

ISO 15926 Reference Data Engineering Methodology



Ontology Summit 2013,
track C
7th February 2013r.

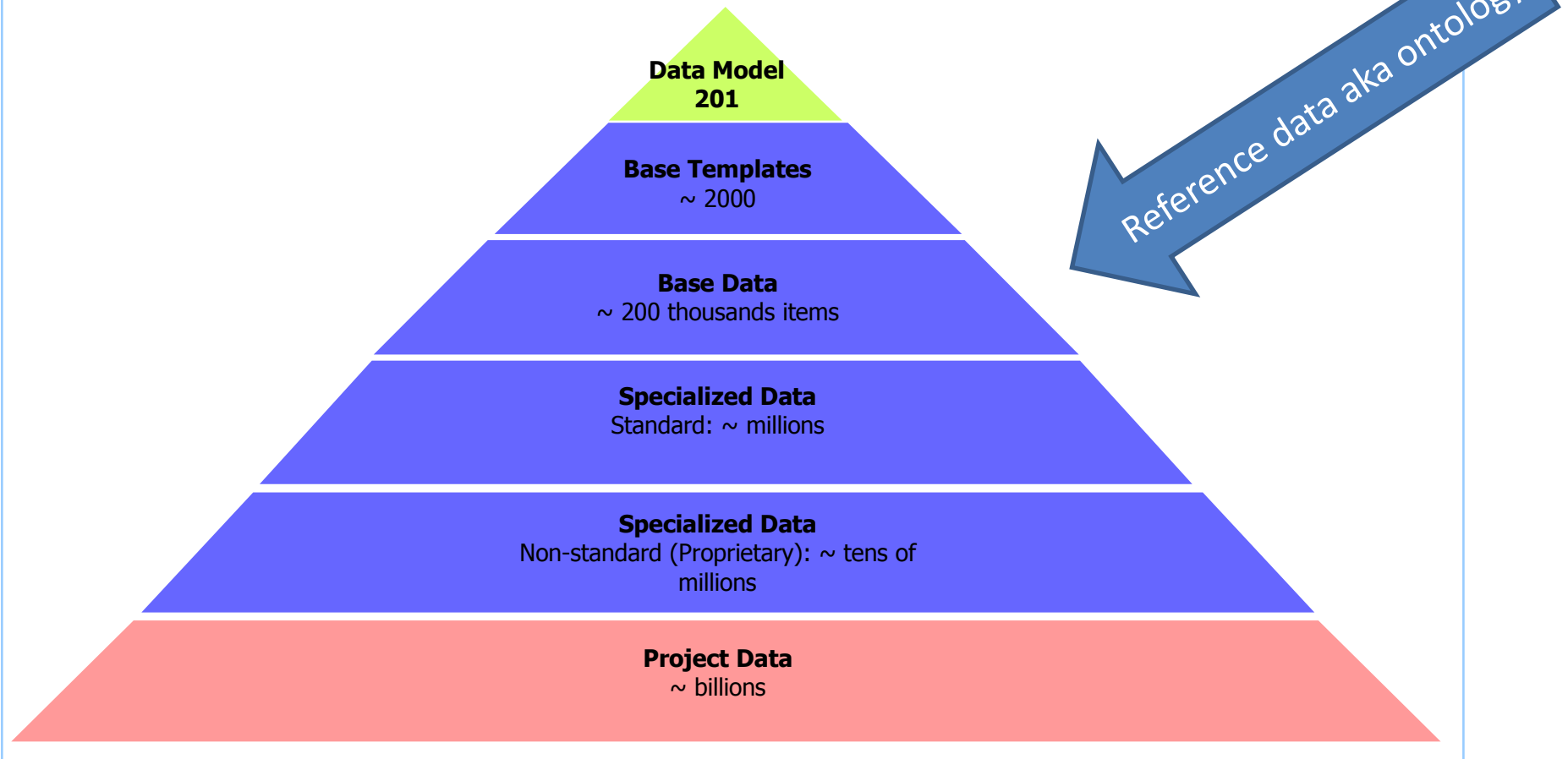
TechInvestLab

- Strategy consultants (systems and organizational engineering, technology management)
- Special expertise: ISO 15926 (since 2008)
- Moscow, Russia

