FIBO™ Build, Test, Deploy and Maintain Methodology

Dennis E. Wisnosky and the FIBO Leadership Team
12/10/2014

This document is not intended to be the definitive guide for FIBO Build, Test, Deploy and Maintain. It is not the compendium of all material. Rather it references the most current URLs containing detailed instructions, files, etc.
Table of Contents

1.0  Introduction ............................................................................................................................................. 3

2.0  FIBO Ecosystem ....................................................................................................................................... 5
  2.1  Collaboration Requirements .................................................................................................................. 6
  2.2  Quality/Testing Requirements ............................................................................................................. 6
  2.3  Language requirements ......................................................................................................................... 6

3.0  FIBO Infrastructure .................................................................................................................................. 8
  3.1  EDMC - Business Conceptual Ontology/OMG Standards Process ..................................................... 8
  3.2  FIBO Teams .......................................................................................................................................... 10
    3.2.1 FIBO Leadership Team (FLT) ........................................................................................................ 10
    3.2.2 FIBO Content Teams (FCT) .......................................................................................................... 10
    3.2.3 FIBO Proof of Concept Teams (FPCT) .......................................................................................... 12
    3.2.4 OMG Task Force (OMG/TF) ......................................................................................................... 12
    3.2.5 FIBO Implementation Teams (FIT) ................................................................................................. 12
    3.2.6 FIBO Process Team (FPT) ............................................................................................................ 13
    3.2.7 FIBO Vendor Team (FVT) ............................................................................................................. 13

4.0  FIBO Process ............................................................................................................................................ 14
  4.1  GitHub .................................................................................................................................................... 17
  4.2  Jenkins .................................................................................................................................................. 19
  4.3  JIRA ....................................................................................................................................................... 19
  4.4  Use Cases ............................................................................................................................................... 20
  4.5  Testing .................................................................................................................................................... 22
    4.5.1 Modeling and Unit Testing .............................................................................................................. 23
    4.5.2 Validating Through Integration Testing ......................................................................................... 23
    4.5.3 Documenting .................................................................................................................................... 24
    4.5.4 Deploying ......................................................................................................................................... 24

5.0  Roles ......................................................................................................................................................... 24
FIBO\textsuperscript{1} Build, Test, Deploy and Maintain Methodology

1.0 Introduction

FIBO is a family of RDF/OWL ontologies developed as an open standard by the EDM Council in close cooperation with the Object Management Group (OMG). In order for FIBO to become a widely accepted standard to be implemented in many systems across the industry it is important to prove and demonstrate FIBO ontologies as they move through the standards process and become available to the industry. Important activities necessary to make that happen include:

- Create a FIBO community including ontologists, vendors, enterprise architects, business architects, technical architects, programmers, database specialists and subject matter experts from among the financial institutions, the regulators and their consultants and services providers.
- Create an environment including a platform and a methodology where this FIBO community can collaborate in a structured process based on “Continuous Improvement”\textsuperscript{2} Agile principles and Model Driven Architectures™\textsuperscript{3}.
- Work within a defined business process model that embodies automatic testing and results in FIBO being ratified as an International Standard.

These objectives provide some detailed challenges to the FIBO infrastructure evolution.

The FIBO Leadership Team (FLT) is a small, highly specialized Leadership Team of subject matter experts in financial services, standards process, ontology and architecture. But in order for FIBO to gain the acceptance that will be necessary for its continued success, it must have the participation of a wide range of contributors with a wide variety of skills and backgrounds. FIBO Content Teams (FCT) and the participation of industry partners through Proof of Concept Teams (PCT) will fulfill this role.

Successful open source projects provide a model that FIBO can follow. In these projects, material can be accessed and adapted for a person, or an organization’s, own use. The most successful (i.e., reusable and valuable) adaptations can be fed back into the standard for others to use. Thus, the FIBO Leadership Team’s role is maturing from being the creators of the standard to being the curators of its ongoing development.

FIBO is viewed as being very similar to the build of a complex, open-source software system. Ontologies are treated in the same way as “sources” like Java or C++ with the same “challenges” such as managing dependencies and merging differences. Modern software engineering principles can therefore be

\textsuperscript{1} FIBO is a Trademark of the Enterprise Data Management Council (EDMC)
\textsuperscript{2} Including “Continuous Integration”, “Continuous Delivery” and “Continuous Deployment”
\textsuperscript{3} Model Driven Architecture is a Trademark of the Object Management Group (OMG)
applied to the process of OWL ontologies such as “Behavior Driven Development” (BDD)/“Test Driven Development” (TDD) “Continuous Integration” and “Continuous Improvement.” Additional principles specific to ontology development include adherence to modeling policies and best practices, testing for logical consistency and testing for deductive closure using a representative knowledge base.

These practices are highly dependent on automation that tends to merge Develop, Build, Test, Deploy, and Maintain into a continuous process. This is the path that FIBO follows, wherein:

**Build** is the process of fleshing out FIBO as both UML and RDF/OWL models using OMG and W3C Standards, respectively. FIBO will have many builds. Initial builds are Foundations, Business Entities and Indices/Indicators in process and Equities and Loans beginning the process.

**Test** includes unit tests, integration tests, stress tests and instance data tests that are automatically run against the ontologies and related artifacts regularly to ensure conformance to requirements.

**Deployment** for FIBO began when the initial FIBO Conceptual Ontology (BCO) began to be formed in 2008 by a team of Subject Matter Experts. The FIBO BCOs are UML/OWL models grouped in a taxonomy [http://www.edmCouncil.org/semanticsrepository/index.html](http://www.edmCouncil.org/semanticsrepository/index.html) by types of Financial Instrument. Included is the definition of each concept and how each concept is related. FIBO BCO concepts are being used in the industry today. FIBO deployment will also include various FIBO Operational Ontologies (FOO) that are extensions of BCO OWL models customized by an operator to support the business of the operator. In these deployments FIBO could become a database, be integrated with reasoners, be offered as a web service or within a service or some other particular architecture.

