IOF ARCHITECTURE TASK GROUP

IOF ONTOLOGY DEVELOPMENT LIFECYCLE AND PUBLICATION PROCESS

EVAN WALLACE
NIST

2024-02-08

REVIEW OF PROPOSED IOF ONTOLOGY DEVELOPMENT PROCESS

Agenda

- Genesis
- Design principles and choices
- Tour of the process
- Summary and next steps

ONTOLOGY DEVELOPMENT INFRASTRUCTURE AND PROCESS

- Motivation:

  Quickly transform IOF from ad-hoc community to a standards consortia

  Enable developing, curating, and managing ontologies as standards

- How?
  - NIST funded a project to provide infrastructure and rules enabling development and governance

DESIGN PRINCIPLES AND CHOICES

- Use model driven standards development approach

  treating OWL ontologies as machine processible “models” from which documentation and browser views can be generated.

- Use a software/systems engineering style workflow to drive and scope ontologies developed by IOF.

  Use cases provide both requirements and serve as supporting materials for understanding the ontologies.

- Use software tools and techniques to track and manage joint development and capture rationale.

ONTOLOGY DEVELOPMENT PROCESS

- Tools – technology support for ontology development
- Actors – responsibilities
- Process – in five acts
- **Goal**: *Continuous Improvement*

TOOLS

- JIRA for project management
- Protégé for editing
- Serializer to maintain canonical form for change tracking and comparison
- GitHub for shared access and version management and tracking
- Ontology Viewer for quick exploration of branches and releases

ACTORS

- Roles for IOF Ontology Development
  - Domain or Subject Matter Expert
  - Ontologist
  - Working Group Chair
  - Business Architect
  - Release Manager
- Boards
  - Ontology Review Board
  - Technical Oversight Board

ACT 1: TECHNICAL OVERSIGHT BOARD (TOB) AND EPICS

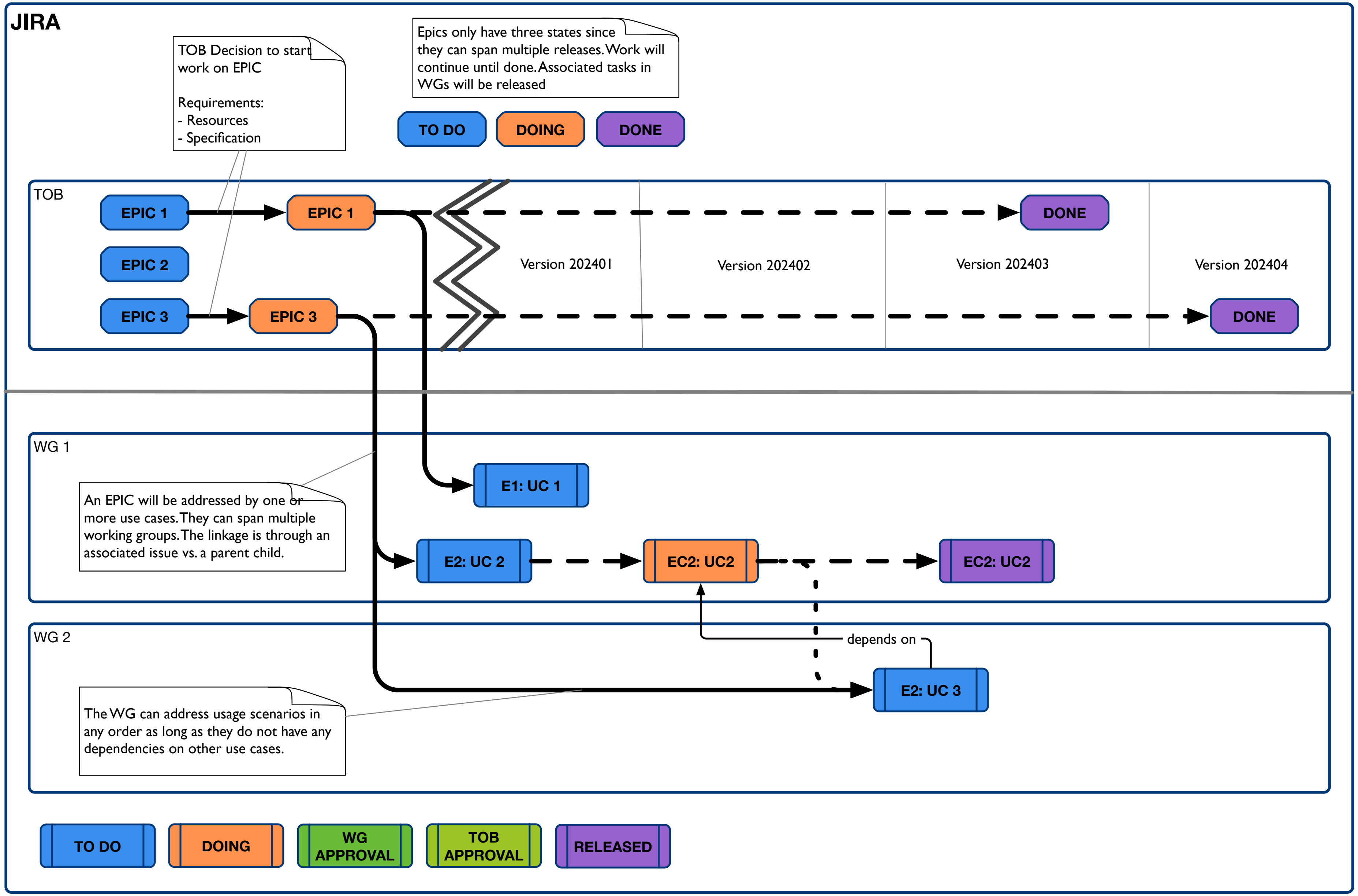
- IOF Strategy
  - Working Groups, Board Members, and Chairs submit requests to the TOB as Epics
    - An *Epic* (taken from agile) is a significant development objective that can span multiple releases and working groups
  - Document the high-level problem statement and set of business objectives addressed by the ontology that provide value
  - TOB, made up of all the WG chairs, decides the order and priority of the *Epics* and where aspects of the Epic will be handled
    - For example: Allen's Temporal Algebra is a general cross-cutting concern addressed in Core
    - Aspects of Temporal relations are required by Supply Chain, Maintenance, etc. for scheduling and delivery…
  - **Resources must be available to proceed to Act 2**