Domain: ontology-based systems federation

- Overview in my presentation at Ontology Summit 2012 (http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2012_03_01).
- Details: ISO 15926 self-education sequence: <http://levenchuk.com/2012/10/01/iso-15926-self-education-sequence/>
- Reference data = ontology

Life Cycle Data Pyramid



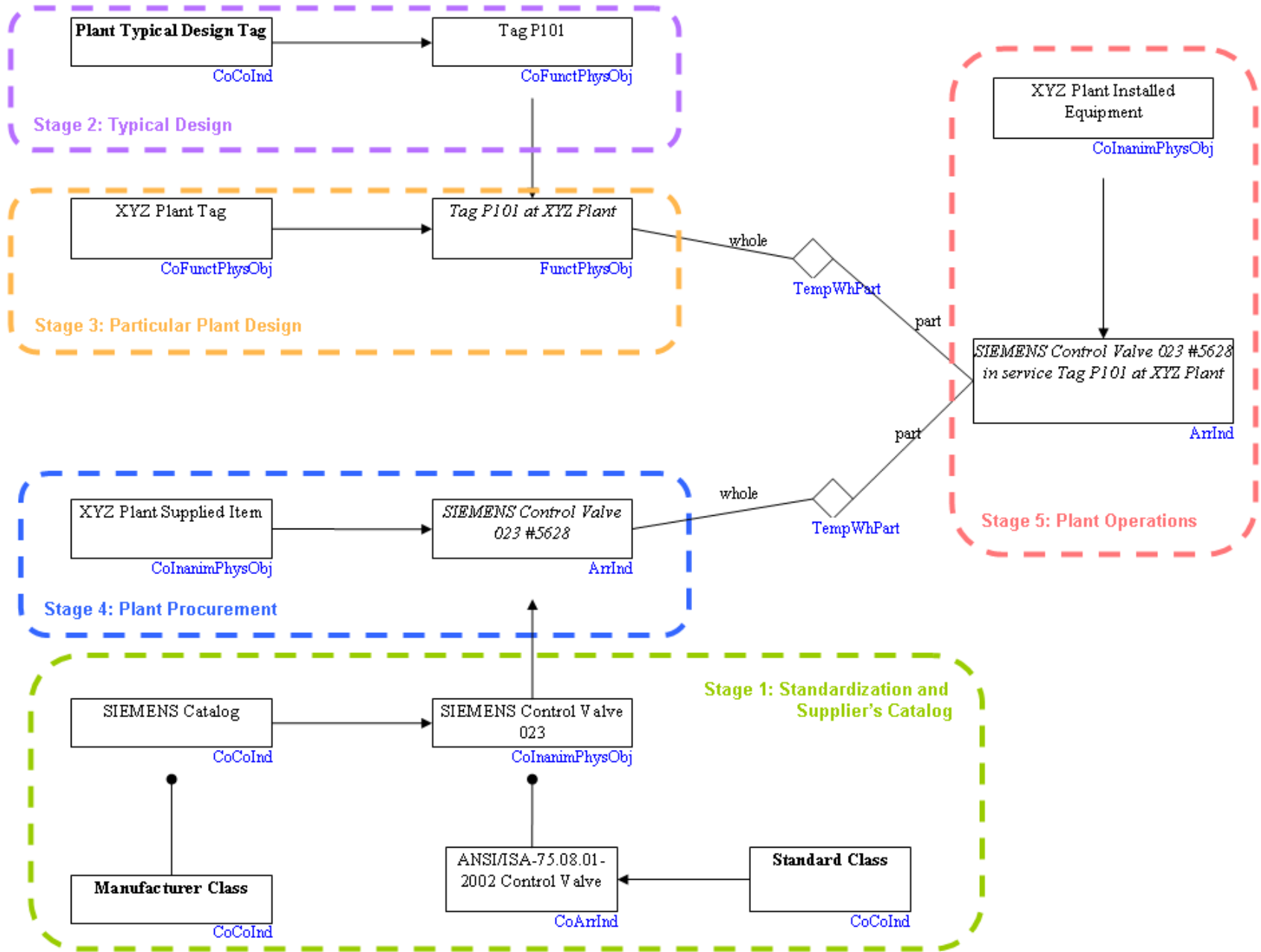
Base data (“domain textbook” level), standard data, non-standard (software schema, corporate standards) require different reference data engineering methods that composed from shared and reused method components and still can use different practices