**Maintenance** is continuous when issues are found in the FIBO Domain Ontologies during build, test and deployment, and as the need for extension occurs as a natural part of building out an increasingly broader family of specifications. FIBO automation, described in detail in this document, extends maintenance to all FIBO participants in the industry for both fixes and enhancements.

The FIBO is a grand vision; it is a standard that will be used across the financial industry for decades. It will grow and adapt as the years go by and user demands change. It is therefore crucial that the FIBO Build, Test, Deploy and Maintain process have clear and unambiguous goals:

**For Build:**

- Securely and unambiguously manage all FIBO content
- Automatically perform any needed conversions into the FIBO publication-ready form
- Document and measure quality metrics for FIBO publications

**For Test:**

- Create full unit test coverage for all the FIBO ontologies (where a unit test is basically a use case represented as one or more SPARQL statements, that can be invoked automatically, as queries against a knowledge base that includes a representative set of individuals, in addition to the ontologies, reflecting the use case)
• Ensure that these unit tests (including the SPARQL queries and representative OWL individuals) are executed on as many vendor platforms as possible, enabling vendors to contribute to the further Build of FIBO and allowing them to demonstrate their technology and how FIBO can be applied to the community and the industry as a whole
• Run use case/SPARQL unit tests and other tests such as OWL validators, reasoned-related tests, performance and volume tests

For Deploy:
• Create a demonstration platform where FIBO can be shown with several standard datasets
• Create a documentation platform where documentation for the various audiences is automatically generated from the ontologies and optionally also from the datasets (showing visualizations of both ontology classes as well as individuals)
• Work with early adopters in financial institutions that have agreed to provide sample deployments that show how FIBO can be integrated into real processes in the industry
• Create a documented set of deployment best practices based on this experience

For Maintain
• Provide for life cycle configuration and issue management
• Encourage deployment teams in the industry to submit change requests through a repository infrastructure
• Use automated tools to rationalize or refactor as necessary

2.0 FIBO Ecosystem
Achieving these objectives requires a FIBO Ecosystem with the characteristics shown in the table

<table>
<thead>
<tr>
<th>A collaborative environment for modelers to share and update models.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Modelers can work in any tool stack that outputs valid OWL/RDF (Protégé, TopBraid Composer, Franz Gruff, etc.), including ODM/XMI—based tools (MagicDraw, VOM, EA etc.)</td>
</tr>
<tr>
<td>• Models are managed in GitHub and stored in a consistent format regardless of source, for superficial (line-by-line) comparison and version management.</td>
</tr>
<tr>
<td>• Automated testing managed by Jenkins and links to issue management in JIRA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A business-facing web presence to enable FIBO users to look up terms, their definitions and relationships viewable in tabular and standard FIBO graphical formats.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Affiliated vendors provide auxiliary services</td>
</tr>
<tr>
<td>• Testing coordinated through Jenkins</td>
</tr>
<tr>
<td>• Reports business-facing displays and any other services</td>
</tr>
<tr>
<td>• Publication and re-distribution including Reasoners and SPARQL endpoints</td>
</tr>
<tr>
<td>• Affiliates manage their own software to avoid technical and license issues</td>
</tr>
<tr>
<td>• Axiom- and Entailment-level diffs for understanding the impact of model changes.</td>
</tr>
</tbody>
</table>
2.1 Collaboration Requirements
FIBO demands a process that allows practitioners of varied backgrounds from the finance industry, their consultants and service providers to view, review, and contribute to FIBO. The standard is free, open and available to financial institutions, regulators, ratings agencies and other players in the industry.

FIBO cannot be developed by the heroic efforts of a single person or even of a single dedicated team. It will require review and contribution from the whole community. The FIBO infrastructure must therefore allow for the free dissemination of the ontologies as well as a means for accepting feedback.

2.2 Quality/Testing Requirements
FIBO will be used as a ‘best practice’ reference throughout the industry. This means that FIBO publications will be held to a high standard of scrutiny for quality. A job of the FIBO Leadership Team (FLT), defined below, is to enforce high standards of quality for the published models, ontologies and specification documents including strict configuration management.

FIBO Build, Test, Deploy and Maintain will be done in any of a wide variety of tools governed by strict EDMC and OMG guidelines. These guidelines are well documented, and, to the extent possible, supported by automated tools. The FLT is able to determine quickly and accurately whether a submission satisfies the FIBO quality requirements through the use of an open source utility known as Jenkins, which is described in some detail below.

2.3 Language requirements
Some of the most exacting requirements are on the language(s) in which the FIBO ontologies are to be published.

Since the FIBO will be used for decades, any language used to express FIBO is required to have the backing of a major standards organization, and to be quite stable.

FIBO ontologies are published in OWL 2⁴, the World Wide Web Consortium (W3C), an international standards body, recommendation for sharing ontologies on the web.

One of the design requirements of OWL as an ontology language for the web is that ontologies in OWL are interoperable with other distributed ontologies. Since the ontologies can be queried with a standard query language (SPARQL), it is possible to transform OWL ontologies, by virtue of their representation in RDF, into and from other modeling systems. This makes OWL a good choice for an interoperable ontology language. In particular, highly skilled architects and ontologists can use their own tools to perform high-volume and high-accuracy ontology updates, and to transform the result into W3C-standard OWL for sharing with the FIBO community.

By virtue of its representation in RDF, OWL can be read, processed and produced by a wide variety of tools, both commercial and open source. The free and open source tools are particularly relevant to this

⁴ See [http://www.w3.org/standards/techs/owl#w3c_all](http://www.w3.org/standards/techs/owl#w3c_all) for the set of documents comprising the OWL 2 Language Recommendations.
discussion. For desktop review and editing of OWL ontologies, open source tool Protégé is the tool of choice. Others, include the free version of TopBraid Composer, a commercial tool from TopQuadrant.