NECESSARY ROLES FOR ACT 2

- **Domain or Subject Matter Expert (SME)**
  - May know nothing about ontologies
  - Knows business needs
- **Business Architect**
  - Translates SME business needs into scenarios and competency questions
- Working Group Members
  - Provide guidance and insight from different perspectives

ACT 2.1: FROM EPICS TO SCENARIOS

- Create Usage Scenarios/Cases
  - For a given Epic
    - **The domain experts and the business architects work together to create use cases**
    - Identify stakeholders in representative organizations that benefit from this ontology
    - Define the subset of the Epic to be addressed in this iteration (scope)
    - Draft scenarios by providing the following:
      - Provide the role of the personnel from whose viewpoint the scenario is given
      - Describe the context and the situation that needs to be addressed
      - Describe the steps to achieve an outcome
      - And provide business value to the organization given a successful implementation
    - **Estimate "Cost" and "Complexity" to address each scenario**



JIRA

TOB Decision to start work on EPIC

Requirements:
- Resources
- Specification

Epics only have three states since they can span multiple releases. Work will continue until done. Associated tasks in WGs will be released

TO DO
DOING
DONE

TOB

EPIC 1
EPIC 1
DONE

EPIC 2

EPIC 3
EPIC 3
DONE

Version 202401
Version 202402
Version 202403
Version 202404

WG 1

An EPIC will be addressed by one or more use cases. They can span multiple working groups. The linkage is through an associated issue vs. a parent child.

E1: UC 1
E2: UC 2
EC2: UC2
EC2: UC2

depends on

WG 2

The WG can address usage scenarios in any order as long as they do not have any dependencies on other use cases.

E2: UC 3

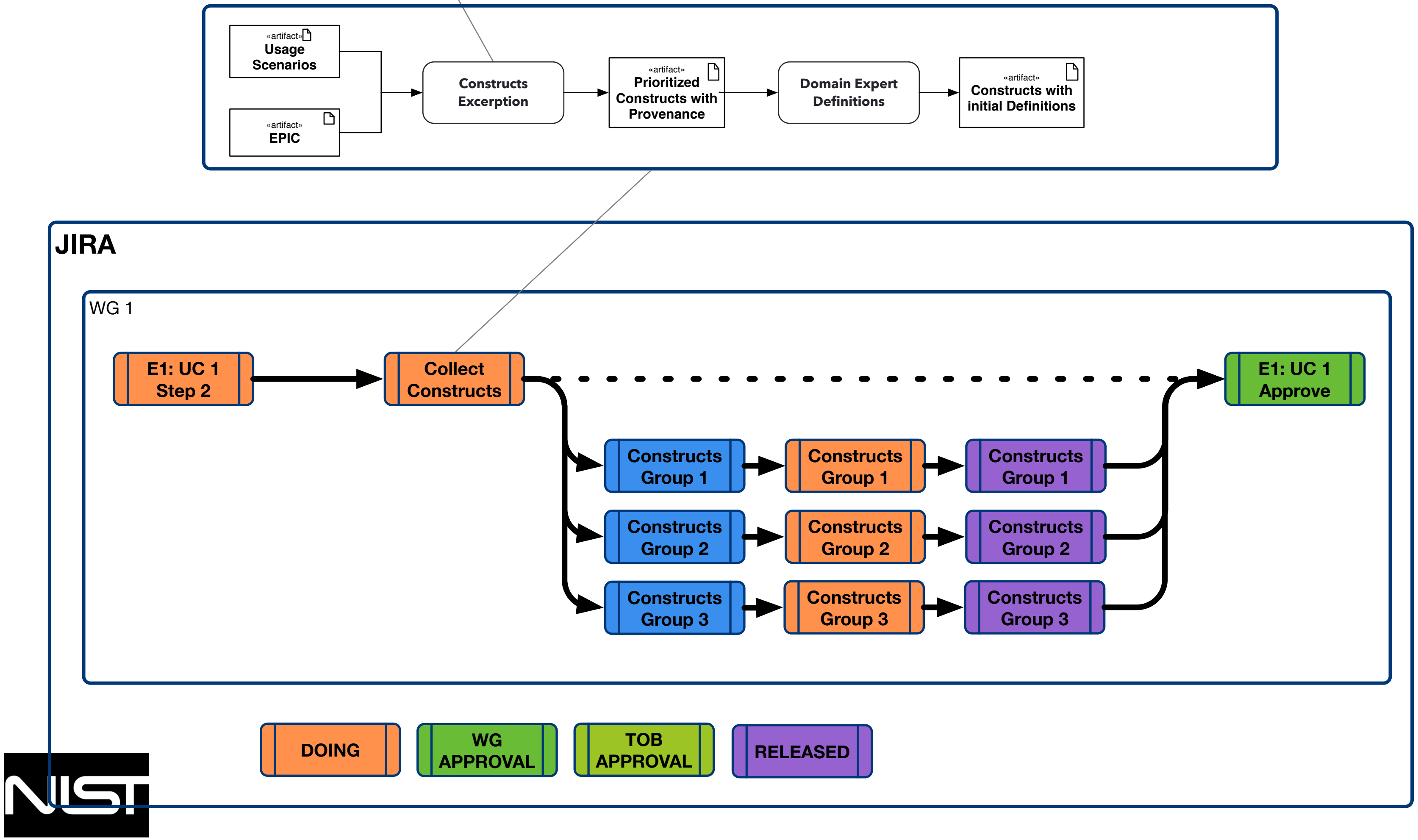
TO DO
DOING
WG APPROVAL
TOB APPROVAL
RELEASED

ACT 2.2: FROM SCENARIOS TO COMPETENCY QUESTIONS

- **For each scenario, craft multiple competency questions to achieve business outcomes**
- **Sample data is required for each competency question**
  - Data can be simulated, but in every case, data provenance must be documented
- **A set of expected results from the competency questions**
- The competency questions are to be cast as SPARQL queries and run using automated CI software
- Example:
  - For manufacturing process X requiring capabilities C to produce part Y, what assets have availability and sufficient capabilities?
  - What tools T and replacement parts C are required to repair pump P with leaky casing?
  - What are the aggregate capabilities of shop S at time T given equipment E and resources R?