Numbers is about eventual state, not present (actual JORD RDL is ~2.8mln triples now)

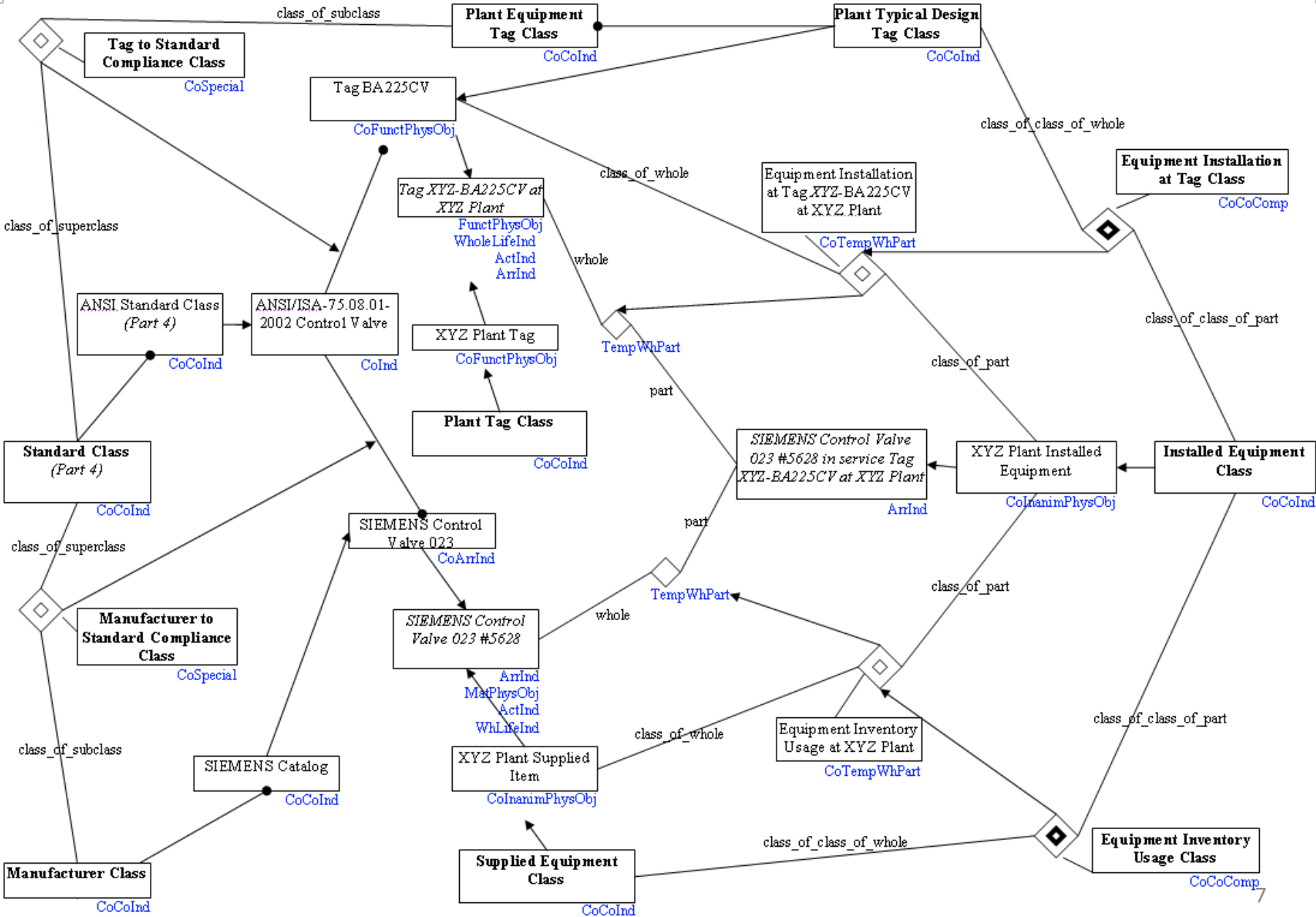
Why we need reference data development methodology