Participants in the OMG’s standardization process may also be able to obtain free licenses to the corresponding UML tools for FIBO, using the Ontology Definition Metamodel (ODM)\(^5\) standard profiles for RDF and OWL in UML. These include, but may not be limited to, No Magic’s MagicDraw UML tool and Visual Ontology Modeler (VOM) plug-in. There are several open source databases and APIs that can read and process queries over OWL, including the OWL-API, RDFLib, Jena and Sesame, as well as databases OpenLink Virtuoso and Systap BigData. Inferencing capabilities for OWL include open source (Pellet, Hermit) and commercial tools, many with free editions (C&P’s Stardog, OntoText’s GraphDB, SPARQLverse) Commercial database products include AllegroGraph, MarkLogic, Clark and Parsia’s Stardog, Oracle and IBM. Adaptive is providing web-based access to FIBO for search and graphical exploration.

FIBO must accept contributions from a wide range of tools, and it is imperative that the FIBO curators be able to tell what has changed from one version to a proposed next version. The best understood and most successful way to achieve this is through a wide variety of tools that can provide text-level diff and merge capabilities. This means that FIBO must be published in a text format that is stable, i.e., small changes in an ontology result in small (localized) changes in the textual representation. FIBO therefore uses the RDF/XML output of Protégé as the standard publication format for its ontologies. FIBO provides a command-line Serializer that converts RDF/XML from any source into a stable format for storing in version control systems (GitHub). This conversion happens automatically when a model is committed to the EDMC repositories. The Serializer is available for anyone to use to make their format comparable to EDMC baselines.

In addition to the OWL 2 language requirement, the standards published by the Object Management Group (OMG) include normative UML diagrams conforming to the ODM standard, as mentioned above. Thus, any UML tool that supports the ODM should be able to import the FIBO OWL ontologies, facilitate documentation via diagrams that conform to the ODM, edit in ODM/UML, and support export conforming to the RDF/XML serialization of OWL. The FIBO standards include the normative OWL 2 representation in RDF/XML and normative diagrams in ODM/UML, in order to facilitate understanding and interoperability with a broad range of tools and repositories.

Certain well-known modeling axioms cannot be represented in OWL. These include role intersection rules and issues with sameAs. This means that FIBO will require a language for modeling axioms that go beyond the capabilities of OWL. FIBO intends to use RIF, the W3C Recommendation for Rules Interchange on the web. RIF will allow users of FIBO to execute axioms using their own preferred semantic rules execution protocols, which may include SWRL, SPIN, SPARQL, Rulelog, TRee and others.

3.0 FIBO Infrastructure

3.1 EDMC - Business Conceptual Ontology/OMG Standards Process

FIBO development began in 2008 at the onset of the Global Financial Crisis. Since then, 29 Finance Industry Domain Business Conceptual Ontologies have been developed in the Unified Modeling Language standard (UML) of the Object Management Group (OMG) and are available at no cost at: http://www.edmCouncil.org/semanticsrepository/index.html

These FIBO Domains were built through a series of online sessions with the Council’s “Head of Semantics and Standards” leading various Subject Matter Experts within the industry. These 29 FIBO Domains are listed in their various states of development at the following URL: http://www.edmCouncil.org/downloads/20140310_FIBO_Development_Scenario.pdf

It is from this list that FIBO Content Teams (FCT) described in Section 2.3 will select a particular Domain to enter the OMG standards process. In 2013 FIBO Foundations (FIBO-FND), was sufficiently developed to enter the OMG process to become the first FIBO standard. FIBO for Business Entities (FIBO-BE) and others, are following on approximately a quarter-by-quarter schedule. In this process each FIBO is tested in its OWL format for mathematical correctness, is repeatedly reviewed by both Subject Matter Experts (SMEs) and ontologists, is transformed to comply with OMG documentation requirements, is voted on by the OMG Financial Domain Task Force and Architecture Board and is published as an official OMG Standard. This EDMC/OMG process is outlined in Figure 1. A more extensive treatment of this process is shown in http://www.edmcouncil.org/downloads/20140310_FIBO_Development_Scenario.pdf which also shows the status of each FIBO according to the color code explained below.

![Figure 1 - FIBO EDMC-OMG Development Process](image)

The colors in this figure are used to illustrate the changing maturity of each FIBO as it is transformed by this process. Initially all FIBO Domains are Red as they sit in queue. Initially these were all of the 29 FIBO Domains referenced in http://www.edmCouncil.org/downloads/20140310_FIBO_Development_Scenario.pdf. FIBO Domains become Pink as they each begin to be tested in various Use Cases by common tools described earlier.

© EDM Council, Inc. 2014. All rights reserved.
FIBO Domains become Yellow as they are validated both by machine and by SMEs. Finally, FIBO Domains become Green as they are approved as ratified standards. At any point in time, multiple FIBO Domains will be in this process, each following the Spiral Development concept outlined in section 4.4. Each is designated with a color to identify its state of completion.

FIBO Deployments can begin when a FIBO Domain becomes Pink, that is, when it is chosen from the FIBO BCO repository. As will be detailed below, Pink FIBO Domains are published by the EDMC as they are being built and tested. It is critical that Deploy and Maintain remain in sync with the official standards process as lessons are learned. An analogy to how this can work are the constant firmware updates of Internet routers each of which enters the market based on an as yet to be ratified, but already well-conceived standard.

This demands a highly developed process managed by state of the art collaboration tools within a complex infrastructure of both people and machines. The machine component of the infrastructure consists of GitHub, Jenkins and JIRA. The people component of the infrastructure consists of the FIBO Teams shown in the figure below. This includes:

- A FIBO Leadership Team (FLT)
- one or more FIBO Content Teams (FCT) operating simultaneously
- one or more Proof of Concept Teams (FPCT) operating simultaneously
- one or more OMG Task Forces (OMG/TF) operating simultaneously
- and implementation teams established by operators to put their own Operational FIBO into practice (FIT)
- Supporting these teams, are:
  - A FIBO Process Team (FPT) working to establish and maintain the FIBO infrastructure
  - FIBO Vendor Teams (FVT) providing FIBO test suites using their COTS

Each FVT will offer a platform on which FIBO test Use Cases may be executed through an automated test service described below (Jenkins) and may make a FIBO ‘sandbox’ available for their customers to exercise FIBO. FIBO Implementation Teams (FIT) are represented by the dark Green intersection of the
FLT, FCT and FPCT in Figure 2 that shows the relationships of each of these teams. The light Green in Figure 2 represents the use of the FIBO infrastructure and the testing process used by all teams.