COMPETENCY QUESTIONS

Extraction of keywords and key phrases from the vocabularies, glossaries, policies, procedures, process, and business architecture artifacts, standards, best practices, and other documentation available to create a preliminary term list, with preliminary definitions and other annotations. Note that natural language processing tools can extract key terms from a corpus of documents. Terms are also solicited from domain experts



«artifact» Usage Scenarios

«artifact» EPIC

Constructs Excerption

«artifact» Prioritized Constructs with Provenance

Domain Expert Definitions

«artifact» Constructs with initial Definitions

JIRA

WG 1

E1: UC 1 Step 2

Collect Constructs

Constructs Group 1

Constructs Group 2

Constructs Group 3

Constructs Group 1

Constructs Group 2

Constructs Group 3

Constructs Group 1

Constructs Group 2

Constructs Group 3

E1: UC 1 Approve

DOING

WG APPROVAL

TOB APPROVAL

RELEASED

ACT 3.1: VOCABULARY DEVELOPMENT

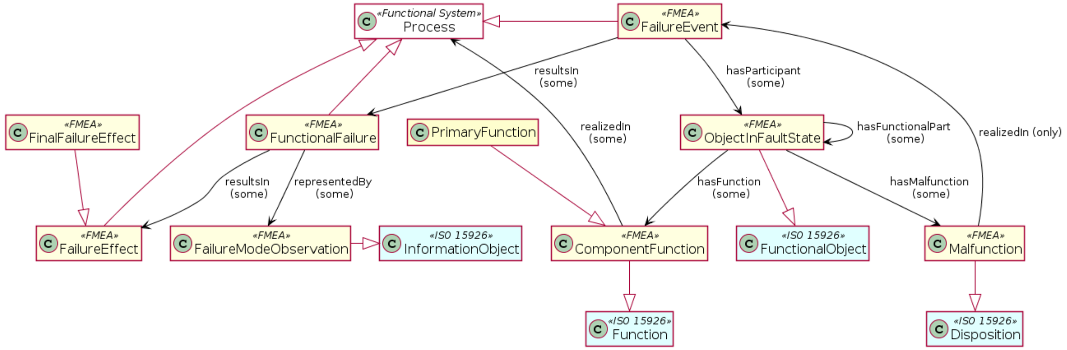
- The domain experts are necessary for vocabulary development
- Using the scenarios, competency questions, and data sets, find key phrases and terms from the following places
  - Industrial and business vocabularies
  - Glossaries
  - Policies, Processes, Regulations, and Procedures
  - Standards and Best Practices
  - Domain experts
- Provide a brief definition for each term and keep track of the sources of each term and phrase

ACT 3.2: CURATION AND NORMALIZATION

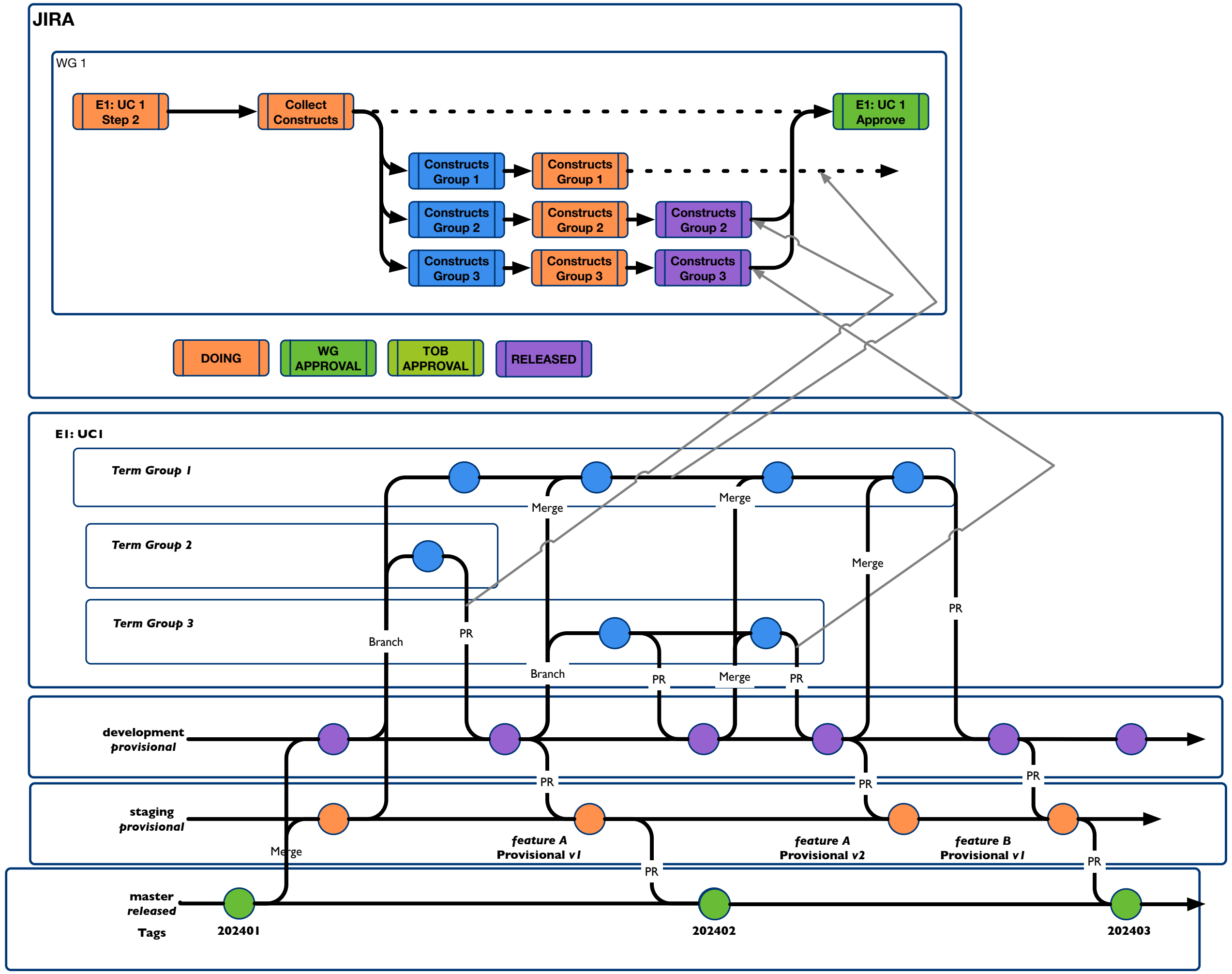
- Determine which terms are critical to support the competency questions and definitions
- Find what terms are required to support the definitions of the critical terms
  - Make sure to include relationships that will become properties
  - Set aside unused terms for later
- Further refine definitions using ISO 704 to create natural language definitions for each term and relations to the other terms
- Create a semi-formal natural definition (without rigor) to help with axiomatization
- Send the content to the ontologist(s) to begin ontological modeling