- Self-education sequence give no guidance about “ontology process” (reference data life cycle, methodology, etc. – there are many synonyms for this). People after self-education still have absence of understanding of whole method context for single data element characterization practice (ontologizing per se, “ontology science”, not ontology engineering still).
- Management had no information about what a strange work “ontology engineering”, “data modeling”, “reference data engineering” they need to approve (seen multiple development attempts failed).
- This is not MDM (master data management) projects, we need have common understanding and cooperation/collaboration between different RDL owners/developers. Way of work should be explicit to be discussed and improved.
- We (TechInvestLab) started development of .15926 Editor and needed explicit activity description to have requirements understanding for our ontology software.

ISO 15926 Equipment Life-Cycle example (short form)



ISO 15926 Equipment Life-Cycle Example (more reference data)



Reference data in .15926 Editor

<http://techinvestlab.ru/dot15926Editor>, freeware

The screenshot displays the .15926 Editor v1.1 interface. The top menu bar includes File, Edit, View, Data types, Common files, Common endpoints, Other sources, Import, and Help. Below the menu is a toolbar with various icons for navigation and editing.

The main workspace is divided into several panes:

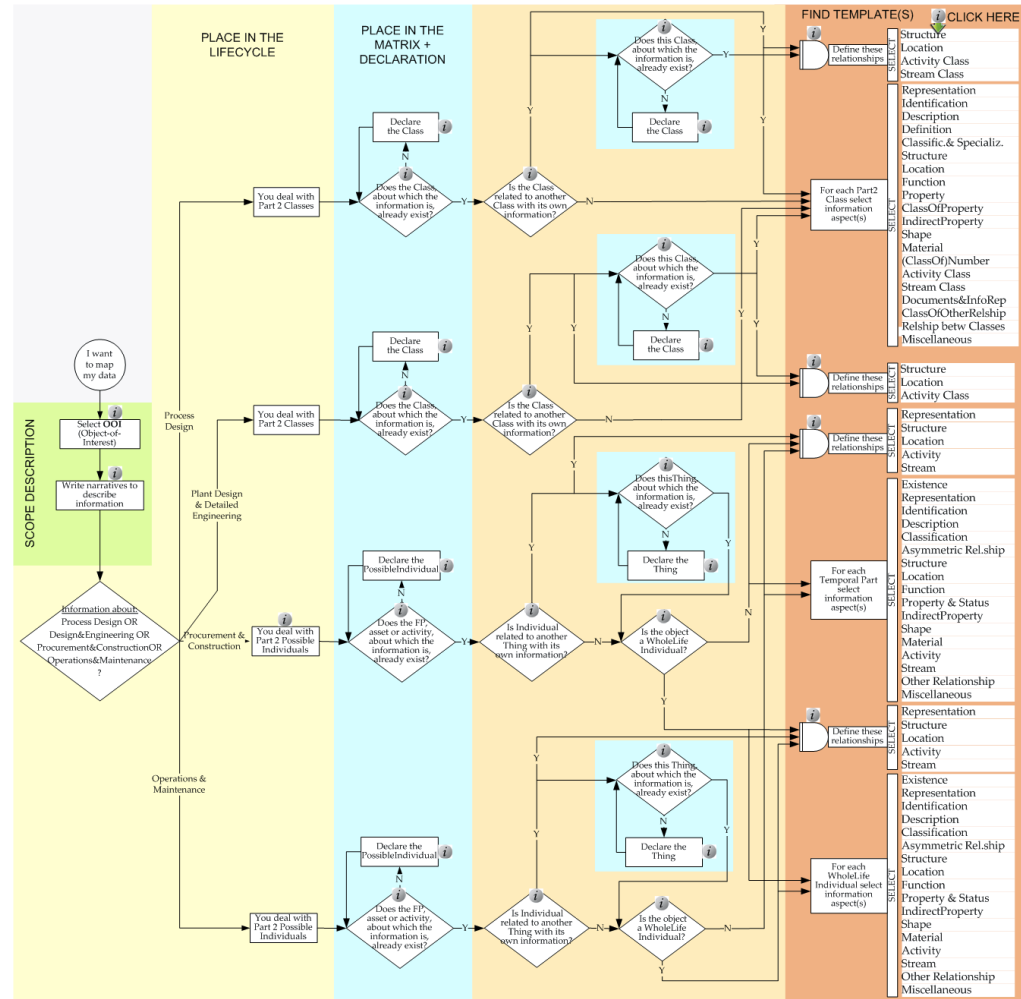
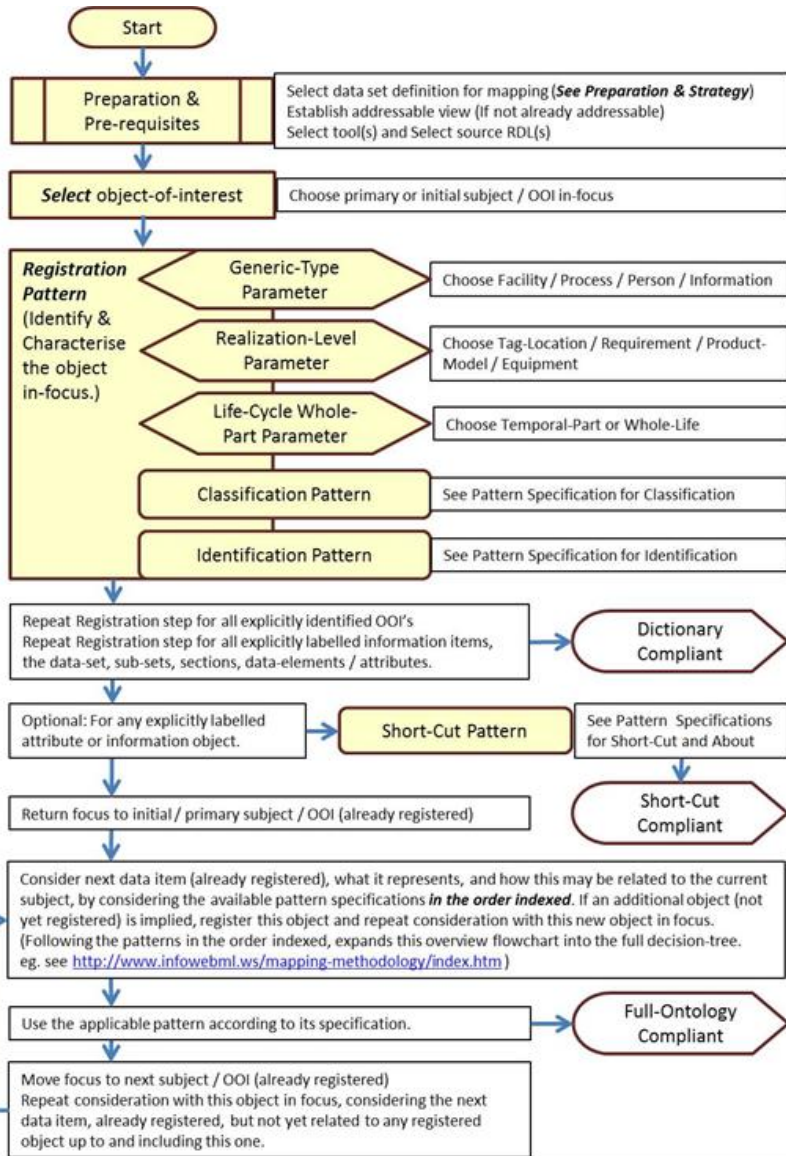
- Project:** A tree view showing the project structure. The selected project is "JORD RDL", and the selected part is "Part 2 types".
- Part 2 types:** A list of classes and relationships. The "Classification" class is highlighted. The list includes: ClassOfStatus, ClassOfSubAtomicParticle, ClassOfTemporalSequence, ClassOfTemporalWholePart, ClassOfUsageOfRepresentation, Classification, ComparisonOfProperty, CompositionOfIndividual, ConnectionOfIndividual, ContainmentOfIndividual, CoordinateSystem, CrystallineStructure, Definition, and Description.
- JORD RDL:** A search results pane for the term "pump". It shows a list of pump classes and their relationships, including: API 610-VS6 CENTRIFUGAL PUMP, API 610-VS7 CENTRIFUGAL PUMP, ARCHIMEDIAN SCREW PUMP, ARMoured PUMP, ASH PUMP, ASME B73.1M CENTRIFUGAL PUMP, and ASME B73.1M PUMP CLASS. It also shows sub-panes for Properties (9), Relationships (3), and Patterns (3).
- Property Value:** A table showing the properties of the selected class. The table is as follows:

Property	Value
PCA RDL URI	http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003#
RDS/WIP URI	http://dm.rdlfacade.org/data#Classification
Part 8 URI	http://standards.iso.org/iso/15926-8/ed1.0/
Name	Classification
Comment	A [Classification] is type of [Relationship] that indicates that the classified [Thing] is a member of the classifier [Class]. [Classification] is not transitive.

The bottom pane shows the "Property Value" table for the selected class, with the following data:

Property	Value
Name	JORD RDL
Location	C:\Users\ailiev\Documents\INCOSE\dot15926\RDL.owl
Module name	
Part 2 namespace	http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003#
Namespaces	xsd http://www.w3.org/2001/XMLSchema# rdf http://www.w3.org/1999/02/22-rdf-syntax-ns#

Patterns (intrinsic aspects)



Situational Method Engineering

«There is no such thing as universal method. We will compose new method of its components for every particular development project».

Ontologies (terminologies, meta-models) of development methods:

- OPFRO OPF (2009, Open Process Framework, <http://opfro.org>) – more than 1100 (systems and software) engineering method components
- ISO 24744 (2007, Software Engineering -- Metamodel for Development Methodologies, http://www.iso.org/iso/catalogue_detail.htm?csnumber=38854)
- OMG SPEM 2.0 (2008, Software & Systems Process Engineering Metamodel specification, <http://www.omg.org/spec/SPEM/2.0/>)
- Contemporary: draft OMG Essence – Kernel and Language for Software Engineering (next submitting to OMG will be 18th February, 2013), <http://semat.org/wp-content/uploads/2012/02/2012-11-01.pdf>

Application of SME to ontology development

- “Software process” is primary domain for SME
- Systems engineering is secondary (for SPEM 2.0 it is already in name, Essence have remarks about it in the future)
- My point: “programming, ontologizing, modeling are all the same domain, but different communities and their favorite terminology”
- SME is fully applicable to ontology development
- Extrinsic aspects explicitly covered by SME, intrinsic aspects require additional discussion.