3.2 FIBO Teams
This section elaborates on the process, and the machine and person components of the infrastructure. The interdependence of these teams cannot be over emphasized. While the color code is designed to show the distinct stages of the FIBO Domain, there is clear overlap between and among the teams that build, test, deploy and maintain them. As a practical matter, at any given point in time a person working on FIBO can expect to contribute his or her expertise to at least two FIBO teams, each of which has a distinct responsibility.

3.2.1 FIBO Leadership Team (FLT)
FLT members include ontologists, architects, subject matter experts and standards process specialists. The role of the FLT includes:

- Develop new and utilize existing standards, frameworks and guidance
- Build a strong FIBO roadmap and strategic framework
- Establish and maintain a FIBO methodology document
- Establish an accessible repository for UML and OWL for all stages of FIBO
- Governance and configuration management
- Manage the OMG process
  - Financial Domain Task Force (FDTF)
  - Request For Comment (RFC) documents
  - Finalization Task Force (FTF) issue resolution
  - Revision Task Force (RTF)
- Adopt and adapt current research and innovation
- Adjudicate changes in GitHub and JIRA
- Establish and mentor FIBO Content Teams (FCT)
  - Equip FCTs to support targeted industry partnerships in building out FIBO

The FLT meets in person at OMG quarterly technical meetings and from time to time at other venues as well as in weekly scheduled GoToMeeting sessions lead by the EDMC.

3.2.2 FIBO Content Teams (FCT)
FIBO Content Teams are formed when industry leadership decides that one of the FIBO Domains in the BCO repository, is ready to move from Red to Pink. Or, that a Domain should be rapidly developed so that it can be elevated to Pink and be prepared to support some operational need.

FCTs are composed of at least Five people mentored by at least one member of the FLT. The Team leader should be from within the FIBO Domain and include: an SME who understands the idea of ontology, an ontologist who understands the Finance Domain, a qualified super ontologist, or an SME from within the Domain. Broad FCT membership is encouraged from throughout the industry including at least two competitors to the Team leaders firm.
The role of an FCT includes:

- Develop Use Cases or User Stories, that can be satisfied by one or more of the 29 FIBO Domains:
  - These will be submitted to the FLT and published on the EDMC website
  - They can be used by the FCT that created them, or they can be used by FPCT teams (see below) to organize proofs of concept
- Form the FCT and prepare for Externality Reviews\
- Follow the FIBO Development Process to update the model:
  - Extract the Domain from the BCO in GitHub by looking at the UML visualization and the RDF/OWL models in the standards based tool of choice
  - Modify the existing model to suit the objectives of the use case
  - Review and validate pre-existing class hierarchies, properties, and restrictions editing and extending each as needed to address both the Use Cases and the integrity of the Domain
  - Carry out SME Reviews on the resulting model
    - For a Red FIBO with “Substantive” status, validate that the resulting model represents the same or enhanced semantics
    - For a Red FIBO with “Draft” status, complete, extend and review the business semantics of the model (in these cases, this happens alongside Externality Review):
      - Render restrictions describable as refinements and/or reuse of properties
      - Render existing classes and properties and disjoints and inverses
      - Edit definitions
      - Harvest additional synonyms
      - Review and edit provenance metadata ensuring that all references to other ontologies that are already in the OMG namespace are referenced appropriately.
      - Create spreadsheets or other business accessible formats to explicate terms, definitions, synonyms, and metadata, etc. reports for off line review
  - Provide Test Data for the Use Cases
- Subject the model to simple Pellet and Hermit tests that are run automatically as Jenkins jobs before committing to GitHub.
- Run the model through the FIBO Serializer and ensure that the model loads in Protege
- Adjudicate issues in EDMC JIRA
- Build and OMG required documentation
  - Request document numbers for, and formally submit, each of the following:
    - Written specification
    - Inventory file
    - OWL files, ODM XMI, UML XMI
    - Model File

---

6 Externality Reviews are intended to identify the required abstractions for a new FIBO and discover related ontologies with those concepts that may become part of FIBO or may be linked to FIBO.

7 Subject Matter Expert (SME) reviews are intended to capture and/or validate the business semantics of the FIBO ontologies. These are carried out using business facing (non-technical) diagrams, spreadsheets and (when available) controlled natural language presentation methods.
Signed submission Letter (with IP rights declaration)
- Submit all documentation to the OMG Financial Domain Task Force and Architecture Board
- Educate OMG issues in OMG JIRA

### 3.2.3 FIBO Proof of Concept Teams (FPCT)

FPCTs are established from time to time by EDMC members to use one or more FIBO Domains in a real world environment that can test the feasibility, or to prove the cost benefit of using FIBO in a particular use case for an operator or for a regulator. As with FCTs, the most important aspect of an FPCT is a leader with a Use Case or User Story.

These can be generated by the FPCT team itself, or taken from other proposed Use Cases or User Stories as published by the EDMC. An FPCT can form beginning with any stage FIBO. A Use Case or User Story may begin with a discussion that no FIBO content is supportive, for example. Then the FPCT actually would create a new Red FIBO. At the opposite end of this possibility, a Use Case or User Story in advance of starting a FIBO Implementation Team would begin with a Green FIBO.

PoCs will be carried out by a member organization to demonstrate how FIBO can be used in their business. While FPCTs are users of FIBO, not developers of FIBO, it is the intention and expectation that FPCTs use the FIBO Build, Test, Deploy and Maintain process and tooling as appropriate.

In contrast to the activities of the FLT during FIBO development, it is not necessarily expected that the data in a PoC will be unclassified and readable by the public or even by other EDMC members. The deliverables should include at least a lessons learned document, outlining strengths and shortcomings of FIBO, and in the latter case, recommendations for improvement, and issues that can be acted on by a content team or the FLT.

If appropriate, a demonstration or video of the PoC is a good deliverable.

