Asset-backed Securities

The Advantages of a Declarative Rule-based Solution
August 2nd, 2010

Contact:
Peter Lin: peter.lin@concordusa.com
Said Tabet: stabet@comcast.net

CONCORD

Agenda

1. Background
2. Business Users and Rule-Based Solutions
3. Concord's Recommendation – RuleML

Background

The SEC is proposing changes to self reporting procedures for asset-backed securities (File Number S7-08-10). The proposed changes will drive financial institutions away from a rating-based approach to asset-backed security to a more comprehensive measurement-based approach. The SEC has distinct goals for these changes and high-level requirements.

Goal of the proposed changes

- Give financial advisors, fund managers and consumers the ability to evaluate Asset-backed securities more accurately in order to simplify and quantify how those securities are measured.
- Increased transparency with the data that is filed and the level of detail in the filings.

The SEC requirements

- Improve the forms that the SEC consumes from financial institutions.
- Required financial institutions to build custom programs that support the changes.
- Drive an overall change from a rating-based securities approach to a measurement-base approach.

Programmer Centric vs Business User Centric

For a solution to succeed, we must consider business users. Business users have different needs than software engineers and programmers. A Declarative Rule-based approach to solving the SEC goals puts more control into the hands of the people that actually use the system – the Business User. This is especially important for system usability, cost containment, and functional change requests .

Programmer-Centric Approach

- Requires an understanding of the specific programming language that the SEC recommends. Sourcing of that talent and cost containment both become a potential problem.
- When errors occur, the user has to engage a programmer to debug the problem.
- Functional changes that are introduced any time after deployment require a re-engagement of programmers to manipulate the application.

Business User-Centric Approach

- Business Users can make functional changes without needing to re-engage the programmers.
- When errors occur, the system provides a business user friendly explanation with instructions and tips.
- Enables automated validation and formal proof.

CONCORD

The Advantages of a Rule-Based Solution

The best way to realize a Business User-centric solution is to leverage the power of Business Rule technology. Utilizing declarative rules, rule engines, test drivers, proof checkers, and English proof explanation, the SEC can leverage a Rule-Based solution that streamlines the process for Business Users and utilizes automation to improve efficiency.

- **Easier for the Business to Use.** Rule-based solution presents information in natural language format, which makes it easier for non-technical users. A common set of terms, concepts and rules provide consistency and reduces errors.

- **Validation.** Using proof checkers and rule engines, the system can validate the mortgages in a mortgage-backed security.

- **Calculation changes are not programmer-dependent.** Equations used to calculate the potential return are defined separately (externalized) instead of being embedded in a program. This makes it easier for the business user to validate, reuse, modify, and substitute the equation in a consistent/transparent way.

- **Proactive and Intuitive.** When errors are detected, the system can give explanations with adjustable level of detail and friendly tips. This is important for traceability and consistency of the decision points.

- **Error Reduction.** With a rule-based solution, financial institutions can analyze the rules, check proofs, and catch errors earlier in the process. This will reduce the occurrence of errors, and reduce the burden on SEC staff.

- **Easier to Audit.** Following Best Practice rule management methodology makes it easier to audit the rules and look at the revisions. Instead of guessing when an error occurred, users can look at the history and find the problem sooner with less work.

CONCORD

Recommendation

Concord recommends a solution that is Business User-centric versus Programmer-Centric. In order to achieve a Business User-centric solution, we must use Business Rule technology. RuleML is our recommended technology framework because it permits the XML-based interchange of Business Rules, is easy for the Business User to work with, and has an Active Developer Community with strong ties to OMG, W3C, and OASIS.

RuleML is a Framework

- Control costs - Frameworks make it possible for market competition (between rule editors, engines, etc.) to exist, thus reducing costs of deployment.
- Drive standards – A common framework that drives standards will be necessary for the SEC to assure successful solutions amongst all the financial industry players.
- Decreased time to market – Frameworks lay the foundation that organizations can build upon. The better the framework, the faster the build process.

