ISO 15926 Reference Data Engineering Methodology

TechInvestLab.ru

Version 3.0

September 13, 2011

About ISO 15926 standard

The ISO 15926 reference data engineering method is informally described in this document. The method is designed to be a part of data integration process for diverse computer applications (CAD, PLM, ERP, EAM etc). ISO 15926 Standard is concerned with design and engineering project data and industrial catalogues data transfer between various information warehouses. The standard follows “all-to-all” integration principle and is designed to be used during the whole life-cycle of an engineering project.

The use of the standard for data transfer between applications is sometimes called «ISO 15926 outside».

The use of the standard to design internal structure of data warehouses – the so called «ISO 15926 inside» – has not been fully implemented anywhere, although many PLM vendors claims that their internal data formats are to some extent “ISO 15926 compliant”. However so far data in engineering information warehouses do not conform to any other common standard also.

The data integration task emerges when an application requires data stored in the format not suited for this application. For example, it is not easy for an ERP system to access data from PLM system warehouse and vice versa. Database schemas of different application do not conform to each other: an “attribute” or “relationship” of one may be an “entity” for the other, an object represented in one database schema with a text string may be represented as a group of number and text fields in the other. The need to do a data transfer among applications is growing, as repeating manual re-entry of data from one application to another devours expensive hours of work of qualified employees, and is error-prone. The number of required re-entries in a big engineering project may approach a dozen.

Reference data engineering for “ISO 15926 outside” is one of several disciplines dealing with master data in order to achieve data quality in life-cycle data management. The principal difference of this technology from the known technologies in master data management is that ISO 15926 standard from the very beginning assumes the use of the same reference data by independent organizations (others often proclaim hat “reference data cross the boundaries of an enterprise” and proceed with the discussion of only one organization). ISO 15926 standardizes access to a master data in a single format, and prescribes storage of master data in a federation of administratively independent reference data libraries. That is why ISO 15926 technology is well suited for data integration not only in an above-average engineering company or a holding company, but also for the industry-wide data integration or even for projects involving several industries.

Reference data engineering for “ISO 15926 outside” also differs from the traditional approaches to implement the STEP family of data exchange standards (contemporary versions of PLCS – AP239, or the coming AP242). ISO 15926 standard covers the whole life-cycle of design and engineering data, including the system utilization stage. Using an original time representation concept (“4D space-time” paradigm), ISO 15926 provides the compact framework not only for the “design and engineering data”, but also for the data required to manage continuously changing configurations of industrial products and working facilities (for example, details of spare part installation). Besides that, ISO 15926 is not restricted to a single engineering discipline, instead it is principally extensible, and thus comparable not with a particular STEP application protocol, but with the whole AP family of this standards).

Further reading of this methodology assumes at least superficial knowledge of the parts 1, 2, 4, 6, 7, and 8 of ISO 15926 (the last three parts exist only as working drafts in ISO), and knowledge of system engineering basic terminology (according to ISO 15288: system of interest, life-cycle, process, etc).

Following ISO 24744 standard of method description, the following method elements are described below:

- 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.

Information, Data, Information Objects

Reference data engineering deals with concepts from three separate domains:

1. Information – these are facts, considerations, orders, requirements, opinions, etc. Talking of information is talking about its substance – the content of the information in the context of its use ("the meaning"). What this information means for our project, what are its consequences? What is the meaning of getting exactly this information, and not the other? What situation in the real world corresponds to this information?

Discussion of meaning is usually not related to the discussion of a notation, naming agreements, syntax, and specifics of information presentation in particular media. Meaning is a meaning and not a form. Information that 2*2=4 is discussed at the level of meaning (that it is exactly 4, not 3 and not 7), and not at the level of presentation (e.g. what numeral system is used to write numbers) or media used.

Information is abstract, it doesn't exist in the material world – but it can represent knowledge about a real world object. Engineering design project information contains what is meaningfully known about the project, independent of the way of presentation, encoding, storage format.

Information is produced and consumed by engineering project participants – builders, heating engineers, managers.

2. Data – this is information presented in a particular formalism. For example, discussing the data in "2*2=4" one may discuss the formalism used – Arabic or Roman digits.

Data are abstract, it doesn't exist in the material world. Indeed, the same data (for example, a simple text string "pipe" encoded in UNICODE, or a large database with a complex schema) can be presented in different instances on a media, including reserve copies (back-ups). The data of all the instances are the same, and should be discussed specifically as data.

Data are the domain of reference data engineers (data modelers), other modelers, programmers, database administrators. Database administrators maintain the consistency of the data and conformity of the information presented in the data to the chosen formalism.

3. Information objects – these are newspapers, magazines, documents (in paper and in electronic form – as files), total computer data warehouses (intended for storage of data in a structured form, without repetitions and suitable for unambiguous search) and any other physical objects storing data. In different information objects / on different media (for example, in a newspaper and in data storage) the same data may be stored (e.g., the text string "pipe").

Data do not live without media. "A file" usually is a set of small magnetic particles, oriented in a special way and placed on a disc made from non-magnetic material. Data from a file may than be loaded to the memory – and becomes there a part of an integrated circuit with different voltage levels.

Information objects lie in the domain of system administrators, whose responsibility is not the consistency of the data stored in those objects, but its safety (via, e.g., back-ups), or distortion during transfer from one media to the other.