### 3.2.4 OMG Task Force (OMG/TF)

The OMG process involves establishing a Finalization Task Force (FTF) when a draft set of documents on a particular new standard is approved by the OMG Architecture Board. It is the responsibility of the FTF to adjudicate issues, in OMG JIRA, that are surfaced during a public comment period. Issues that are not resolved by an OMG/FTF and new issues that arise over time, based on usage, may be resolved via the FTF or are transferred, once the specification is finalized, to a Revision Task Force (RTF). FIBO Domains are Yellow during this process. When the OMG AB is satisfied that the issues have been addressed, the standard is proposed for adoption by the broader OMG membership, at which point it becomes a formal specification. That is, the FIBO becomes Green. All revisions post-formalization are handled by the RTF.

### 3.2.5 FIBO Implementation Teams (FIT)

FITs are represented in the Green of Figure 1. FITs are created by the operators with the sole desire to solve particular business problems. FITs extend FIBO BCOs into FIBO Operational Ontologies (FOO). These FOOS are executable designs that solve a particular business problem of the operator.

A FIT will be staffed mainly from a member organization, but the FLT may be asked to provide mentoring advice to the FIT at all stages of development, from use case identification to identifying an appropriate
FIBO to use, to the development of an appropriate FOO. Lessons learned by supporting FITs in this way will be fed back into the core FIBO ontologies and documentation.

3.2.6 FIBO Process Team (FPT)
The FIBO Process Team is responsible for establishing and maintaining the FIBO Infrastructure. This includes management of web servers for FIBO Namespaces, GitHub/Jenkins management and how they are referenced in EDMC JIRA. In particular, the FPT is responsible for liaison with all vendors, helping them to set up Jenkins slaves for FIBO testing. While the FIBO Content Teams are primarily responsible for the development of the tests themselves, the FPT is responsible for the mechanism of automatic test jobs, making sure that these tests run correctly on the Jenkins servers, that EDMC JIRA manages the FIBO content process and that OMG JIRA manages the OMG documentation process.

The FIBO process team includes members from each of the other FIBO teams who represent both user requirements and particular technical expertise in one or more of the infrastructure platforms/systems. The FPT is in constant communication with OMG and EDMC webmasters to ensure synchronization of all applicable websites.

3.2.7 FIBO Vendor Team (FVT)
Vendors represent a broad user community, for themselves and for their clients, through their expertise in platforms, technological insight, and research that intersects with domain depth applied to real-world business problems.

To participate in the FIBO Continuous Integration and Improvement lifecycle, as a designated FIBO Vendor Team, a vendor must provide a Jenkins slave platform for testing FIBO use cases as part of the EDMC Council’s Jenkins-JIRA infrastructure so that the Council’s Jenkins server can schedule generic and vendor-specific test jobs directly to vendor products on vendor hardware.

Vendors are also urged to make a FIBO ‘sandbox’ available for their customers to exercise FIBO.

FIBO Vendor Teams are expected to lend their expertise to, and collaborate with, FIBO Content Teams and FIBO Proof of Concept Teams deploying FIBO models to their own platform and tool environments. The FLT is happy to work with any vendor who is willing to provide resources to help determine how that vendor can best contribute to the FIBO community and infrastructure.

Contributions that the vendor provides to FIBO include:

- Testing in real-world scenarios: operational use and test cases outside and beyond those provided by other teams; test suites, structural tests, performance and scalability stress testing; issues, resolutions, and gaps in domain coverage
- Tooling for modeling, visualization, application development and deployment, integration and interoperability
- Data for testing in various business and regulatory scenarios
- Deployment to their own and others’ platforms in operational environments, on end-to-end platform ecosystems that may include SaaS, PaaS, and cloud hosting in general
• Compatibility with and conversions between UML, OWL and other modeling paradigms, and the application of reasoning and rules
• FIBO applications and extensions, domain and ontology research and engineering, implementation and deployment stories, possibilities for factoring, and the application of reasoning, rules, and optimization in real-world scenarios
• Free or nominally priced versions of platform, tools, and related products
• Socialization of FIBO in and among the end-user community
• And, documentation of the above, as appropriate, within the FIBO Build, Test, Deploy, and Maintain infrastructure

4.0 FIBO Process

The simplified EDMC-OMG process shown in Figure 1 and detailed in appendix A, evolves each FIBO Domain through the build and test phase. It does not show deploy and maintain that can begin at the discretion of any of the teams by extracting a FIBO model from the EDMC or OMG repositories.

GitHub is the primary repository for all FIBO content including the RDF/OWL files, documentation, use cases, test data, schedules, and more. GitHub manages FIBO which resides on a variety of virtual servers under EDMC name spaces herein referred to as the EDMC Website. At the discretion of the FLT, some of these namespaces are accessible in the wild for Build and Test and under restricted access for Deploy and Maintain. See https://GitHub.com for details.

Jenkins is the primary FIBO Continuous Integration (CI) platform. Jenkins allows automation of the testing process by executing standard test scripts. See http://jenkins-ci.org for details.

An EDMC JIRA instance is the mechanism for managing FIBO content build and test. An OMG customization of JIRA is the primary FIBO issue tracking mechanism for OMG issues. See https://www.atlassian.com/software/jira?_mid=17ac4851a5f8095301ccb8787bf6f472&gclid=CJWuwf7I7b8CFUVo7Aod_hiAgA and http://solitaire.omg.org/browse/FIBOFTF#selectedTab=org.omg.jira.task-forces%3Ataskforce-issues-panel for details.

This section discusses the FIBO Build, Test, Deployment and Maintenance Environment based on GitHub, Jenkins and JIRA being its core components. A significant challenge is keeping these necessary tools and the data contained within them in synchronization. This is the primary responsibility of the FIBO Leadership Team, but involves all of the other Teams using any FIBO. Figure 3 shows the relationships between and among these systems as FIBO progresses from Red to Green.
The colors in this Figure match the purposeful colors in Figures 1 and 2. The sequence represented by the numbers matches the sequence of what actually happens as a FIBO moves through the steps of Figure 2 and [http://www.edmCouncil.org/downloads/20140310_FIBO_Development_Scenario.pdf](http://www.edmCouncil.org/downloads/20140310_FIBO_Development_Scenario.pdf).