RuleML is a Business Users "programming language"

- Declarative language – The Business User can "declare" in Controlled English what his/her rules are going to be. No need to be a programmer.
- Supports formal logic and enables proof – The Business User can test their rules and prove them logically correct.
- Separates model from data – As requirements change over time, the Business User can change models by modifying their rules separately from any data changes, without need for a programmer to disentangle the business logic from the data.

CONCORD

Recommendation

Using RuleML framework will give the SEC the option of publishing changes in RuleML format, which financial institutions, fund managers, financial advisors and traders, investors in general can use without the need for a software programmer. This will reduce the cost of implementing changes for financial institutions.

Publishing Changes in RuleML

- Reduces costs – For many financial institutions, it takes about a year to implement changes in regulation, resulting in high maintenance overhead.
- Improves quality – Many financial institutions and managers struggle to understand regulations, which results in different interpretations and implementation.
- Decreased time to market – Publishing changes in RuleML format provides a way for fund managers, financial advisors and traders to see the impact of the change quickly and make decisions in a timely manner.
- Simplified and integrated audit - The audit process is generally a complex and costly activity involving resources at different levels. Using RuleML framework leads to better reconciliation of the changes and differences between the company's view and the regulators, and simplifying the integration with their internal systems.

CONCORD

Asset-backed Securities

Appendix

CONCORD

Agenda - Appendix

1. Comparison of Technologies
2. Python and other Languages
3. RuleML
4. Use Cases
    - Change in Forms
    - Self Reporting

CONCORD

Comparison – Programming Languages

To realize a declarative rule-based approach, the language must have four attributes: declarative, support formal logic, enable proof of the application and have a well defined standard.

Language	Declarative	Formal Logic	Enables Proof	Defined Standard
Python	✗	✗	✗	✗
Java	✗	✗	✗	✓
C#	✗	✗	✗	✓
C/C++	✗	✗	✗	✓
Ruby	✗	✗	✗	✗
RuleML	✓	✓	✓	✓

CONCORD

Typical Approach – Python and Other Languages

Traditional software development approach is iterative/procedural and often involves separate teams. It is good in a tightly integrated environment where teams work closely with a unified development process and schedule.

- Business analysts gather business requirements
- Developers review the requirements with business analysts
- The work is divided into tasks and assigned to developers
- Developers implement the functionality and run unit tests
- When the components are ready for integration, the developers work together on integration and resolve any issues
- When the application is ready, it is deployed to a testing environment
- Quality assurance team runs through a series of tests and report bugs
- Developers fix the bug and queue changes for next round of testing
- Manager decides when a release is ready
- New release is created and pushed to production

CONCORD

Pros and Cons of Python

Although Python is popular, an objective analysis of the language is critical in the selection process

Pros

- Python is Open Source and available to everyone
- The language is popular
- There's a large community of developers

Cons

- Python does not have an official language specification
- Implementation of Python in Java and .NET do not guarantee compliance, since Python does not have a formal certification process to validate implementations
- Python uses a dynamic type system, which makes it more difficult to audit and validate programs
- Not a declarative language, which is very important especially in this context. Python is procedural.
- Imperative languages do not provide proof the program is accurate and true

CONCORD

Ideal Approach – RuleML

Using a rule-based approach is more friendly for business users and formalizes the filing process. This makes it more efficient and provides the following benefits.

- Easy for business users to use and understand
- Provides detail explanation of errors
- Provides suggestions for fixing issues
- Provides a way to validate the business rules
- Provides a way to validate the data
- Makes it easy to automate
- Makes it easy to audit
- Uses a well defined standard, which is vendor neutral
- Clearly defines the vocabularies and terms so that users have a common understanding
- Provides a clear audit trail
- Uses a declarative approach

Pros and Cons of RuleML

Like all technologies, RuleML has strengths and weaknesses.

Pros

- RuleML is a declarative language http://www.ruleml.org/
- RuleML is a modular language. It uses sub-languages to support for different types of logic and rules as well as extensibility.
- Supports formal logic. Developed with the participation of leading expert researchers and practitioners as well as leading organizations worldwide
- Open standard with active community including business community and academic research. Major vendors participate in the standards effort, product built around RuleML in various verticals
- Rules, functions, data and model are divided into components

Cons

- PRR 1.0 specification was released in December 2009 http://www.omg.org/spec/PRR/1.0/index.htm
- Commercial support for PRR 1.0 isn't on the market yet
- Requires discipline and expertise to build systems with RuleML

Typical Business Use Case – Change in Forms

The proposed changes for asset-backed securities can be summarized by the following bullet points.