Reference Data

Reference data – is the name for the data used in several projects, and/or at different life-cycle stages, and/or data of interest to different users. National Registry of Legal Entities, industry products catalogue or method catalogue, tax rate list, database schema of a geo-information system, "data schema" for 3D models of a building construction or of an industry equipment – all these are reference data examples.

Reference data are really data, and not information, i.e. formal notation used for presentation is important. Reference data include not only "the data for end use", i.e. filled forms (for example, catalogues), but also "the data describing the structure and the meaning of other data", i.e. templates/forms to be filled.

Ability to speak the database design language is not of great use in this case: ISO 15926 does not make a traditional division between "database schema" and "data in a database". There is no rigid boundary between the form and its content, instead, there is a formalization of "templates for template representation", and many more levels within the same formalism. A "database schema" in ISO 15926 falls under the reference data, and the data defined with this "schema" may in turn be used as a "schema" for some other "data", thus becoming a reference data. To avoid confusion, let's meanwhile forget about the databases, we shall return to them when we'll be discussing the application of ISO 15926 to the data in specific warehouses.

Types of Reference Data

Reference data, created according to ISO 15926, consist of reference data items.

Reference data items are of the following kinds:

- Classes of individuals, classes of relationships and classes of classes, used for breaking down and grouping of individuals. By no means mix up these classes with classes from the object-oriented programming! Classes in ISO 15926 are from the set theory, they do not have "methods" and do not hold "objects". The set theory used is also not classical, actually the sets are non well founded, but this is not really important at the level of this methodology.
- Individuals. Note that in ISO 15926 individuals are only those entities which have extension in space and/or in time, and only individuals intended for use in many projects, e.g. "Moscow" or "ISO" will become reference data. According to the ontological nature of ISO 15926 "Moscow" is seen as the real city, existing on Earth and "ISO" is seen as a collection of real people, offices, documents and computers.
- Templates, used to create statements about individual data. The simplest way to understand what templates are is to imagine them as empty tables, whose column names are called "template roles". They are called "roles" because they define roles for the values written in the correspondent column. For example, "Pipe T123" in one template (defined by some table) may play the role (be written in the column) "Supplied item", and in the other template the same "Pipe T123" may play the role (be written in the column) "location of a flow"

- Template instances, i.e. filled rows of tables-templates, containing (in the case of the reference data engineering) statements about other reference data items.

Templates are by no coincidence compared to tables: the tool common to all engineers of reference data are electronic tables and more often the work with templates is done in Open Calc or MS Excel.

By the level of generality of the real world information represented, there are several categories of ISO 15926 reference data, organized as a pyramid:

- «Data model» – it is a set of 201 entity types from ISO 15926 Part 2, duplicated as classes in reference data;

- Core classes – classes that are a commonly used subdivisions corresponding to terms used in common language. Examples of core classes are pipe, floor, pump, or light bulb.

- Standard classes – classes whose specification for membership is owned or controlled by a standardization body (ISO, national standards bodies, or industry consortia) and is publicly available.

- Specialized classes – classes including de-facto classes, manufacturer classes, proprietary classes and other classes, whose peculiarities are out of the scope of this document. These classes specifications and membership rules can be obtained from the analysis of domain textbooks, enterprise standards, even from specifications of particular projects.

- Proto templates – these are predicates, describing in logical form the 201 entity types of the ISO 15926 Part 2 data model.

- Base templates – these are templates designed to define arbitrary statements restricted only by 201 entity types of the ISO 15926 Part 2 data model. Base templates allow general statements about entities and their relationships, the only requirement being their classification by such general types as "physical object", "Gregorian date and time", "connection", "class of class", etc. The collection of commonly used base templates is maintained at https://www.posccaesar.org/wiki/SigMmt/Templates.

- Core templates – these are templates helping to form statements about the relationships between user data and core classes. Core templates allow statements restricted only by relating entities you describe to core terms/concepts, expressed in the common language "from the dictionaries". Core templates are defined by the restrictions imposed on the roles in base templates. Actually, the same statements can be formed directly from the base templates, but the base template should be chosen to match the particular situation, whilst the core template structure is already "tuned" to that situation.

- Specialized templates – these are templates helping to form the more and more specific statements, stating the relationships between user data and specialized classes and even particular individuals. The use of specialized templates can be restricted in a specific industry or even in an organization. Specialized templates are created from base and core templates by imposing additional restrictions on the template roles. It is assumed that it is easier for a domain expert to find a proper template in a big heap of very specialized templates, compared to construction of the same statement by restricting of very general templates (e.g., it is easier to write “water flows from a tap” than to create a sequence like “a faucet is attached to a pipeline”, “a faucet has a hand-operated drive”, “a pipeline contains a flow”, “temporal part of a flow consists of a liquid phase of H2O”,…).

Project Data

Reference data are used to shape the engineering project data in a form suitable for all computer applications involved. Type of an engineering project may vary: a plant construction, a manufacturer product catalogue, even a business processes reengineering project. Project data from any domain have a common form in a unified language of ISO 15926 – instances of templates (imagining templates as tables – rows in such tables where each column corresponds to a template role).

It can be said that all data present at each stage of an engineering project can be classified either as a reference data (defining the form) or as a project data (defining the content).

For example, at the design stage reference data contain the information that “there are pipes with 100 mm nominal diameter”, “there are pipes with 200 mm nominal diameter” (these might be standard classes defined by national standard). Among the reference data there is a template that allows “to place an object A into a category B” (it is a base template). On its basis one can create specialized template for pipe classification, and populate project data with statements (instances of specialized templates) “object T123 is a pipe with 100 mm nominal diameter”, “object T567 is a pipe with 200 mm nominal diameter 100 mm”, etc.