1) A FIBO Content Team is formed as described above, and selects a Red FIBO from the queue of FIBO Domains that are in the GitHub Repository. This FIBO becomes Pink. The FIBO is modified as necessary according to the process of Section 3.2.2. Issues may be discovered and are recorded in EDMC JIRA for management. Some issues are corrected manually. Use case data, developed by the FCT, is used to test the FIBO through the automation provided by Jenkins. This Pink loop continues until all issues are resolved.

2) Jenkins testing returns no errors meaning that test scenarios provided by the FCT are positively concluded – EDMC JIRA has no unresolved FIBO content issues. The FIBO becomes Yellow. That is, it is ready to be documented according to OMG rules and to enter the OMG Standards process.

3) This Yellow FIBO perhaps has additional issues discovered during the public OMG Request For Comment (RFC) period. These issues are documented in OMG JIRA. FIBO content issues discovered are referenced to EDMC JIRA where they are adjudicated by EDMC GitHub/Jenkins. An OMG customized JIRA tool within an OMG Finalization Task Force (FTF) chartered by the OMG Architecture Board adjudicates OMG documentation issues. This FIBO stays Yellow until
all JIRA issues are resolved. These Yellow FIBO Domains are published on the OMG website as draft standards.

4) When all issues of all types are resolved, the Yellow FIBO is elevated to Green as an OMG standard by the OMG Architecture Board and published on the OMG website as a ratified standard.

Figure 4 further details the content and use of the FIBO repository in all of its facets by the various teams exercising their responsibility using a variety of tools as described below in the figure.

1) The FIBO Leadership Team (FLT) is responsible for all FIBO content, process and Infrastructure. All FIBO Domains in all stages are stored in the EDMC website and references from GitHub.

2) FIBO Content Teams (FCT) extract the FIBO Domain useful to them as published in OWL, UML and various supporting documentation. FCTs and others supporting FIBO can create OWL files in any way they like, but they will store them in GitHub in RDF/XML as generated by Protégé. Any non-Protégé “Supporting OWL” files may also be stored in GitHub, but will not be published to the EDMC website, nor submitted to the OMG for ratification. Creation of diagrams for business users and data modelers can be done using any tool that provides useful output. These outputs must be generated from the OWL files.
3) Testing will be done with OWL and RDF-based tools as automated GitHub/Jenkins jobs managed by JIRA and supported by the FIBO Vendor Team (FVT).

4) When this has iterated to the satisfaction of an FCT, documentation is prepared and goes to the OMG Financial Domain Task force (FDTF), the Architecture Board (AB) and a Finalization Task Force (FTF). In Figure 4, all of this is referred to as OMG/TF 4.1

4.1 GitHub

The open-source collaboration software Git, and the open cloud-based platform GitHub, have emerged as the de-facto standard for open-source collaboration. GitHub provides an open forum for reviewing and contributing to shared projects, and is very well-suited to the needs of FIBO. This section elaborates on the specifics of the use of GitHub and other infrastructure components for FIBO.

GitHub confers many advantages for a collaborative team. GitHub uses “branches” and “forks” to keep track of proposed and accepted changes in the form of “Pull Requests.” Team members can create a fork, modify multiple files, and “commit” them back into their own local “clone” of their “fork.” They can then “push” all their “commits” to their fork on GitHub. Any other collaborator can create a clone of that fork as well, or create his own fork and merge their own fork with the original fork at GitHub, run tests, make comments or propose changes, and create a “pull request” for the “root” fibo repository. The “owners” of that root repository (the FLT) can then accept or reject that pull request. The process is much less error prone than other means of collaboration such as email, thumb drives and unstructured cloud storage because it keeps a running track of all changes. It is a proven method for large, diverse teams to coordinate their work. It will work for non FIBO chartered teams, when the fibo repository goes public, when anyone can create a fork and make changes in that fork and propose changes to the root-repo via Pull Requests.

FIBO observes the following policies with regard to the use of GitHub. These policies are typical of collaborative projects on GitHub, and are sensitive to many technical considerations of web publication:

1. FIBO Domains are stored in separate directories in a single Git repository called fibo. These directories are named for each of the FIBO Domains, e.g., fnd, be, ind, etc. These directories are siblings in the FIBO repository.

2. Each Domain directory contains a number of directories corresponding to modules of that Domain. The names of these directories are lower-case, with multiple words separated by hyphens and no embedded spaces, e.g., ../securities/asset-backed/mortgage-backed-securities.

3. Each collaborator creates a fork of the fibo repository. A contribution to FIBO is delivered by means of a pull request to the fibo repository. The FLT must accept or reject a pull request in its entirety; therefore requests that entail multiple atomic changes are less likely to be accepted.

The use of GitHub is closely coupled with policies regarding release management in FIBO. FIBO follows the strategy known as “Semantic Versioning” as outlined at http://semver.org/. Namely, a version number has three components called Major, Minor and Patch (or Sequence).
4. FIBO releases are indicated in the master branch of the fibo repository with _tags_ that reflect the version number. The tag name begins with the letter “v” (for “version”), followed by the major, minor and sequence numbers separated by underscores, e.g., “v1_2_1.”

The use of GitHub is also closely related to naming conventions concerning the IRIs of resources in the FIBO ontologies. The following naming conventions are followed in the FIBO development:

5. While a Domain is in development (Red, Pink or Yellow as defined above), all IRIs will begin with _http://spec.edmcouncil.org/_/. When a Domain graduates from Yellow to Green (as part of the OMG process), the namespaces are migrated to begin with _http://www.omg.org/_/. Other conversions (e.g., from lower-case hyphenated names to OMG standard camel-case) will also be performed at this time (eventually by an automated process). The local part (the final segment after the last “/”) of resource IRIs will not change, and conform to the usual conventions of the Semantic Web (UpperCamelCase for Classes and Individuals, lowerCamelCase for properties).

