Modelling Geometric Objects with ISO 15926: Three proposals with a comparative analysis

Geiza M. Hamazaki da Silva[1,2], **Bruno Lopes**[3], **Gabriel B. Monteiro Lopes**[2]

[1] Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Av. Pasteur, 296, Urca – Rio de Janeiro-RJ – Brazil

[2]Computer Graphics Technology Laboratory
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225, Gávea – Rio de Janeiro-RJ - Brazil

[3]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225, Gávea – Rio de Janeiro-RJ - Brazil

Abstract. *In the field of Oil & Gas, the ISO15926 proposes a standard for integration, sharing, exchange and delivery of data between computer systems based on the standardization of data formats and an ontology approach to represent common industry classes and relations. Due to the structure and the large number of terms defined at this standard, the complexity of creation information models is high. This aims to consolidate a methodology for modelling geometric objects following the structure of ISO 15926. We take into account the need for complete abstraction between geometry and business data. Three approaches are presented with a comparative analysis, which should reveal the appropriate practice to be adopted both in manual and in software supported ISO15926 compliant information modeling.*

1. Introduction

The National Institute of Standards and Technology (NIST) reported in 2004 that the costs generated by lack of proper interoperability between systems used in capital projects in the United States were around US$15.8 billion per year. This was in great part due to the absence of international standards for interoperability used in the projects that were analyzed by NIST. The use the information technologies to effectively integrate design, construction and business processes was not widely observed [Lee et al. 2012, Gallaher et al. 2004].

Aiming to promote interoperability for industrial automation systems for process plants, the ISO 15926 standard is designed to simplify the integration of data to support the life-cycle activities and processes of production facilities (as in Fig.1). The ISO 15926 standard is concerned with the storage of information, constructing knowledge bases for integration of life-cycle data for process plants including oil and gas production facilities [ISO 15926-1 2004]. These knowledge bases are modeled with structures in “First-Order Logic” and implemented based on an ontology approach to information, consistent with the W3C Web Ontology Language (OWL) [d’Aquin and Noy 2012, Consortium 2004].

The OWL standard is used to represent common industry terms that are mapped to the ontology with classes and relationships [Batres et al. 2007]. These terms are modeled

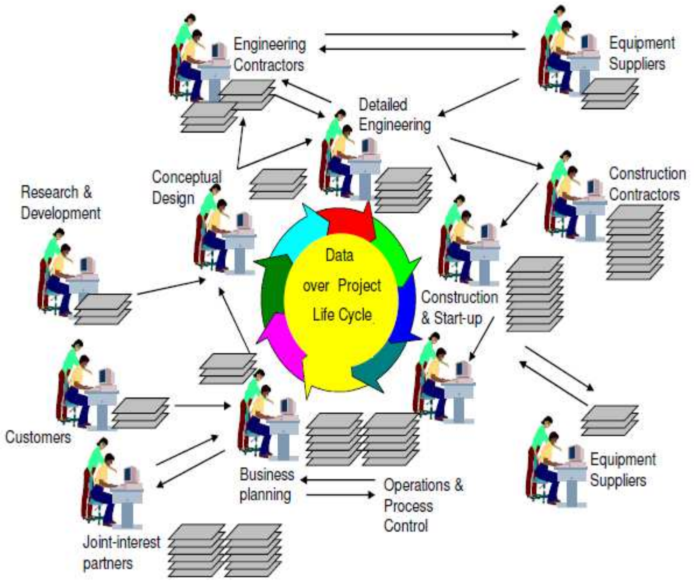

Figure 1. Data over Project Life Cycle – modified from [Pawsey 2010]

using the data model [ISO 15926-2 2003] and the initial reference data set (ISO 15926-4, 2007) which are shared databases or data warehouses used to describe industrial project lifecycle concepts. The ISO 15926 standard consists of several parts. Some of them are published, such as parts 1, 2, 3, 4, 7 and 8, while others are under development (see [IRING User Group 2012]).

Due to, among others, the high complexity of modelling concepts by the structure of the standard and the large amount of terms defined by its ontology, there is no consolidated methodology for information modelling in ISO 15926. It is essential for the usability of this standard that this complexity be hidden by the template use [ISO 15926-7 2011]. The most basic templates must be modelled using only entities from ISO 15926 - Part2 and ISO 15926 - Part4 as a requirement to be compatible with its conceptual model. Therefore, this basic model may be specialized to accommodate any field of engineering knowledge, as geometry, whose relevance is present in any Engineering schematic, 3D model, datasheets etc.