Note that the statement “there are pipes with 100 mm nominal diameter” was a project data for a person creating the piping catalogue (that was the system of interest for another engineering project!), and the base class “pipe” was a reference data for that catalogue project.

It is easy now to see how your project data with pipe tags on piping diagram become reference data for somebody at the later life cycle stage (e.g. at the utilization stage, when the project data will be about individual numbers of physical items replaced, and the pipe tags will be “reference data” that don’t change).

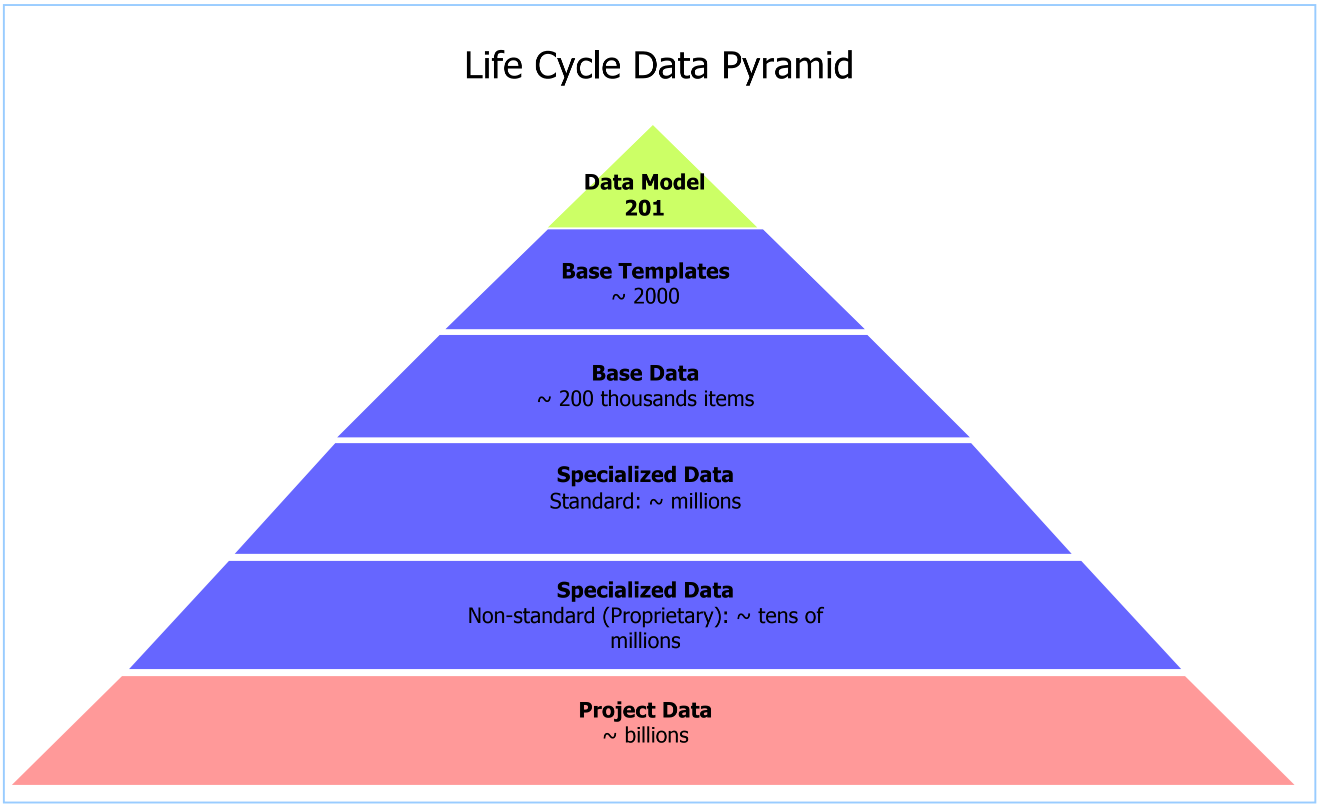
Now we can better specify the definition of the reference data given above: “data used at different life-cycle stages, and/or data of interest to different users”. While the project data are in the hands of its creator (the creator of the information) they are subject to change, completion and revision, or transfer to another user. Starting from the moment of a first use of these data by that another user to describe his/her own data (usually it happens at the next stage or at least next steps of the life-cycle) – the data become the reference data that cannot be easily changed or revised without the risks of the project data consistency loss.

One can say that reference data describe the world at the level of “how the world is in general”, while project data describe “how the things are in particular project at the particular life-cycle stage”. Still, the rigorous logical distinction between the reference data and the project data in borderline cases is difficult, that is why the management of the reference data sets is of great interest in the task of data integration.

Life-Cycle Data Engineering

Life-cycle data for engineered object (nuclear power plant, submarine, forging press) – are all the data that appear and are used at any moment of the life-cycle, from a concept and up to a retirement. Life-cycle data according to ISO 15926 reference data engineering methodology are considered a single whole, and include a "data model" (201 data type from ISO 15926 Part 2), reference data common for many projects, project data (that may become reference data during their own life-cycle).

As we have noted above, "classes", "templates", "individuals" forming the reference data, are named for brevity "reference data items". Reference data items are defined incrementally and may be seen as "a reference data pyramid", which becomes "a life-cycle data pyramid" after addition of the project data. The top of the pyramid is at "the philosophy level of generality", and the bottom contains statements about particular properties of particular individuals in the particular projects.