- Securities Act Rule 415 will be amended
- S1 and S3 form will be phased out for Asset Backed securities
- New form SF1 and SF3
- New forms for self registration require the filer to prove the following
    - The assets in the pool will produce the cash flow described in the prospectus
    - Risk retention
    - Party obligated to repurchase the asset for breach of representation
    - Issuer will file exchange act reports
- Detailed information about each asset in the pool must be provided
- Python program is proposed to be part of the new self registration report

CONCORD

Solving the problem using RuleML

RuleML uses proven techniques from twenty years of real world rule projects and provides the following benefits

- Define a base model which users can extend
- Define the calculations as built-in functions
- Define the business rules
- Define the terms and vocabularies
- Enables the user to prove the filing is accurate
- Enables the user to modify the asset pool data and re-evaluate the security
- Enables the user to validate each asset in the pool meets the defined quality
- Enables the user to validate the cash flow for each asset pool is accurate
- Provides a detail explanation of non-compliance
- Automate validation and reporting
- Maintain a detailed audit trail

CONCORD

Python in action – Self-reporting Use Case

Although Python is popular, an objective analysis of the language is critical in the selection process

Pros

- Python language is powerful and flexible
- It provides all the functionality needed to write the program specified by the proposal
- Java and .NET both have third party libraries that support Python syntax