ACT 4.1: ONTOLOGY DEVELOPMENT

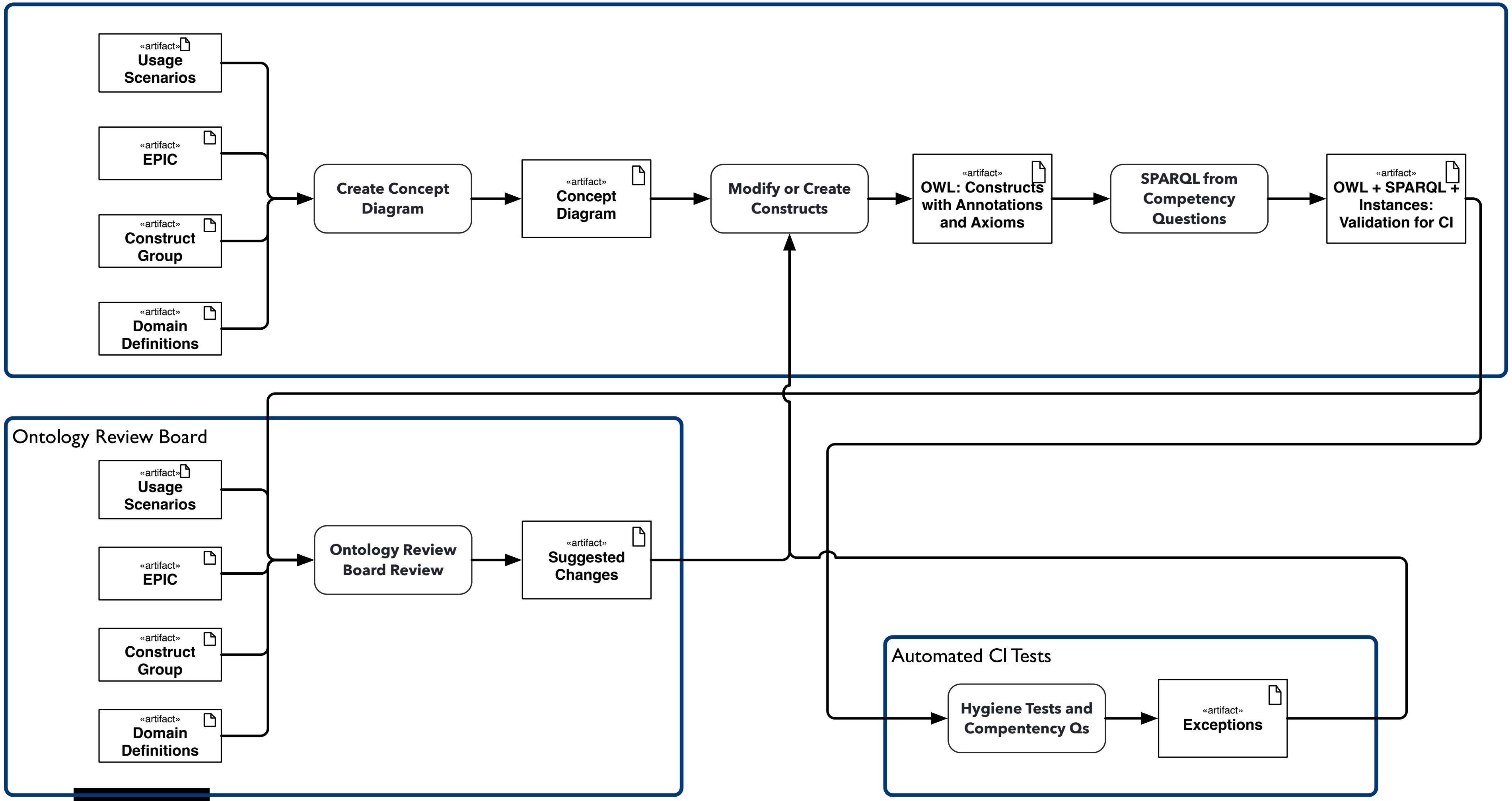
- Take the results of the vocabulary development phase and begin creating or refining Constructs
  - A construct is an OWL Class or Object Property
  - Separate constructs into two categories: Doman and Cross Domain (Core) terms
  - Create a diagram of the ontology to help visualize the relationships (e.g. PlantUML)
  - Create OWL and annotations for first-order-logic axioms and definitions, semi-formal NL
- Request Review from Onto Review Board
- Create SPARQL for competency questions
- Iterate with the working group Onto Review Board