"Data engineering" in general is the way of transforming information to data according to some formalism. "ISO 15926 Reference Data Engineering" – is the way of transforming information to data according to the formalism prescribed by the Part 2 "data model" and to the rules described in the other parts of the standard.

Reference data engineering for a particular project consists of the following: finding (definition) of templates ("tables") and finding (definition) of classes necessary for their further specialization. These are necessary steps in the construction of data for the description of the structure and meaning of other data (either project data, or another level of reference data).

"Project data engineering" is a specific domain decision making: definition of a flow parameters, or pipeline geometry, or equipment choice, etc. Sometimes project data engineering is the process of transformation into data of the information about the decisions taken by others. Project data have a specific life-cycle, involving special professional roles (engineers, cataloguers, etc) and quite different from that of reference data.

However in the data integration process the reference and the project data life-cycles intersect, this is why processes of project data life-cycle will be touched below also.

Reference Data Libraries

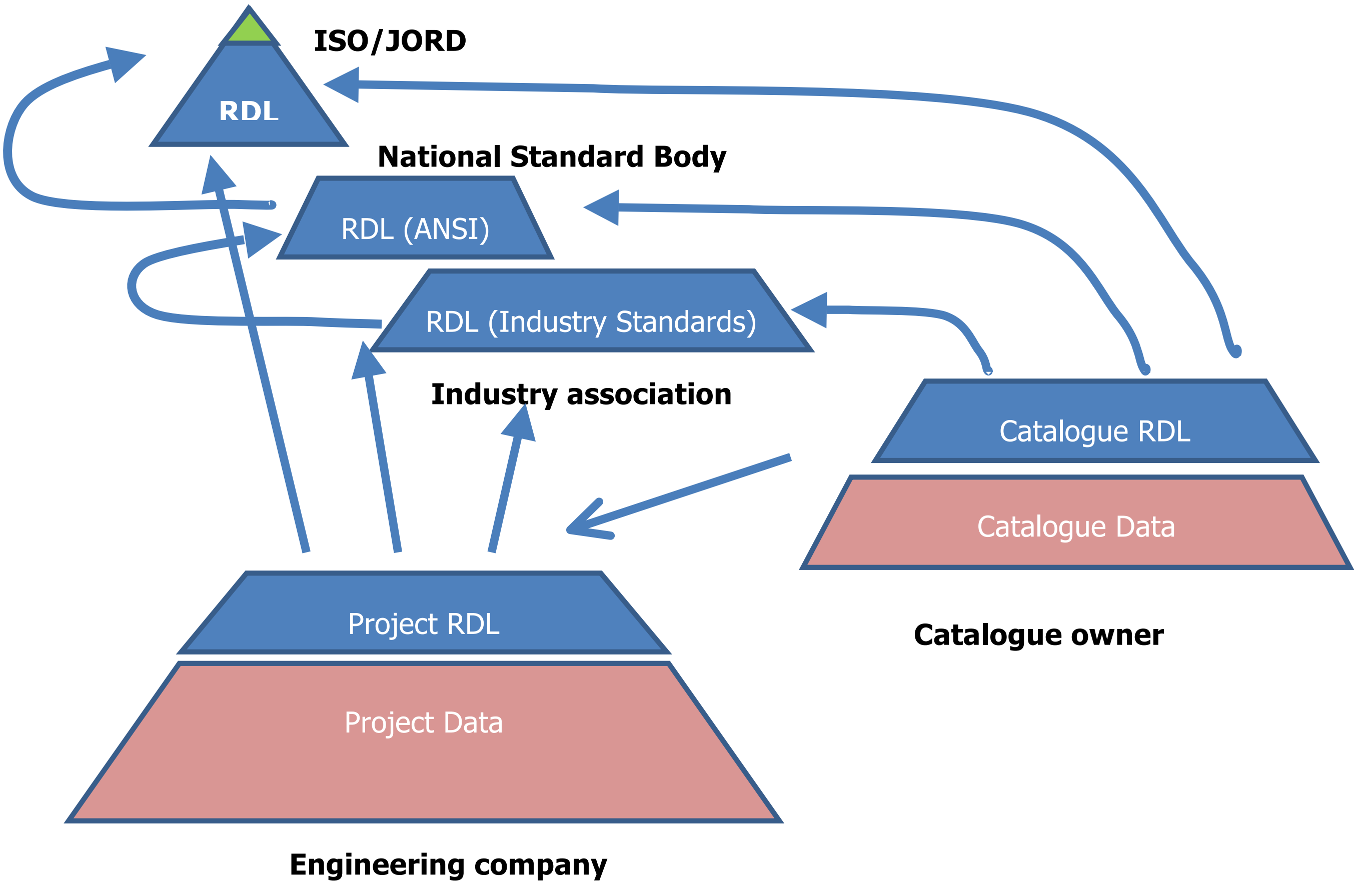
Reference data are being developed by diverse organizations. The set of reference data, administratively controlled within an organization, is called a **reference data library (RDL)**. Reference data libraries may reference one another, using general classes, templates or even individuals from other libraries to create new definitions of specialized classes, templates and individuals. However, in theory any library can contain even base classes and base templates if they are not used by others, or if the library owners did not find them in the available or trusted libraries.

