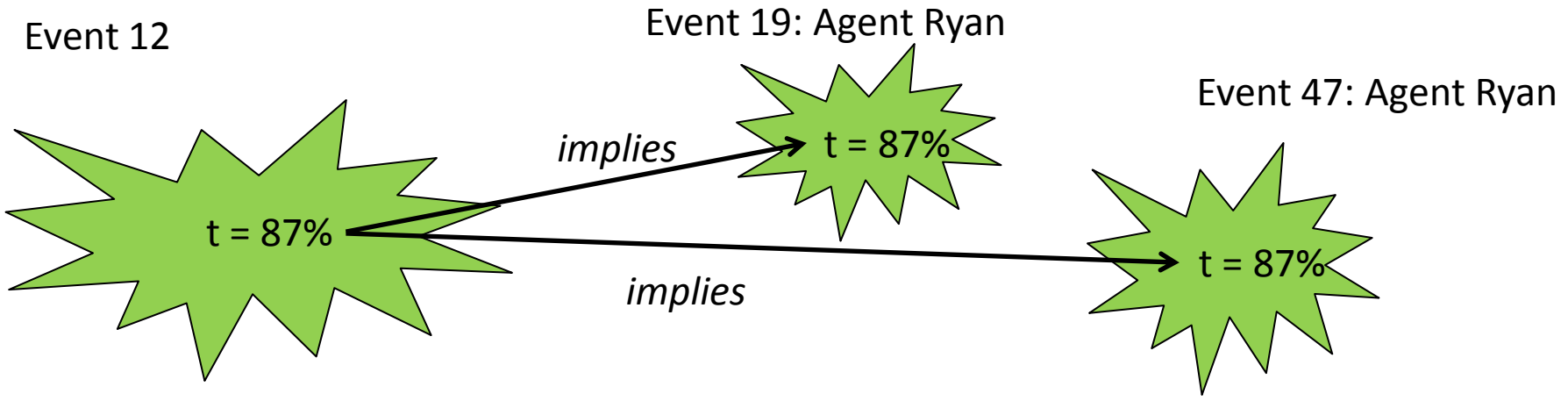
- Basis for textual templates

Rulelog enriches Text Extraction

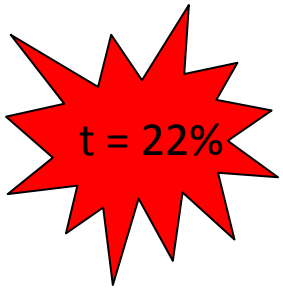
- Leverage Rulelog's high expressiveness and flexibility
- Mappings between multiple terminologies or ontologies



Rulelog adds Context to Text Extraction



Event 95
Agent Ryan



- Knowledge can be updated
- Knowledge is defeasible
  - prioritization for conflicts

Priority of Event 95 > Priority of Event 12

Therefore, Agent Ryan is no longer credible

Outline

- What
- How
- Case Studies
- Windup

Lessons Learned from Case Studies

Financial, E-commerce, and Defense customers benefited from:

- Agility: Flexibility and ease of authoring, fast updating
- High accuracy and transparency
  - Explanations and provenance
  - Lower risk of non-compliance or confusion
- More Cost Effectiveness – less labor, SMEs in closer loop
- Improved volume, velocity, variety of data – with deep reasoning
  - Optimized loading of millions of facts (e.g., triples) as input assertions
  - Optimized dependency-aware updating of millions of inferred facts
  - External dynamic querying of triple stores
  - Mapping text ↔ logic using Rulelog methodology, e.g., for terminology
- Leveraging investment in Smart Data tech: RDF, SPARQL, OWL

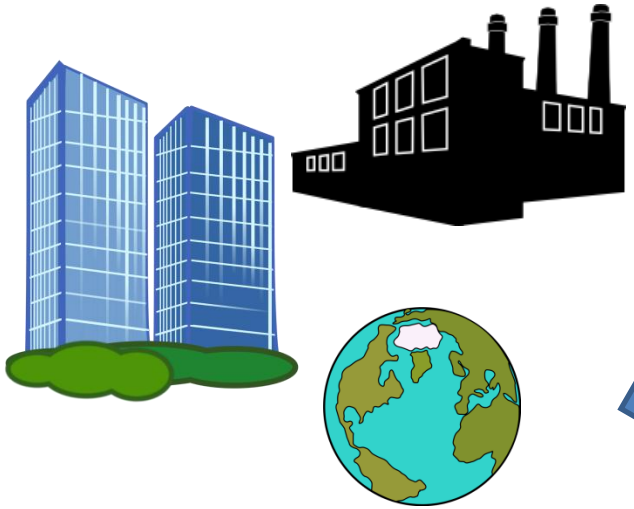
“Smart Rules for Smart Data”, unpacked

- The smart rules work with smart data as **input** assertions
- The smart rules **infer** smart data
- The smart rules are fully declarative/semantic and themselves **become** rich data not procedural code
- The smart rules overall **leverage** investments in smart data

- Coherent helps customers develop and deploy smart rules KBs and applications, for a range of tasks and domains
  - Via its Ergo platform **product** capabilities
  - Via its professional **services**

Evolution of Enterprise Knowledge Management

(repeat slide)

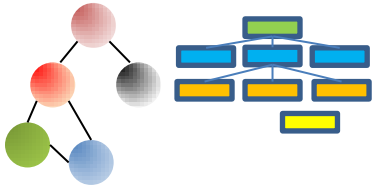


Data integration with flexible schemas, more meta-data. E.g., RDF, SPARQL, OWL.

Augment relational/traditional data stores.