BRANCH MANAGEMENT



DEVELOPMENT PROCESS



«artifact» Usage Scenarios
«artifact» EPIC
«artifact» Construct Group
«artifact» Domain Definitions
Create Concept Diagram
«artifact» Concept Diagram
Modify or Create Constructs
«artifact» OWL: Constructs with Annotations and Axioms
SPARQL from Competency Questions
«artifact» OWL + SPARQL + Instances: Validation for CI
Ontology Review Board
«artifact» Usage Scenarios
«artifact» EPIC
«artifact» Construct Group
«artifact» Domain Definitions
Ontology Review Board Review
«artifact» Suggested Changes
Automated CI Tests
Hygiene Tests and Compentency Qs
«artifact» Exceptions

4.2: ONTOLOGY DEVELOPMENT MANAGEMENT

- Ontologies are managed on GitHub and branches and commits are associated with Jira issues
- **Ontologies must be committed using the "serializer" to rewrite the ontologies in a canonical form so Git can diff them properly and changes can be reviewed**
- Tool integration allows SDO work in Jira to track changes made in GitHub repositories
- Branches are merged to master once TOB votes to accept the content into the latest version
- Intermediary merge branches are used to support collaborative development and staging releases

Arizona State University

ACT 5.1: APPROVE AND RELEASE

- Once the changes are complete
  - WG Approval as a vote to send to TOB
  - TOB Approval to include in the next release
  - If approved, create a pull request to merge into the main branch, starting the CI test processes
- All content MUST be verified by automated testing in CI process
  - Pass all hygiene, competency, and rule tests
    - Hygiene tests validate the existence of necessary annotations, annotation structure, and content
    - Runs Reasoner for logical consistency
    - Evaluates all SPARQL queries to verify competency questions and rule tests
  - All rules, SWRL/SHACL, must include tests, and all validations must pass

ACT 5.2: RELEASE PROCESS

- Releases occur when content is ready for publication
  - Content can be released at different levels of maturity
  - Two categories: *Provisional* or *Released*
- Iterations for refinement of content to improve normative releases
- We intend to move towards time-boxed releases semi-annually or quarterly
- Continuous integration and deployment architecture allows for beta use of verified content before official tagged release
- Regression testing a must
- A release occurs when the TOB votes to tag a set of content as a release
  - Releases are `YYYYNN` where `NN` is a monotonically increasing release number resetting to `01` every year: `202302` is the second release in 2023