Reference data libraries can be:

- International (see below for information on the current state of affairs).
- National (for example, storing classes and templates extracted from the country body of standards, ANSI, GOST or DIN).
- Industry (an industry here can be defined as some industry association or even a large holding company).
- Enterprise-wide.
- Project related (a project can unite several enterprises from different industries).

Reference data libraries can (and ideally – have to) reference each other, and, as such, form a library “federation” (recall that each of the libraries is managed independently!).

Reference Data Libraries



Reference data library contains “registered” data. As any common reference book assumes the collaboration of authors and verification process, for RDL the data will become acceptable only after a formal approval, not automatically. Public (international and national) RDLs have to declare very specific regulation procedures for addition or change of the reference data, allowing interested persons to propose such additions or changes.

It is advisable to have a “sandbox” within a library, where one can create, verify and test reference data preparing them for the formal approval.

Reference data libraries should support unique identification of reference data items. Currently, in the frame of iRING project there is a mechanism of centralized assignment of unique identifiers, but how these identifiers will be moved to the library federation mentioned above (with the start of JORD project) is still an open question.

Mapping

Any data (both project and reference) are stored on some media. Nowadays such media are data warehouses – “database” or “repository” data management software installed on reliable server hardware. Various data handling applications and their respective data warehouses are being created separately for design engineers and purchase managers, schedulers and accountants. If we go deeper, even within the engineering domain separate project data warehouses are being used by various disciplines - civil constructors, HVAC engineers, piping engineers, electrical engineers, automation and control specialists and a lot of other professions.

The aim of “project data integration” is to ensure availability of various data warehouses for all specialized applications requiring their data (speaking in terms of processes), or to create a “common project information space” (speaking in terms of data infrastructure).

Direct “point-to-point” data exchange among the warehouses in this situation is simply impossible: the data from the different warehouses are defined (and formalized) differently (have different data models), leading to the creation of a set of programs operating “point-to-point” and resulting in huge expenses – the number of pairs is growing quadratically with the number of points.

With ISO 15926 data transfer among data warehouses is performed by the translation of the project data into neutral form, independent from the warehouse data model. Reference data engineering ensures availability of neutral templates for project data transfer. The data exchange schema becomes significantly simpler compared to “point-to-point”. The data from any warehouse (for example, P&ID CAD) are now transformed in the neutral format (the set of templates suitable for this project), which is defined with continually updated ISO 15926 reference data, instead of translating it directly into other data warehouse format used in the project (PLM from different vendors, for example).

In order to use project data from one warehouse in the other warehouses according to ISO 15926 it is necessary to write an “adapter”, that takes the native data from the warehouse and exports it into ISO 15926 format. This allows communicating warehouses (containing application data, including CAD data) “to see” each other uniformly (using the Part 9 terminology, they are presented as “façades”, though in this text we omit technical details).

This “all-to-all” interaction supported by the “common language” we shall call the “**data integration**” (as opposed to “**data transfers**”, when the “point-to-point” transfer is agreed between parties explicitly – HVAC to piping, scheduling to accounting, facilitating data transfers on case by case basis). Format and reference data of ISO 15926 are used here only for data exchange, and not for data storage or processing, thus such method is called “**ISO 15926 outside**”.

Each adapter contains a dictionary to translate data from “native” warehouse language into common ISO 15926 language. This dictionary is called a **mapping**. To facilitate the translation of project data into ISO 15926 common language it is necessary to ensure the possibility for the project data warehouse adapter to link to reference data items (classes, templates and template roles) in the mapping. Unique identifiers assigned by the reference data library are used for this purpose.

Mapping can roughly be seen as a table with two columns. The first column contains the warehouse project data description items (items defining the “warehouse language”, its set of tables). These items are alternatively called metadata, data schema, data model, or are present in warehouse as classifiers or master data items. The second column contains links to classes, templates and template roles from ISO 15926 reference data.

Reference Data Warehousing

Reference data, as well as project data, are stored in data warehouses in the form prescribed by the warehouse software (relation database, object-oriented repository, triple store, etc).

To be available for use reference data should be presented in the form compliant to ISO 15926, i.e. they should be also transformed by appropriate adapter according to the mapping from a warehouse format to ISO 15926 formats – as described in Parts 8 and 9 of the standard.

Reference data mapping is much easier compared to the mapping of the project data, and it may be omitted – it is usually built in during the reference data library software development, and reference data engineers do not spend time on this task. An exception is the case of transformation of some big third-party database into the reference data for the "common information space" (a manufacturer products catalogue can serve as an example). In this case the catalogue data is treated as project data and complete mapping is built for catalogue warehouse with ISO 15926 adapter to store this catalogue as reference data for future use.

Upon request RDL should provide the complete description of reference data items used in the mapping, including restrictions and cardinalities, template axioms, metadata (creation date, author data, synonyms in foreign languages), etc.

Today it is possible to create company reference data warehouse with iRING Sandbox software, as a set of XMpLant files (XML format, currently not compliant to Parts 8 and 9 of the standard), or as a set of OWL files (in the format prescribed by Part 8 of the standard).

Reference Data Libraries Today

At the international level there should be an ISO controlled reference data library. Meanwhile publicly available "official" international reference data library is called PCA-RDS (it is controlled by POSC Caesar Association – PCA). It is not actually a reliable reference data library, it is rather a reference data heap, although even in the current state it is used in some projects (you need to install a Java–client http://rds.posccaesar.org/downloads/PCA-RDS_client.zip, do not access it from browser!).