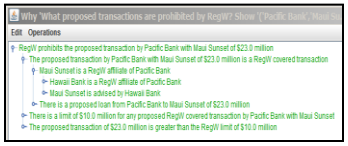
Smart Data

Graph/Linked Data bases



Smart Rules

Decisions & Analysis



Deep reasoning, more complex knowledge. E.g., Textual Rulelog.

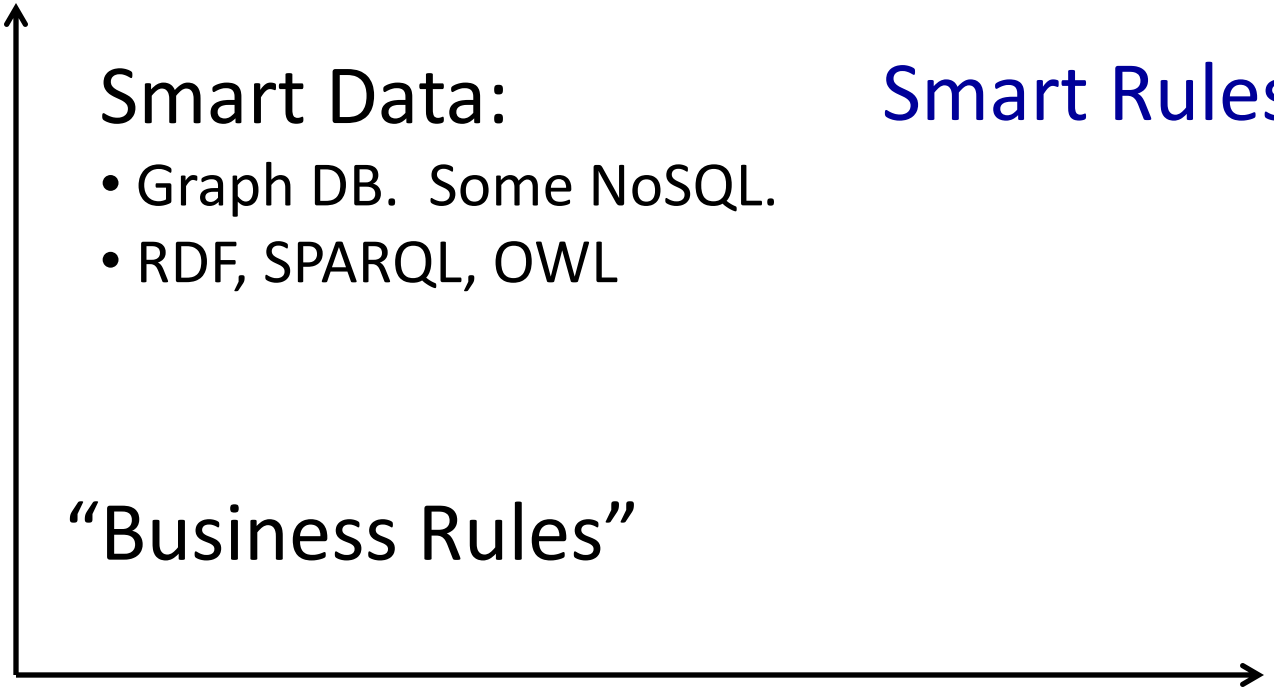
Leverage Smart Data investment.

Thank You.

Smart Rules for Smart Data™

OPTIONAL SLIDES FOLLOW

Market Evolution



Ergo Suite – The Coherent Knowledge Platform

- Dramatically expands the capabilities of database and reasoning systems
  - Adding or updating assertions, and posing queries,
    is much easier, faster, cheaper, and more under user control
  - An advanced logic engine operates under the covers. Handles probabilistic too.
  - Full explanations in English are provided,
    exposing the context and meaning behind the results
    - Every relevant assertion is a step in the chain of reasoning that leads to the final answer
- Benefits automation of:
  - Policies: organizational, compliance, and legal
  - Decision making: routine, exceptions, alerts
  - Learning: interactive tutor, in-depth explanation of solutions
  - Information Access: fine-grain control and tracking
  - Information Analysis: including collaboration and scenarios
  - Information Integration: from diverse sources, using structured info and text

Lessons Learned from Case Study 1

- Coherent's Ergo Suite technology successfully automated Regulation W, demonstrating its utility for Regulatory and Policy Compliance
  - Highly Accurate on test data
  - Full Explanations – in English, with chain of reasoning and provenance
  - Reduces key elements of compliance risk
  - Cost Effective implementation – flexible; with electronic audit trail
- Concrete Business Benefits for Financial Compliance
  - More Cost Effective – less labor, subject matter experts in closer loop
  - More Agile – faster to update
  - More Overall Effectiveness – firmer deeper understanding
    - Lower risk of non-compliance or confusion

Case Study Lessons – Bigger Picture

- Knowledge work by professionals revolves largely around continuing education (a.k.a. training)
  - Need to cope with ever-growing info amounts, complexity, and expectations
- The customers were very excited by the availability of comprehensible detailed explanations
  - Compliance non-IT people could understand them – and validate decisions
  - Analytics without sufficient explanation/transparency is hard to trust, hard to use, and hard to learn from, individually and organizationally
- Knowledge work in turn revolves around orchestrating and integrating multiple knowledge sources and analysis components.
  - Coherence and synergistic power in combining are critical
- Textual Rulelog meets these requirements well
  - Flexible, expressive, semantic, open, transparent

E-commerce marketing: Pricing & Promotions

- Example 1: Loyal customer gets 5% discount

```
price(of(item(?i))((for(customer(?c))(is(?p)) :-
    normal(price)(of(item(?i))(is(?pn)) \and
    ?c(is(a(loyal(customer)))) \and
    ?p \is ?pn * (1 - 0.05).
```

- Example 2: New customer gets $20 coupon on first $100 of purchases
- Via API, query Ergo as business logic within application managing overall customer experience

END of OPTIONAL SLIDES