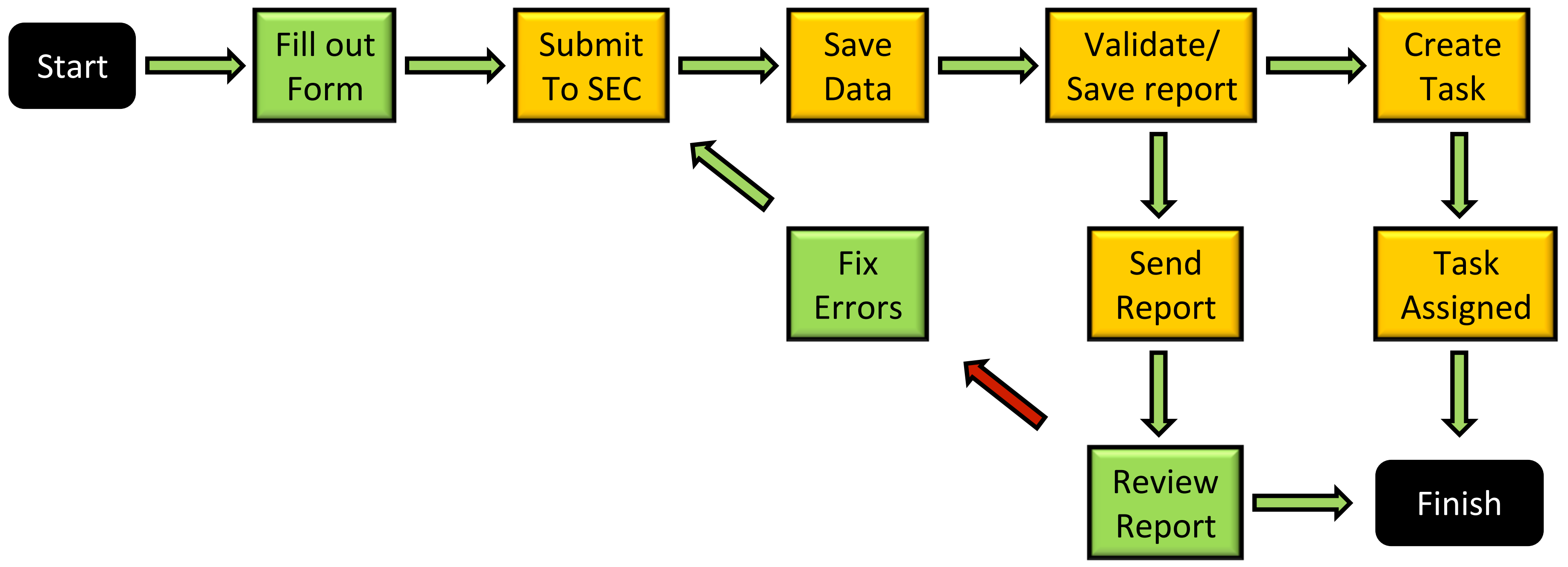
Cons

- No assurance Java and .NET implementation are 100% accurate compared to Python written in C.
- Python language is not govern by a standards organization and does not have a formal validation process
- Python uses a dynamic type system, which makes it more difficult to audit and validate programs
- The proposal does not define a common model
- Without a formal approach, the quality of the python applications will vary and create more work for the SEC staff

CONCORD

Self-reporting Validation Workflow

The SEC can improve efficiency by employing automated validation as part of the workflow. This would give financial institutions faster responses and more details to amend their filing. It would also create the appropriate task, which streamlines the internal process.



Use Case – mortgage backed security example

Using a declarative business rule approach, the contents of the filing would be organized by rules, calculations, mortgage data and report details.

Built-in Calculations

- Weighted Average Maturity
  - Loan weight = outstanding amount / total outstanding of mortgage pool
  - WAM = Sum(loan weight * months to maturity)
- Weighted Average Coupon
  - WAC = Sum(loan weight * loan rate)
- Total outstanding amount for each pool = Sum(outstanding amount of the mortgage)

Mortgage Pool Analysis

- Calculate the cash flow and compare against the stated value provided by the filer
- Examine the prepayment rate for the security and calculate risk
- Calculate risk of default for each mortgage
- Calculate an aggregate risk of default for the pool

Validate the filing is correct

- The business rules are correct
- The calculations are correct
- The mortgage data is complete and correct
- Execute the business rules and compare the results with expected cash flow

CONCORD

Use Case – mortgage analysis

The system would have rules that detect subprime loans, increase risk of default and estimate the impact on the mortgage backed security.

- If the credit rating of the borrower is below 640, then calculate risk

- If the social security number of the borrow is not valid, then report compliance error

- If the total risk for the mortgage pool is above the threshold, then report compliance error

CONCORD

Use Case – Common Terms

An important part of a declarative rules based approach is defining the common terms. This establishes the definitions clearly and how it is used by rules, calculations and self-reporting forms.

- **Loan-to-value (LTV) =** the amount of the loan compared to the value of the property
- **Debt-service coverage ratio (DCR) =** (Annual net income + Amortization/Depreciation + other non-cash and discretionary items) / (Principal repayment + interest payments + lease payments)
- **Loan weight =** is the weight of the mortgage compared to the entire pool. Calculated by dividing (outstanding loan amount / total outstanding amount for the pool)
- **Weighted Average Maturity =** the average maturities of the mortgages in the pool. Calculated by adding (loan weight x months to maturity)
- **Weighted Average Coupon =** the average of the coupons of the mortgage pool. Calculated by adding (loan weight x mortgage rate)

CONCORD

Glossary of Terms

A definition of the technical terms used in the presentation are provided for convenience.

•**Rule-based approach** – A method of building software applications where the business rules are externalized from application code. This allows business users to change the business rules without engaging a software engineer and going through the entire software development lifecycle.

•**Declarative rules** – Declarative rules define the business rules as logical statements. Declarative rules do not define how the rules should be evaluated or the order they should run unlike imperative or procedural systems.

•**Business Rule Engine** – A software application built to run business rules.

•**Test Drivers** – A software application built for testing and validating business rules are correct. This is means loading data into the rule engine and executing the rules. The actual results are compared to the expected results.

•**Formal proof** – Given a set of facts and rules that are logically valid, with or without the help of a theorem prover (a computer program), one can validate and verify that the inferences or conclusions that have been made are also true within this context

•**Natural Language** – Rules are presented in a controlled English, instead of a programming language.

CONCORD

CONCORD

509 2nd Avenue South | Hopkins, MN 55343 | 952.241.1090
www.concordusa.com