Hopefully in the future PCA-RDS RDL will be renamed (or even replaced, as data quality grows) in the frame of the JORD project, jointly run by POCS Caesar Association and FIATECH consortia.

As a reliable source of reference data ISO 15926 reference data today exists mostly at the "data model" level (201 concepts), at the core level (concepts that often appear in the engineering field) and slightly at the standard level (international standards, ISO) of the life-cycle data pyramid. The situation with the templates is worse - the set of base templates approved by professional community ("black belts") is unsatisfactory small, and core templates are being developed at the level of specific projects and so far are not publicly available.

Today creation of a "common information space" for some project data is dependant not so much on a qualified personnel (able to create necessary mappings for warehouses used), as on the existence of reference data subset to support the nature of the given engineering project. Without the relevant reference data no mapping could be established.

Meanwhile the best solution is to gather suitable reference data from the warehouses available (it may be a rather modest collection, though), and to develop missing reference data by analyzing every warehouse being integrated. Currently, 80% of all the efforts needed for the "creation of the common project information space" can be spent on the reference data engineering.

The process of collection (finding and/or development) of reference data necessary for making statements in ISO 15926 language from some dataset is called "**characterization of the dataset**".

Creating reference data for the particular engineering project, one should remember the perspective – these data definitely will be of use for other people in other projects and organizations, and it is hard to forecast which of them exactly. For example, there are almost no standard classes, while almost every project requires a lot of reference data of the national level. There are some industry initiatives on ISO 15926 reference data, with the high possibility of effort duplication in different industries.

Development of ISO 15926 reference data is both "pragmatic" and "semantic". It is pragmatic because it serves the needs of data transfers required to achieve a particular goal of a particular project. It is semantic because it is relevant for data integration even in unknown situations, as it works with "data values" and not with the "data meaning". The unity of the "common information space" within one engineering project as well as between various projects is dependant on one basic principle - all reference data are specialization of a general view of the world, predefined by the "data model" – type system declared in ISO 15926 Part 2. Therefore all reference data can be fully described with a set of first-order logic statements, using predicates derived from 201 proto-concepts of this "philosophy-computer esperanto".

Reference Data Life-Cycle

Reference Data Engineer (or "Data Modeler" – the title coming from the heritage of the "database world") facilitates the following stages of reference data incremental life-cycle:

1. Project dataset identification. The dataset to be translated into ISO 15926 format should be unambiguously identified. The examples are: a set of paper pump specifications from different vendors, a volume of industrial installation drawings, a database describing organization structure, design database in a CAD warehouse (in a PLM system – like ENOVIA V6, or in a single CAD database, e.g. SmartPlant 3D). Data transfer from one CAD or PLM warehouse to another is the most frequent task defining the dataset, but the standard can be applied as well for the data transfer from paper (or scanned) document to a computer warehouse.

Identification of the project dataset is the important stage, requiring a lot of time and efforts, mostly because of the number of meetings and negotiations among the data providers and data consumers. Recall that you see initially only information objects (printouts, computer servers and files on disks), as the data are the abstract objects, identified and discussed separately.

It's important to divide project data explicitly: part targeted for transfer and part that will be leaved outside a transfer and used only within one warehouse by compatible applications.

It's important also to answer explicitly the question of configuration and change management: to what extent your data will be "alive". You need to decide, will the transfer include only final results of some stage (e.g. only approved drawings package), or your integration goal is the "synchronous translation" of continually changing decisions taken during the design process as soon as they are recorded in the warehouse.

Special care should be taken to keep track of planned data sources and target of transfer – where they will appear finally.

Depending on the results of this stage the volume of work on reference data engineering will become clear, and the usefulness of work product also depends on this stage completion. A huge effort to create reference data library may be of zero value in real data transfer – if reference data created are not required and at the same time the data really needed for mapping are absent.

Work products of this stage are data sets selected for characterization and the understanding of the whole work context: what are the source and target warehouses for these data sets.

2. Identification of Data Description Items. At this stage you need to define what will be mapped (what items describe project warehouse data in the mapping table) to facilitate the data transfer for each data warehouse selected at the previous stage (both source and target warehouse).

During the identification of project data description items you need to have a substantial understanding (semantic and ontology knowledge) what are the objects in the project dataset (pipes, documents, people,…) and what are the relationships between them (attachment, description, responsibility,…). Additionally, you have to find the metadata required (version numbers, alternative namings, creation dates, original storage places, working language etc.).

The work product of this stage is the list of warehouse data description items. What items will appear in the list depends on the warehouse internal format, or “language” (recall that the “warehouse” may be a pile of paper documents). For different warehouses this list of warehouse data description items will be called differently – metadata, data model, data schema, metamodel. Important parts of this list are attribute lists and classifiers used in warehouse data model definition.

This work is very time-consuming and has a lot of subtleties. For example, be cautious just to use database schema of an identified warehouse directly. Most likely you will find the lack of “words” to build a complete description of data for the transfer into the other warehouse. Drawing files created in a CAD are referencing some items that are not in the database, but can be found in the configuration files or even in the software code. Warehouse database stores data, but not always all the data. Consequently not only the database schema should be analyzed, but also the final documents (e.g., drawings). One should try to discover the project data description items in the software configuration files and sometimes even identify them from software interfaces.

The situation is even more complicated as the engineers may easily ignore the intents behind a warehouse database schema while adopting software to the project. They may use the warehouse database in an unexpected way (for example, store “nominal diameter” in a field named “internal diameter” – and this will remain unnoticed until the data are transferred to another CAD). Again, this can be avoided with careful analysis of output data samples – drawings, completed specifications, printed reports etc.