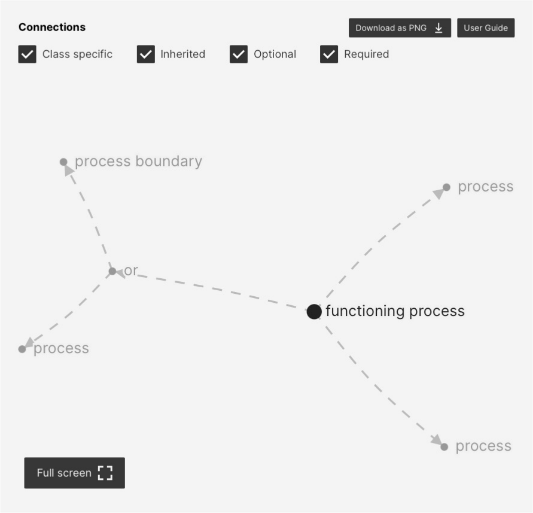
DOCUMENT AND VISUALIZE

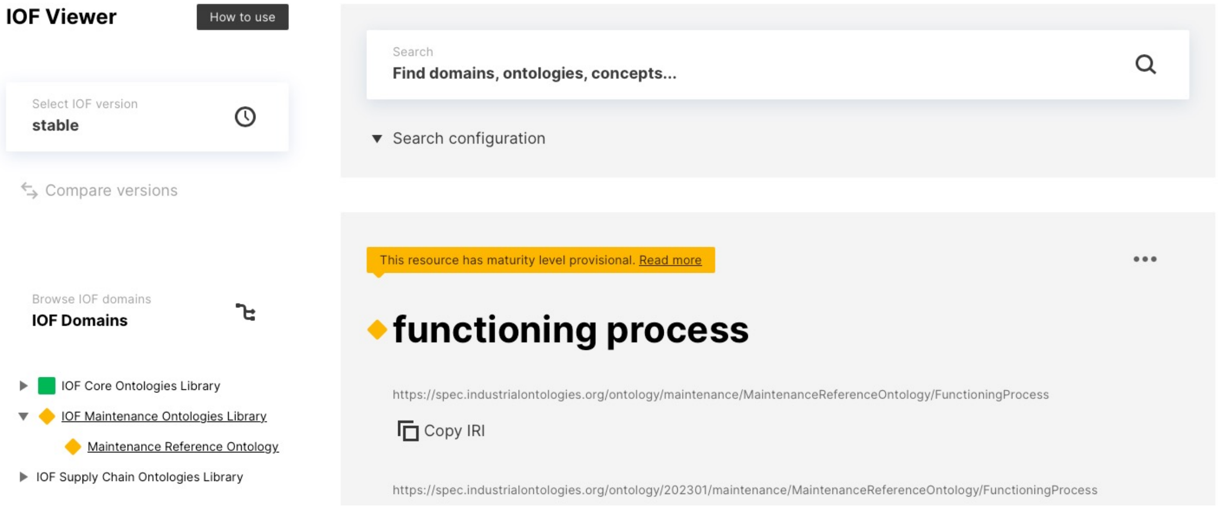
IOF Ontology

inter-operable, cross-functional, cross-industry, data-model-driven, extensible

GitHub repository

Data model for functioning process





Select IOF version
stable

This resource has maturity level provisional. Read more

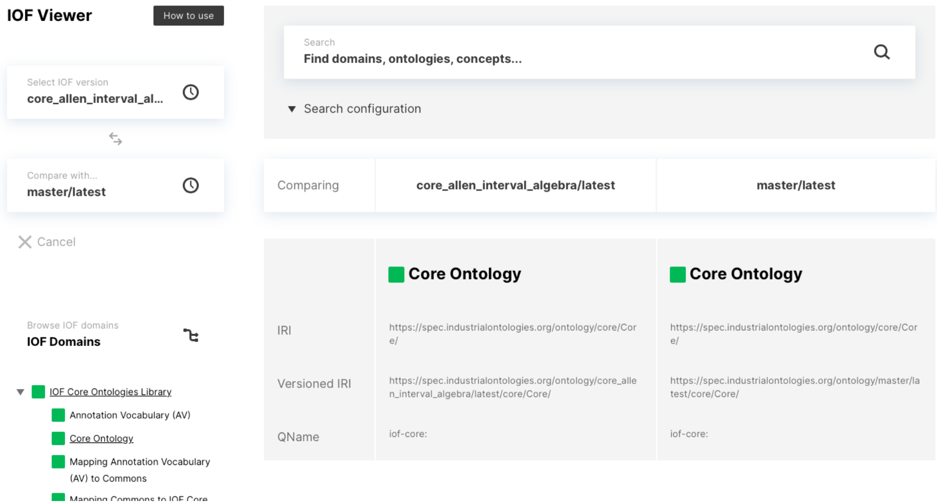
◆ **functioning process**

https://spec.industrialontologies.org/ontology/maintenance/MaintenanceReferenceOntology/FunctioningProcess

https://spec.industrialontologies.org/ontology/202301/maintenance/MaintenanceReferenceOntology/FunctioningProcess

Browse IOF domains
IOF Domains

- ▶ 🟩 IOF Core Ontologies Library
- ▼ ◆ IOF Maintenance Ontologies Library
  - ◆ Maintenance Reference Ontology
- ▶ IOF Supply Chain Ontologies Library



Select IOF version
core_allen_interval_al...

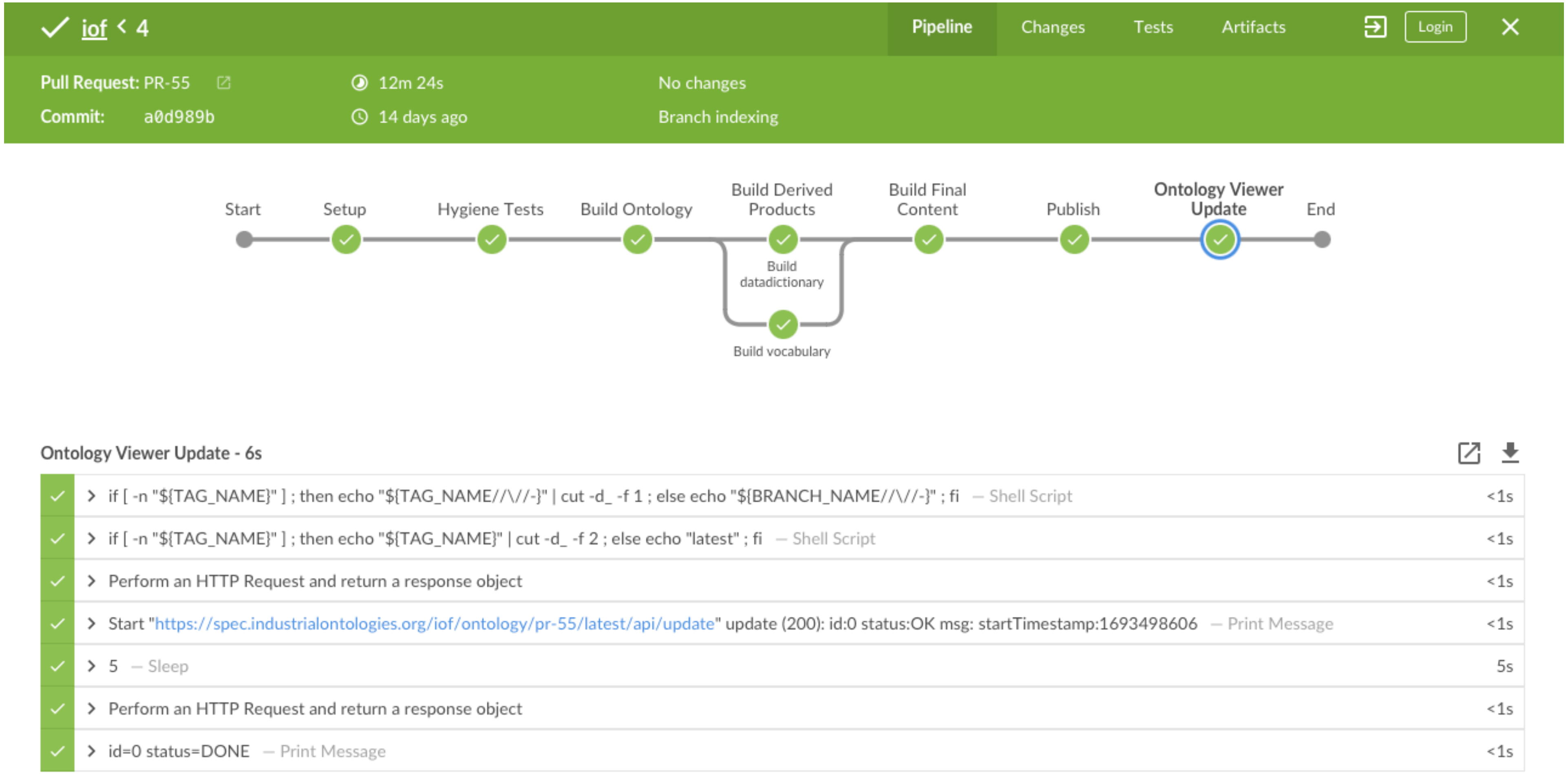
Compare with...
master/latest

Comparing	core_allen_interval_algebra/latest	master/latest
	🟩 Core Ontology	🟩 Core Ontology
IRI	https://spec.industrialontologies.org/ontology/core/Core/	https://spec.industrialontologies.org/ontology/core/Core/
Versioned IRI	https://spec.industrialontologies.org/ontology/core_allen_interval_algebra/latest/core/Core/	https://spec.industrialontologies.org/ontology/master/latest/core/Core/
QName	iof-core:	iof-core:

Browse IOF domains
IOF Domains

- ▼ 🟩 IOF Core Ontologies Library
  - 🟩 Annotation Vocabulary (AV)
  - 🟩 Core Ontology
  - 🟩 Mapping Annotation Vocabulary (AV) to Commons
  - 🟩 Mapping Commons to IOF Core

CONTINUOUS INTEGRATION/CONTINUOUS DEPLOYMENT



Pull Request: PR-55
Commit: a0d989b

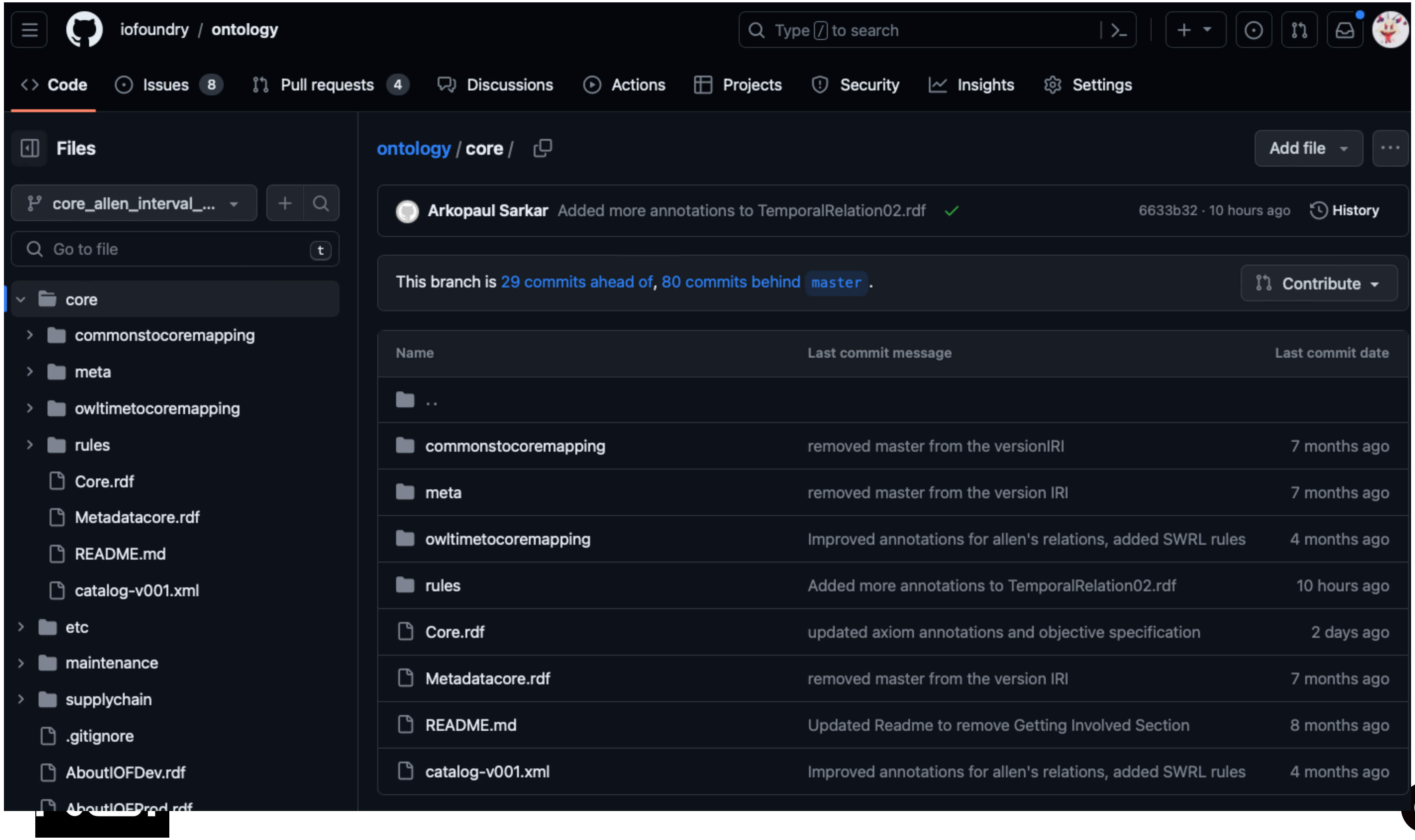
12m 24s
14 days ago

No changes
Branch indexing

Ontology Viewer Update - 6s

```
✓ > if [ -n "${TAG_NAME}" ] ; then echo "${TAG_NAME//\//-}" | cut -d_ -f 1 ; else echo "${BRANCH_NAME//\//-}" ; fi — Shell Script      <1s
✓ > if [ -n "${TAG_NAME}" ] ; then echo "${TAG_NAME}" | cut -d_ -f 2 ; else echo "latest" ; fi — Shell Script                        <1s
✓ > Perform an HTTP Request and return a response object                                                                               <1s
✓ > Start "https://spec.industrialontologies.org/iof/ontology/pr-55/latest/api/update" update (200): id:0 status:OK msg: startTimestamp:1693498606 — Print Message   <1s
✓ > 5 — Sleep                                                                                                                          5s
✓ > Perform an HTTP Request and return a response object                                                                               <1s
✓ > id=0 status=DONE — Print Message                                                                                                   <1s
```

CONTENT REPOSITORY



- core
  - commonstocoremapping
  - meta
  - owltimetocoremapping
  - rules
  - Core.rdf
  - Metadatacore.rdf
  - README.md
  - catalog-v001.xml
- etc
- maintenance
- supplychain
- .gitignore
- AboutIOFDev.rdf
- AboutIOFProd.rdf