Engineers working on Capital projects use Computer-Aided Design and Drafting systems (CADD), which, for representing 3D and 2D schematics, ultimately use geometric objects (or primitives), such as: ellipses, polylines etc. Thus, to interoperate geometry related information, a standard is required for the structured data that describes the geometric objects. This is offered by the ISO 15926 Part3 [ISO 15926-3 2007], which defines the catalog of geometry and topology terms.

This work presents three approaches for modeling geometric objects following ISO 15926 and a comparative analysis among them. In the next session we present a brief introduction to this standard and after that we present our proposals. Then, we present the conclusions, work in progress and future work.

2. ISO 15926

The ISO 15926 standard (Industrial automation systems and integration, integration of life-cycle data for process plants including oil and gas production facilities [ISO 15926-1 2004]) consists of several parts. Some of them are published, like parts 1, 2, 3, 4, 7 and 8. At the time of this publication, parts 7 and 8 of ISO 15926 had been submitted to the ISO standard approval process, under TC184/SC 4. What follows is a brief introduction to the published ISO 15926 parts.

Part 1: Overview and fundamental principles [ISO 15926-1 2004] – Specifies a representation of information associated with engineering, construction and operation of process plants.
Part 2: Data Model [ISO 15926-2 2003] – Describes the entities used by the standard to represent the process plant life-cycle information. It is designed to be used in conjunction with reference data [ISO 15926-4 2007]: default instances that represent information common to users and process plants.
Part 3: Geometry and Topology [ISO 15926-3 2007] – Defines objects in the reference to data library for geometry and topology. It is based on ISO 10303 [ISO 10303-1 1994] and the dictionary of standard shapes are extracted from the ISO 10303-42 [ISO 10303-42 2003] and ISO 10303-104 [ISO 10303-104 2000].
Part 4: Reference Data Library [ISO 15926-4 2007] – Support for a specific life cycle depends on the use of appropriate reference data based on the data model [ISO 15926-2 2003].
Part 6: It defines a methodology for development and validation of reference data.
Part 7: Templates Implementation methods for the integration of distributed systems [ISO 15926-7 2011]. A template is seen as a data schema and the part 7 describes a catalog of templates and defines an implementation-independent template methodology for definition, verification, expansion of templates, as well as presenting an initial set of templates to allow the use of the conceptual model ISO 15926- Part2. It consists of the definition of the signature and axioms in first-order logic; verification and expansion are done with the software Template Expander.
Part 8: Implementation methods for the integration of distributed systems – OWL implementation. This part defines the specification for data exchange and lifecycle information integration using RDF and OWL to describe the templates of part7.

3. Modelling Geometric Objects

According to ISO15926, complex objects must be defined as templates, concepts that are defined using basic entities until they are reduced to basic terms (Proto and Core Templates). They must be compliant with ISO 15926 - Part2 and ISO 15926 - Part4, ensuring the integration of data portability and interoperability.

To define a best practice of how to represent geometry and topology of the manufactured and geological objects of an industrial process in ISO 15926, the ISO 15926 Part3 was created [ISO 15926-3 2007]. It presents a huge library of basic terms and definitions to be used for modeling.

According to ISO 15926, the geometric objects and properties must be modelled using Templates (ISO 15926-Part7). They are defined by decompositions of terms into simpler ones, in finite steps, until they are reduced to basic (or primitive) geometric terms. These basic terms (Core Templates) must be ISO 15926-Part2 compliant [Silva and Lopes 2011].

We present three approaches of modeling geometric objects, regarding a circle (geometric entity) as an example of how to use them.

3.1. Identification of ISO 15926-Part3 Elements

In the modeling process, it is important to understand the requirements of the object that will be modelled (stage 1). At this moment we will identify the object (e.g. circle) properties according with the ISO 15926-Part3 [ISO 15926-3 2007]. All the entities definitions present in this work were extracted from the ISO 15926-Part3. Any term in boldface represents a term in the ISO 15926 ontology. Circle definition:

> *An object is a **circle** if and only if: 1-it is **curve**; 2-it lies in a **plane**; 3- there is a centre point that is equi-distant from each point in the curve. NOTE 2 A **circle** has the geometric properties: radius; center and plane. These properties can be given for a **circle** by a **axial_reference_placement** and a **radius**. A **circle** has two alternative values for the **axial_reference_placement** corresponding to opposite directions for the normal.*

According with the definition, the concept **circle** is subclass of the concept **curve** and it is defined by a **radius**, a central point and a plane. So, the properties of a **circle** can be defined by the concepts **radius** and **axial_reference_placement**.