ISO 15926 Reference Data Engineering Methodology

- Developer: TechInvestLab
- Version 3, 13 september 2011, 19 pages --
http://techinvestlab.ru/files/RefDataEngenEnglish/RefDataEngen_ver_3_English.doc
- Explicit (but informal) usage of ISO 24744 as «development method description checklist».
- Development = engineering, not science (like computer science vs software engineering: algorithmics vs requirements, testing, etc.)

Method elements described

Following ISO 24744 standard of method description, for ISO 15926 reference data engineering we described:

- work products;
- system of interest (reference data) life-cycle stages ;
- processes performed during a life-cycle;
- organization (roles and tools applied);
- data modeling languages/notations.

In the current version of the method only the tasks of a single role are described: tasks of a “reference data engineer”, also called “data modeler”. Roles of a “model tester” and a “project data administrator” are mentioned but not elaborated.

Work Products and Tools

Work Products

- Reference data (consists from reference data items: classes, individuals, templates, template instances). [we have now more: patterns]
- Project data
- Reference data libraries
- Mappings

Tools

- Reference data editor
- Mapping editor
- Adapter

Languages

- Part 2 language [part 3 for geometry]
- Template language
- RDF and OWL [serialization]
- [was missed in v3] Mapping language, patterns



Reference data (aka ontology for system federation/data integration projects) life-cycle

Stages:

1. Project dataset identification
2. Identification of Data Description Items.
3. Characterization of Data Description Items.
4. Mapping
5. Project data transfer
6. Verification of transfer results

Reference data engineer role

The professional role of a reference data engineer requires the following competencies:

- Knowledge of technical English.
- Understanding of the set theory and logic in the curriculum of mathematics for engineers.
- Knowledge of all Parts of ISO 15926.
- Good communication skills (to be able to speak with the specialty engineers about data characterization).

There are two main levels of qualifications here:

- “Black belt” qualification allows development and evaluation of data for the registration in the reference data libraries at the levels higher than enterprise level – global, national, industry.
- “Yellow belt” qualification allows development and evaluation of data prepared for the registration in the RDLs at enterprise or project levels – where data items are almost always specializations of higher level data.

Lessons learned

- Good as possible checklist (e.g.: what is your dataset?!), but no clear checkpoints. Should be performed “checklistization”.
- V.3 outdated now (more than one year of ISO 15926 community development -- pattern language, mapping practices, new experience, new tools, more understanding of intrinsic aspects).
- Not a good guidance during engineering endeavour (What is the health of a project? What we should do now? What is abstract notions, and what is concrete artifacts we should use in the work?)
- Iterative but looks like waterfall, it is misleading.
- Have no explicit mention of practices, only life cycle stages.

What to do:

- Rewrite using contemporary development methodology standard: [draft] of OMG Essence

Essence: separation of Kernel and Practices

- Language: Alpha (has states, states has checkpoints), activity, work products (has detalization level), etc.: defined by Essence
- Kernel (instances of language constructs, abstract):
 - data sets, reference data are Alphas: need to be developed for domain
- Practices (concrete):
 - work products (template sets, mappings as programs)
 - Activities (selecting of data set, writing of mapping program)
- Methods
 - Compositions of practices that suited for particular projects (reference data of text, reference data of Excel table, reference data of CAD database)

Kernel Alphas

(Abstract-Level Progress Health Attribute)