Sometime the best solution is to ignore a database schema of the warehouses and work directly with the paper printouts (though the database schema should be addressed later, as the adapter can access data from the warehouse only, not from the papers). However, paper documents may help to elicit the important data description items such as fields filled by hand (e.g. signatures), notes, different line types (e.g. a group of data fields singled out by common border in the printout may signal some common essence, possible unnamed in datamodel).

Data description item identification influences the choice of the datasets described at the previous stage, hence these stages are performed iteratively and often overlap in time.

This stage is essentially a research project and should be planned with enough time and resources.

3. Characterization of Data Description Items. At this stage ISO 15926 reference data are created for further use in the mapping. This is the most complex and effort-consuming part of work.

Input work product for this stage is a set of data description items identified during the previous stage (you can visualize them as the filled column of the mapping table for a warehouse adapter).

The focus of this stage is the work with reference data libraries chosen for this project (internal project or enterprise library and external libraries – coming from other enterprises, national, JORD/PCA-RDS).

Output work product consists of reference data items necessary for the mapping and registered at the reference data libraries at appropriate levels.

The characterization consists of the following steps:

- Selection of an interrelated set of data description items (sometimes called an “object information model”).
- Selection of reference data library (libraries) for use – the sources of classes and templates (take a special care about the question of trust – not any library available may be trustworthy – and the choice of the libraries may require unexpectedly deep research).
- Reference data libraries search for the data items (classes and individuals) that can be specialized or used directly for the data description items classification. At this step corresponding reference data items should be found both for project data description items and for their metadata (as a rule, at the level of classes of individuals and classes of relationships), and for their relationships (among classes of relationships). Special attention is paid to such warehouse data description items as attributes, parameters, units of measure. For these entity types ISO 15926 establishes some rather counterintuitive reference data types.
- Addition of missing reference data items (classes and individuals), and establishment of relationships between found and added items. This activity includes type assignment, classification and specialization of the reference data items, and definition of relationship attributes and relationship classes. To complete this step analytic diagrams are constructed in the format of Parts 2 and 7 of the

standard (see the examples at http://www.15926.info/ and http://techinvestlab.ru/iso15926_sample_diagrams). At the end of this step the reference data created should have items correspondent to the lists and classifier facets used to describe the data in the warehouse.

- Template boundary definition and template role identification based on the analytic diagrams prepared (at the previous step). The availability of core, base and specialized templates should be taken into account. The choice should be made: whether to use the templates found in the reference data libraries directly or to create from core and base templates new specialized templates tailored for the project data. The examples of the template definitions can be found at https://www.posccaesar.org/wiki/SigMmt/Templates/.

- Consistency check in the context of the general view of the world ("data model"): if an "individual" is indeed an "individual" and is not a "relationship", or if a "beam" has by chance inherited a "voltage" property, etc. Currently there is no instrumental support for automated checking; the diagrams are best checked visually. Thus it is recommended to involve the "Model Tester" role (desirably possessing the "black belt" qualification).

- Registration of new reference data items (classes, individuals, templates) in the project RDL. The peculiarities of this step depend on the type of RDL maintenance technology and of the warehouse chosen. Presently options for storage are: keeping MS Excel spreadsheets or OWL files at the company servers (without standardized access to the reference data); usage of iRING Sandbox software (supporting the standardized access to the reference data "facade" as RDF/OWL SPARQL endpoint). At this step the decision is made regarding the alternative national language naming for reference data items. If national language names are required for the particular engineering project, they are registered as reference data in the local project RDL.

- Promotion of new reference data items to external RDLs. If added reference data items are of interest not only in the particular project, but also at the industry, national or even international level - they should be shared. For example, templates could be added into the section "Proposed" at the Web-page https://www.posccaesar.org/wiki/SigMmt/Templates, so that after checking and approval procedures others can use it. Following the external RDL rules one can register new classes and probably even alternative national language names for reference data.

- Reference data verification. After reference data are registered in RDL it is possible in a number of cases to perform the automated verification, when special software checks for the correctness of the whole set of created reference data: if there are any contradictions in the type assignment, specializations, classifications and relationships between classes, individuals, relationships, templates and roles. Presently the "ready-to-use" verification software is absent; in the future it will possibly depend on the formats of chosen data warehouses. Our .15926 project has plans in the direction of the automated reference data verification.

4. Mapping. At this stage the correspondence between the ISO 15926 reference data items and the project warehouse data description items is created.

Usually the mapping is performed as the identification of the project data description items with the roles of specialized templates chosen or created in the reference data libraries at the previous stage of data characterization (common name for this activity is the "mapping of project data description items to template roles"). This is required to transform project data from the warehouse to the neutral ISO 15926 format. Transformed data may be then loaded into any other warehouse which has a mapping to the same reference data items.

"The magic of semantic technologies" of ISO 15926 is that in theory the mapping can be done to the reference data items (related classes and individuals) through different templates, not necessarily through the same set of templates with the same set of roles. If the project data from some data warehouse are obtained as instances of a template you had not used before in your mapping, you have a "semantic" solution. You have to access the reference data library containing the unknown template, get the template axiom from there, build the analytic diagram for it (if the diagram is not provided by that RDL), comprehend the meaning of the template from the diagram structure and role restrictions, and do the mapping of new template roles to your warehouse project data description items.