6. Each ontology is indicated by an IRI (the so called ontologyIRI) that reflects only the major version, using the following scheme:

   http://spec.edmcouncil.org/<family>/<major-version>/<domain>/<module..n>/<ontology>

   where:

   - `<family>` is always fibo
   - `<major-version>` is just an integer number prefixed with the lower case letter v representing the major version of the `<family>`
   - `<domain>` refers to one of the 29 FIBO domains in lowercase, and any number of modules (typically just one) inside that domain

Then next to the ontologyIRI, each ontology can also have a versionIRI, which includes the full version number with all three components (major, minor, patch) or can simply contain a branch name rather than a “spade in the ground”-tag:

http://spec.edmcouncil.org/<family>/<version|tag|branch>/<domain>/<module..n>/<ontology>

where:

- `<version>` or `<tag>` (which is basically the same here) are in the format `<major version>_<minor version>_<sequence number>`
- `<branch>` can be “master”, “yellow”, “red”, “pink” or any other name as long as it does not look like the above tag name

The relationship between owl:ontologyIRI and owl:versionIRI is outlined in the OWL 2.0 Recommendation.

Example:

- owl:ontologyIRI: _http://spec.edmcouncil.org/fibo/v1/fnd/domain/etc_
- owl:versionIRI: _http://spec.edmcouncil.org/fibo/v1_2_3/fnd/domain/etc_
7. The namespaces used for the definition of all resources (classes, properties, etc.) will reflect the major version of the ontology in which that resource is defined, e.g.,

<http://spec.edmcouncil.org/fibo/v1/fnd/law/jurisdiction/LegalSystem>

These conventions are reflected in GitHub in the following ways:

8. The structure of the directories from the domain level down (e.g., fnd, be, etc.) matches the structure of the base URIs of each FIBO ontology. In the example in point 7 above, the folder structure from fnd down is fnd/law/jurisdiction.

These conventions allow easy reference to the ontologies during development using the existing infrastructure of GitHub. For example, the file jurisdiction.owl in the example above is published by GitHub at the URL

https://raw.githubusercontentusercontent/edmcouncil/fibo/v1_3_21/fnd/law/jurisdiction.owl

where “v1_3_21” is the name of the GitHub tag corresponding to the release.

This corresponds to the owl:versionIRI

<http://spec.edmcouncil.org/fibo/v1_3_21/fnd/law/jurisdiction>

The web server at spec.edmcouncil.org performs this translation automatically, so that the FIBO ontologies conform to Linked Data publication best practices, allowing them to be accessed easily using commonly available semantic web tools.

4.2 Jenkins

Jenkins is an open source product that can be installed in a master/slave configuration. The Jenkins master server has a web-based interface that can be used to define and monitor “jobs.” Jobs can be activated manually, or by many different types of “triggers.” One important trigger is a “push” into the GitHub repository that could trigger the execution of many different jobs on the Jenkins server. Another important trigger is the successful or unsuccessful execution of another job that allows for “pipelines” of jobs that can be run in sequence or in parallel.

One Jenkins master server can manage any number of slave servers. A slave Jenkins server has no web based interface but is basically a job-execution “container” that can be automatically and remotely installed by the Jenkins server on any target host (provided that the Jenkins master has SSH access to the target host). All jobs are defined on the master, but can be “tagged” to only run on specific slaves or on any available slave based on various metrics. In that sense, the Jenkins server is just a job scheduler.

4.3 JIRA

When there is a commit to GitHub, the commit message starts with the JIRA issue number as follows: “<JIRA issue number> <commit message>”. Any GitHub repository, including each fork of it, can thus be connected to a central JIRA server so that each GitHub commit ends up in the Activity log of the
appropriate JIRA-issue. Also, JIRA and Jenkins can be connected to each other as well, so that each unsuccessful run of a Jenkins job ends up as a comment in the Activity log of all the JIRA-issues that were involved in the scheduling of the job. For example, if a pull request involving 10 JIRA issues is accepted, and various Jenkins jobs start running because of the fact that the pull request is accepted, then all the involved JIRA issues will get affected by a new comment generated by Jenkins, with a direct link to the failed job.

Figure 5 is a simplified picture of how GitHub/Jenkins and EDMC JIRA work together for testing, membership review and reviewer feedback. The EDM Council owns and manages the GitHub account, the Jenkins master and EDMC JIRA. Vendors can “plug in” their Jenkins slave server to run as many of the test jobs defined at the master as possible. Each vendor can install their own products on their own slave server, such as an RDF database, where jobs can access that product only when they run on that particular slave. This allows generic and vendor specific jobs to run side by side so that each vendor can opt in to run as many of the FIBO jobs as possible, showing how their product performs and works in the most optimal way. This is true for database vendors and vendors that can generate documentation sites or convert files from OWL to UML or vice versa.

4.4 Use Cases
Critical to the success of FIBO is that it satisfies the business needs of the Finance Industry. These business needs will range from proving regulatory compliance to design or redesign of business processes and IT systems. Understanding and documenting the needs of the user community and how a proposed solution will satisfy those needs has been the bane of engineers in every endeavor of
mankind. Diagrams, and other types of flow charts gave way to a structured approach known as “Use Cases” in the 1980s. More recently, usually associated with Agile Software Development, Use Cases may instead be “User Stories.” A common theme of both Use Cases and User Stories is that both must contain testing criteria to determine if the user need is satisfied.

An acceptable Use Case or User Story provides the baseline set of requirements for that Domain/Sub-Domain to graduate from the Red FIBO stage to the Pink FIBO stage.

For either a User Story or a Use Case to be accepted by the FLT, it must include scripts in SPARQL that perform a viable test on some portion of the emerging model. This is required before an FCT can be commissioned, and prior to an FPoC. The Use Cases and User Stories will provide the basis for determining whether or not a given concept is needed, whether some set of relationships (properties) are needed, and “when to stop.” The FLT will determine whether or not the Use Case or User Story is sufficient to commission an FCT. The choice of presenting a Use Case or a User Story depends on the scope and nature of the business need and the experience of the team.