> *An object is a **radius** if and only if:1-it is a function between geometric objects with a unique **radius** and **metric_space_length**; 2- it specifies the radius. An object is an **axial_reference_placement** if and only if: 1-it is a function between geometric objects with a unique axial placement and **axis1_placement** (which is a **metric_space_point** and a direction denoted z); 2- it specifies the position and orientation of the geometric object.*

The concept **radius** is defined by a **metric_space_length**, that stores the measure of the radius. So the concept **radius** is used to link the measure with an object that has a **radius**, at this case the **circle**.

By the definition, the concept **axial_reference_placement** is used to connect a plane with an object. This plane is defined by the concept **axis1_placement**, that is composed by a set of points (**metric_space_point** and one direction (**direction**). Then the concept **axial_reference_placement** will connect the **axis1_placement** and the **circle**.

3.2. Identification of the Necessary Templates in Part7

After the requirements are known, it is necessary to analyze the data and the relationships that will be used in the modeling process (stage 2) [Kim et al. 2011]. In ISO 15926, the first step is to look for the concepts and relationships (templates) that will be used to model the object, ensuring that they are defined either at the Reference Data Library (RDL) or the Template Library (TPL) [Association 2008]. If it does not exist, it is necessary to ask to PCA [POSC Caesar Association 2012b] or its Special Interest Groups [POSC Caesar Association 2012a] to add it to the databases.

During the circle's modelling process, it was observed that some concepts were not connected with each other. By the Fig. 2, only the **circle** is connected with the class

curve, because **circle** is subclass of **curve**. In its definition, the concept radius is part of a circle, but it is not a **circle** (analogous to **axial_reference_placement**), so it is necessary to compound this relation. The compositions of these relations will be done with the construction of templates, whose methodology is described by the document ISO 15926 Part7 [ISO 15926-7 2011]. The templates hide the internal complexity of the models (described by the axioms), since access is given by the elements present in the signature.

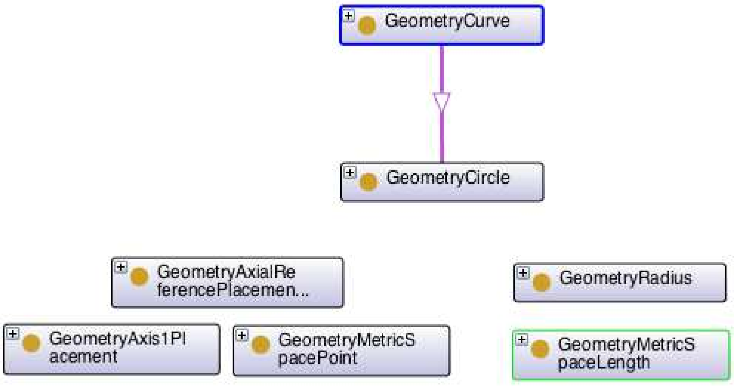

Figure 2. Identified Classes at Part3

The modeling process of a template has two steps:

1. Definition of the signature, that describes the elements that compound the relationship;
2. Definition of Axioms/Sentences in First Order Logic (FOL), that describes the semantics through the relations between the elements presented in the signature.

The axioms will be used to verify the consistency of the template. This verification is done by a tool called Template Expander that expands the axiom until the description with concepts defined in ISO 15926-Part2 or ISO 15926-Part4 [ISO 15926-7 2011].

The specification of a template axiom in FOL is done with the *if and only if* logical connective, where the signature of the template is on the left side, and the sequences of formula connected with the conjunction connective are on the right side.

Example: The template **RealMagnitudeOfProperty** is used to connect a concept classified as a **property** with a numeric value and a **scale** (as in Table 1 and the following axiom).