If an unknown template is found to be a specialization of other template, also unknown to you, you need to follow the whole chain of specializations and to obtain the axioms for all templates used to construct the template of interest. This operation is in essence the "expansion" of the template up to the level of Part 2 entities, described in Part 7 of the standard.

The work products of this stage are mapping tables, tailored for the adapters of warehouses used in the data transfer.

The mapping procedure depends on the software used – different software may use very different interface realizations for a "mapping editor" ("translator") and different structures of mapping tables. The project data warehouses in turn may exists as Excel electronic tables, relational databases, object databases of different structure – and all these cases can have their own pecularities.

Mapping creation is the utilization of the work product "reference data" created at the previous stage.

Mapping is performed for every warehouse participating in the common information space – for all sources and targets of project data. This is not a very difficult task (as the major work – creation of missed reference data – is finished at this moment), but it requires the knowledge of peculiarities of the project data representation in the warehouses and the knowledge of ISO 15926 reference data libraries used.

5. Project data transfer. Data transfer between project warehouses becomes possible when both warehouses have mapping of their data description items to the same set of reference data (in the simplest case – to the roles of the same set of specialized templates). Via the source warehouse mapping its adapter creates exchange files (or the set of messages in the case of on-line queries), and via the target warehouse mapping its adapter places data into the target warehouse in its native format.

Thus the precondition of a project data transfer is the presence of two adapters: for source and target warehouses.

Input work products for this stage are output work products of the previous stage – mappings (correspondence dictionaries recognized by adapters) for warehouses participating in the data transfer.

Output work products of this stage are exchange files containing project data in the neutral format of ISO 15926.

At this stage a new professional role is required – “Project Data Administrator”. Formally this stage is more related to the project data life-cycle. We have described it here as essentially linked to reference data engineering.

6. Verification of transfer results. At this stage the completeness and consistency of the data delivered to the target warehouse is checked. Verification may be performed manually or with the help of software specific for the target warehouse. Most probably verification will reveal that not all the desired project data were transferred from the source warehouses, because of the incomplete or wrong identification of project data description items at the stage 2.

Incremental nature of a reference data life-cycle. It is important to note that it is almost impossible to achieve data integration “in one pass” (i.e. to identify all the warehouses for a big project, to define all necessary data sets, to find all reference data items, etc.). Usually there will be several “passes”, each for a small portion of data depending on the current needs of engineers. That is why the life-cycle presented here is incremental (in ISO 19760 sense): data transfer capabilities are increasing with each successful pass.

Data Modeling Languages

Reference data engineering uses many formal languages of data modeling:

- “The Part 2 language” – these are 201 data types, defined in Part 2 of ISO 15926 in EXPRESS data description language (ISO 10303-11) and illustrated with EXPRESS diagram language. This list of data types has to be learned by heart for “fluent” use in reading and writing, but only by those creating the core and base templates and defining base classes (these data engineers are spoken about as “black belts”). Those who create specialized classes and define specialized templates need to know these 201 types well enough. These data engineers are spoken about as “yellow belts”.

- Template language – this is the language of Part 7 of the standard, and it is essentially a subset of the first-order logic language (FOL). The FOL is used for presentation of template axioms (see examples of templates at the pages referenced above). However it is possible to do template specialisation without firm knowledge of this language. The Part 7 diagram language is a “must” for reference data engineers, enabling the construction of analytic diagrams for the templates.

- RDF and OWL are to be known only by those who does not use software tools (e.g. iRING Tools or .15926 software) to form the mapping files and project data files, or who founds the quality of those software tools unsatisfactory.

Reference Data Engineer Role

The professional role of a reference data engineer requires the following qualifications and 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” and “yellow belt”. “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.

Training of a reference data engineer starting from a qualified employee with initial knowledge of mathematics and ISO 15926, takes usually three months of full employment under the qualified guidance of at least “yellow belt” - before the moment of getting the “yellow belt” level.

Required Tools

Necessary tools for reference data engineering and project data transfer are: reference data editor, mapping editor and adapter.

Reference data editor allows to search for reference data (classes and templates) in RDL federation and to register reference data in local RDL. This is the main tool of characterization. It works usually with a “sandbox” of local RDL, where the results of the work are collected. At the moment the only reference data editor is provided by iRING.

For other realizations of the standard (e.g. XMpLant) iRING editor is not suitable, and the editing have to be made with other tools, e.g. in XML editor.

We do not know about any cases of successful application of (any version of) Protégé ontology editor for a real project reference data. It is probably impossible without writing specific plug-ins. Even in this case it is doubtful, as Protégé does not work with external RDLs and does not allow loading of real RDLs in OWL format – because of their size.

Mapping editor defines mappings. Its inputs are RDL data and a data structure of a particular data warehouse.

Adapter is able to transfer data structure of a warehouse to a mapping editor, and with access to the mapping it can form project data files from the warehouse or to load data from those files into the warehouse.

Hence, adapters are unique for each data warehouse type. For example, SmartPlant Foundation will require one adapter, AVEVA NET Platform – an other, Bently ProjectWise – one more, and today all of these are only planned. iRING project (http://iringug.org) has started the development of the adapters for SmartPlant, with possible extension for products of other vendors (http://iringug.org/wiki/index.php?title=IRINGTools_Interfacing_Project).

Today iRING has the mapping editor and the "universal" adapter, which requires the copying of the information from the real warehouse to the temporary relational database (meaning additional simplification steps of the data warehouse structure) and provides mapping to that intermediate data structure.