Use Cases and User Stories can be specific or comprehensive. Use Cases use stick figure diagrams to show relationships. User Stories use Story Maps. Reference contains details on common approaches to for each practice.

An example of a specific Use Case follows:

“As a potential shareholder, I want to see which party controls organization X, so that I can assess who appoints the board of directors.” This is often called a Competency Question.

A sample use case diagram for this simple use case above might look like this:

---

10 “User Stories”. Agile Alliance.
An example of a more comprehensive Use Cases would be:

“As part of our AML efforts, I need to know which party controls an organization, any entities owned by that organization, and who has a beneficiary relationship to those (sub-)organizations, so that I can determine if certain common money laundering patterns might exist in this organization.”

Specific Use Cases and User Stories can be used as drivers for unit tests (see below) during the Modeling Phase, while more comprehensive Use Cases can drive ontology development, e.g., by providing competency questions or driving commonality and variability analysis during the Validating Phase.

FIBO encourages the collection of Use Cases, especially comprehensive Use Cases, from interested parties in the industry. Contributions of Use Cases require little commitment, but are nevertheless an important and effective way to inform the direction of the FIBO effort.

4.5 Testing
FIBO will adhere to “Behavior Driven Development” (BDD)/“Test Driven Development” (TDD) “Continuous Integration” and “Continuous Improvement” principles applied throughout the FIBO Build, Test, Deploy Maintain process as part of the Continuous Integration Process as previously introduced. See\(^{12}\) and\(^{13}\) for additional background on these concepts.

\(^{12}\) http://www.slideshare.net/cmungall/ontologies-andcontinuousintegration

\(^{13}\) http://martinfowler.com/articles/continuousIntegration.html
An example of this spiral methodology is shown in the diagram below:

The FIBO Continuous Integration Cycle includes FIBO model building, testing and validating through integration testing, deploying PoCs to further improve the FIBO modules and ready them for OMG approval, and successfully passing OMG approval, then maintaining the growing community of FIBO modules staged for public use. The testing overlay at each of these stages includes:

4.5.1 Modeling and Unit Testing - Performing unit tests on the FIBO ontologies is critical. Unit testing is a method by which the quality of ontologies can be maintained during the development process. The sources of unit tests are the User Stories and the Use Cases, as previously stated. The test data, the query, and the desired result can all be stored as simple text files, so that conformance to the test can be easily verified.

The Jenkins framework allows for automation of this process. Unit tests will be developed on the submitters workstation, verified in Protégé and when pushed into a repository, Jenkins will run all relevant tests. Any failure to match the desired output will be reported back to the user who pushed that change.

Unit Test of the model assures that the model structure is logically sound, and readies it for the subsequent testing of model substance in the Integration Test process.

4.5.2 Validating Through Integration Testing - The substance validation portion of the cycle includes creating instances/individuals of the portions just modeled, generating SQL/SPARQL statements that implement the simple use cases that apply to the recently modeled portion(s), and running those as
initial proofs that the content of the model meets desired results. The new portions of the emergent models are the combined with the rest of the model set, and are run with all SPARQL tests, as an ‘integration’ test, much like system regression tests. The model (or model portions) may then be run through advanced ontology testing tools, such as OOPS! and OQuaRE\(^\text{14}\), as a final validity cycle.

### 4.5.3 Documenting

By this stage, the model is basically in tested state for this turn of the cycle, and the Use Cases and testing results may be used as part of the FIBO Documentation.

### 4.5.4 Deploying

The testing results are then made part of the OMG submission. While responsibility for the assurance of complete testing logically falls on the FIBO Content and the Proof of Concept Teams, any of the other teams which touch the FIBO Model will be responsible for Unit Testing, and Use Case generation of their portions of the emerging model; and they may be called upon by the FLT for further testing activities.

### 5.0 Roles

There are several roles to play or “privileges” to have within the FIBO Develop, Build, Test, Deploy, Maintain Ecosystem. These roles are largely defined by the FIBO Team structure.

<table>
<thead>
<tr>
<th>Role</th>
<th>GitHub Admin</th>
<th>GitHub Contributor</th>
<th>GitHub Viewer</th>
<th>Jenkins Admin</th>
<th>Jenkins Job Admin</th>
<th>Jenkins Viewer</th>
<th>JIRA User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leadership Team (FLT)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Content Teams (FCT)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PoC Team(s) (FPCT)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Vendor Team(s) (FVT)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Process Team (FPT)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMG Task Force(s) (OMG/TF)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Industry Implementation Teams (FIT)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material of those of the author(s) and do not necessarily reflect the views of members of the Enterprise Data Management Council, or any of the organizations affiliated with the FIBO Leadership Team participants.

Acknowledgements

This report is based upon work supported by the Enterprise Data Management Council (EDMC) in cooperation with the following Individuals as members of the FIBO Leadership Team and the FIBO Foundations Ontology Finalization Task Force (FTF) of the OMG.

Michael Bennett, EDM Council
Pete Rivett, Adaptive Inc.
Elisa Kendall, Thematix Partners, LLC
David Newman, Wells Fargo
Dean Allemang, Working Ontologist, LLC
Richard Beatch, Bloomberg L. P.
Dennis Pierson, Mphasis
Jacobus Geluk, Bloomberg L. P.
Bobbin Teegarden, OntoAge
Michael Atkin, EDM Council
Dennis E. Wisnosky, Wizdom Systems, Inc.
There are Four FIBO Types

Red FIBOs are in Queue. They are the Business Conceptual Ontologies that have been maturing since 2008. They are resident on the EDMC website as UML files and in GitHub as both UML files and RDF/XML. Pink FIBOs have been selected by a FIBO Content Team (FCT) formed to satisfy a particular industry need. They begin a formal validation/Testing process using open source and other tools managed by GitHub for version control, Jenkins for testing and JIRA for issue resolution.

Yellow FIBOs have entered the OMG Process. They comply with the OMG rules for public comment and Green FIBOs have exited the OMG process as ratified standards. They are ready to become Operational Ontologies.