Table 1. Signature of RealMagnitudeOfProperty

Order	Rule	Type
1	hasProperty	Property
2	valPropertyValue	ExpressReal
3	hasScale	Scale

```
RealMagnitudeOfProperty(x1, x2, x3) <->
    property(x1) & ExpressReal(x2) & scale(x3) &
    exists u (MagnitudeOfProperty(x1, u, x3) & IdentificationByNumber(x2, u) ) .
```

What follows is a research about the template modeling process, regarding a circle as an example.The first approach presents a simplified modeling process. As some properties of the model are hidden due to its simple construction, it is necessary to understand the full model to infer these properties by queries. In the second approach, the model has more properties explicit and therefore the modeling process is more difficult it is possible to access the properties with simpler queries (it is not needed to know the full model to infer the properties in queries). The third approach proposes an intermediate abstraction between the first and the second one.

Alternative 1: Easy to Model but Difficult to Query. As defined by the ISO15926-Part3, **axial_reference_placement** and **radius** are functions that connect concepts, so they are candidates for templates.

In this alternative, the model of the template has a low granularity, it hides some possible templates without compromising the model structure. We constructed three templates: **RadiusTemplate**, **MetricSpacePointTemplate** and **DirectionTemplate**. The first template will connect one object that has a radius with a value that describes the length of radius. Its signature is shown at Table 5

Table 2. Signature of RadiusTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithRadius
2	hasRadius	RealNumber
3	hasLowerBound	RealNumber
4	hasUpperBound	RealNumber

The parameters of the template signature above are of an object that has a radius (**circle**), the radius value and the lower and upper bounds of a scale.

```
RadiusTemplate(x1, x2, x3, x4) <->
    ObjectWithRadius(x1) & exists m radius(m) & hasEnd1(m,x1) & hasEnd2(m,k)
    exists k metric_space_length(k) & exists j scale(j) &
    exists l ( PropertyRange(l) & LowerUpperMagnitudeOfPropertyRange(l, j, x3, x4) &
        RealMagnitudeOfProperty(k, x2, j) ).
```

The concept **metric_space_length** alone does not represent the numeric value of a radius, it defines just a measure. A relationship between this measure and the circle is done by the template **RadiusTemplate** that relates this measure with a scale.

Some of the templates that are necessary to the modeling process can be found at the ISO 15926-Part7. In the template proposed above, the templates **RealMagnitudeOfProperty** (see Table 5) and the **LowerUpperMagnitudeOfPropertyRange** are at ISO 15926-Part 7.

According to ISO 15926-Part3, **metric_space_length** is subclass of **property**. Thus, it satisfies the condition of the template **RealMagnitudeOfProperty**. The template **RealMagnitudeOfProperty** claims an **scale** object, defined by the ISO 15926-Part2 . The scale is used to define a range of allowed values. To model a scale the template **LowerUpperMagnitudeOfPropertyRange** is necessary to connect two values with a scale, that is connected with a numeric value and a **metric_space_length**.

The template **MetricSpacePointTemplate** will connect an object with a **metric_space_point** with three real values (Table 3) that defines a plane according to ISO 15926-Part3.

Table 3. Signature of MetricSpacePointTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithAxialReferencePlacement
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

```
MetricSpacePointTemplate(x1, x2, x3, x4) <->
    ObjectWithMetricSpacePoint(x1) &
    exists c(CoordinateSystem(x1, c) & ListOfReals3Template(c, x2, x3, x4)).
```

The template CoordinateSystem, presented at ISO15926-Part7, specifies a plane with its three coordinates, that are connected with the template **ListOfReals3Template**.

The template **DirectionTemplate** connect three real values to represent the direction of the object (see Table 4).

Table 4. Signature of DirectionTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithDirection
2	hasDirectionX	RealNumber
3	hasDirectionY	RealNumber
4	hasDirectionZ	RealNumber

```
DirectionTemplate (x1, x2, x3, x4) <->
    ObjectWithDirection(x1) &
    exists c ( CoordinateSystem(x1, c) & ListOfReals3Template(c, x1, x2, x3) ).
```

Bellow is presented a graphic example of templates instantiations (**RadiusTemplate**, **MetricSpacePointTemplate**, **DirectionTemplate**) to construct the circle with a radius which the value is 3, with the position(1,2,3) and the direction expressed by the coordinate (1,0,0). It uses the following diagram language. The example is in Fig. 4. All the following Figures follows the legend in Fig. 3.