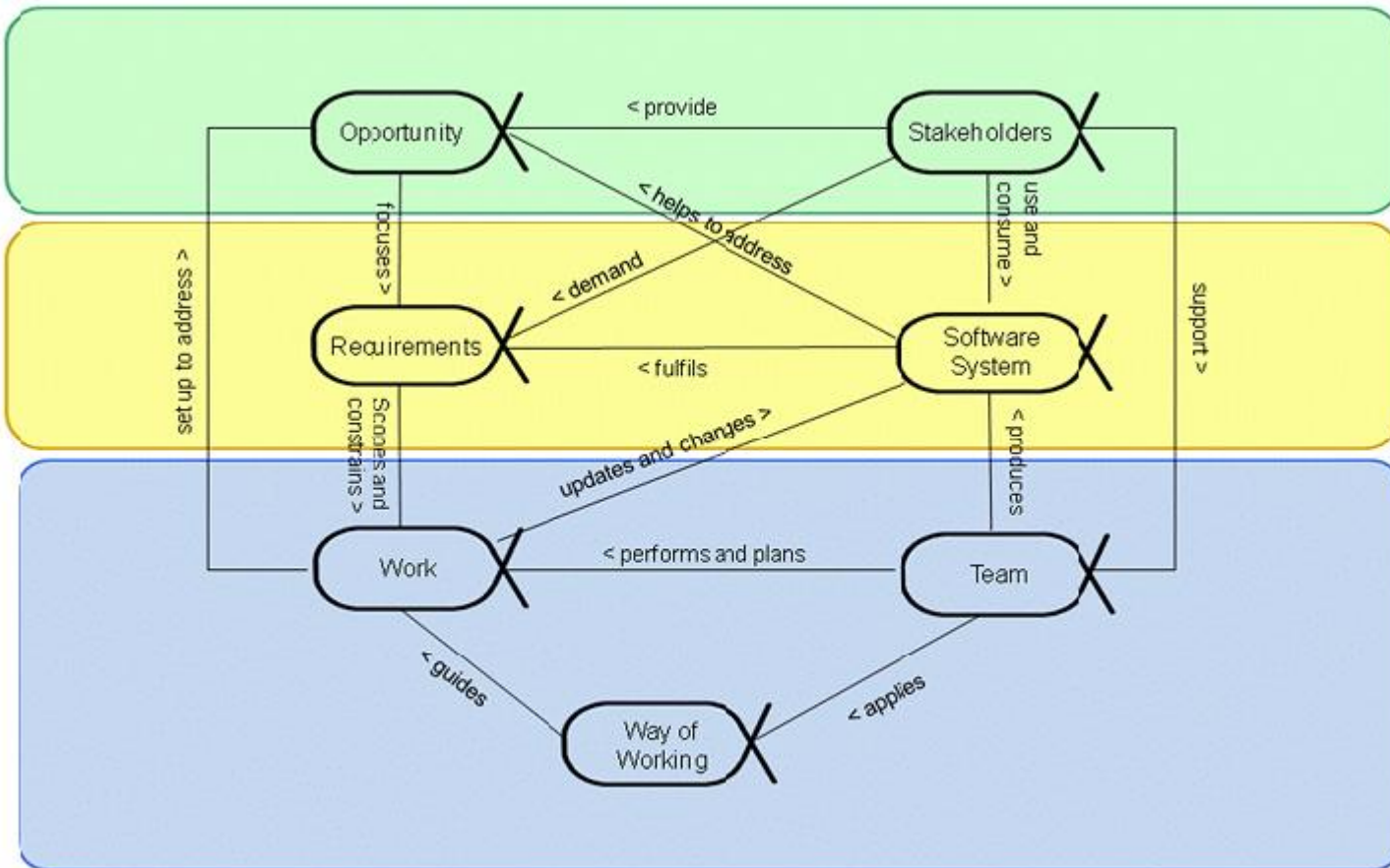


Figure 4 – The Kernel Alphas

- Are we need «Ontology engineering» separate kernel or it is extension for «software engineering» domain?
- Sub-alpha definition needed
- States for all sub-alphas needed
- All of t is before practices (associated work products and activities).

Thank you!

Anatoly Levnehcuk

<http://levenchuk.com>

ailev@asmp.msk.su

Victor Agroskin

vic5784@gmail.com

<http://techinvestlab.ru/ISO15926en>

+7 (495) 748-53-88