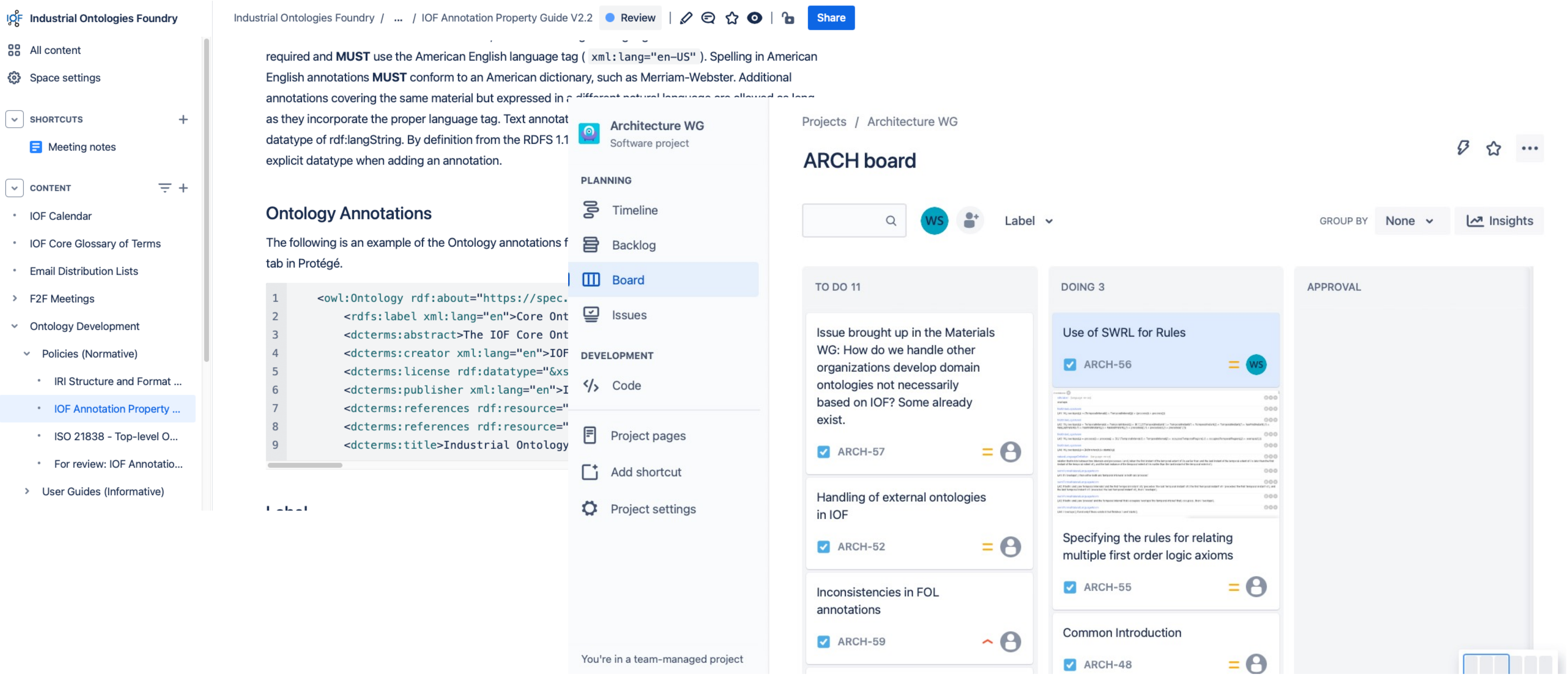
ontology / core /

Arkopaul Sarkar Added more annotations to TemporalRelation02.rdf ✓ 6633b32 · 10 hours ago

This branch is 29 commits ahead of, 80 commits behind `master`.

Name	Last commit message	Last commit date
..		
commonstocoremapping	removed master from the versionIRI	7 months ago
meta	removed master from the version IRI	7 months ago
owltimetocoremapping	Improved annotations for allen's relations, added SWRL rules	4 months ago
rules	Added more annotations to TemporalRelation02.rdf	10 hours ago
Core.rdf	updated axiom annotations and objective specification	2 days ago
Metadatacore.rdf	removed master from the version IRI	7 months ago
README.md	Updated Readme to remove Getting Involved Section	8 months ago
catalog-v001.xml	Improved annotations for allen's relations, added SWRL rules	4 months ago

JIRA AND CONFLUENCE


Industrial Ontologies Foundry
All content
Space settings
SHORTCUTS
Meeting notes
CONTENT
IOF Calendar
IOF Core Glossary of Terms
Email Distribution Lists
F2F Meetings
Ontology Development
Policies (Normative)
IRI Structure and Format ...
IOF Annotation Property ...
ISO 21838 - Top-level O...
For review: IOF Annotatio...
User Guides (Informative)
Industrial Ontologies Foundry / ... / IOF Annotation Property Guide V2.2
Review
Share

required and **MUST** use the American English language tag (`xml:lang="en-US"`). Spelling in American English annotations **MUST** conform to an American dictionary, such as Merriam-Webster. Additional annotations covering the same material but expressed in a different natural language are allowed as long as they incorporate the proper language tag. Text annotat datatype of rdf:langString. By definition from the RDFS 1.1 explicit datatype when adding an annotation.

Ontology Annotations

The following is an example of the Ontology annotations f tab in Protégé.

```
<owl:Ontology rdf:about="https://spec.
    <rdfs:label xml:lang="en">Core Ont
    <dcterms:abstract>The IOF Core Ont
    <dcterms:creator xml:lang="en">IOF
    <dcterms:license rdf:datatype="&xs
    <dcterms:publisher xml:lang="en">I
    <dcterms:references rdf:resource="
    <dcterms:references rdf:resource="
    <dcterms:title>Industrial Ontology
```

TO DO 11

- Issue brought up in the Materials WG: How do we handle other organizations develop domain ontologies not necessarily based on IOF? Some already exist. ARCH-57
- Handling of external ontologies in IOF ARCH-52
- Inconsistencies in FOL annotations ARCH-59

DOING 3

- Use of SWRL for Rules ARCH-56 WS
- Specifying the rules for relating multiple first order logic axioms ARCH-55
- Common Introduction ARCH-48

APPROVAL

Arizona State University

THE EPILOGUE

- Implementation
  - Tools and technology in place
  - Structure and rules of for naming defined
  - Begin introducing process in key domains areas
  - Iterate on process, refine as we go