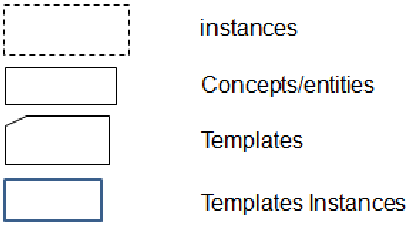

Figure 3. Legend of diagrams

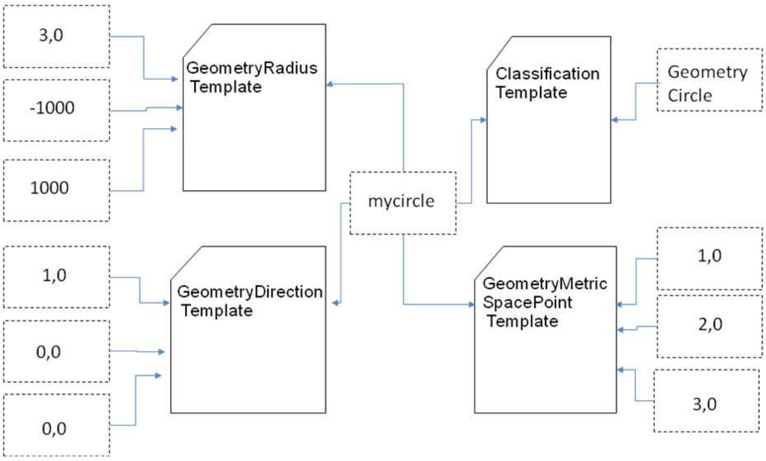

Figure 4. Instantiation example of alternative 1

Alternative 2: Hard to Model but Easy to Query. In this alternative, the granularity is high. The model is defined with five templates. As the first alternative, all the properties that define a **circle** are modelled. This process is more complex, but once it is modelled and instantiated, the queries about any properties will be done with ease.

The template **RadiusTemplate** (Table 5) is used to join all the properties about a **radius**. Its signature is the same as that of alternative 1, but the axiom that describes the template model is more detailed.

Table 5. Signature of RadiusTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithRadius
2	hasRadius	RealNumber
3	hasLowerBound	RealNumber
4	hasUpperBound	RealNumber

```
RadiusTemplate(x, y, z, w) <->
    ObjectWithRadius(x) & RealNumber(y) & RealNumber(z) &
    RealNumber(w) & exists m ( radius(m) hasEnd1(m,x1) &hasEnd2(m,k) ) &
    exists k (metric_space_length(k) &
        exists  j (Scale(j) & exists l (PropertyRange(l) &
                LowerUpperMagnitudeOfPropertyRange(l, j, z, w) &
                RealMagnitudeOfProperty(k, y, j))) &
        exists p (MappingTriple(m, x, k) & radius(p)) ) .
```

In the formula above, the template **MappingTriple** [ISO 15926-7 2011] joins the object with **radius** to its properties.

The template **AxialReferencePlacementTemplate** (Table 6) defines the circle's plane. It relates six real values: three that define the **ReferencePoint** and three others that defines the **Direction**.

```
AxialReferencePlacementTemplate(q,px,py, pz, dx, dy, dz) <->
    ObjectWithAxialReferencePlacement(q) & exists k(axis1_placement(k) &
        ReferencePointTemplate(k, px, py, pz) &
        ReferenceDirectionTemplate(k, dx, dy, dz) &
        exists p(MappingTriple(p, q, k) & axial_reference_placement(p))).
```

Table 6. Signature of AxialReferencePlacementTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithAxialReferencePlacement
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber
5	hasDirectionX	RealNumber
6	hasDirectionY	RealNumber
7	hasDirectionZ	RealNumber

The template **AxialReferencePlacementTemplate** uses the **ReferencePointTemplate** (Table 7) and **ReferenceDirectionTemplate** (Table 8). These templates define the reference point and the direction respectively.

Table 7. Signature of ReferencePointTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithReferencePoint
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

Table 8. Signature of ReferenceDirectionTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithReferenceDirection
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

```
ReferencePointTemplate(x, px, py, pz) <->
    ObjectWithReferencePoint(x) & exists k ( metric_space_point(k) &
        exists c(CoordinateSystem(k,c) & ListOfReals3Template(c,px,py,pz)) &
        exists p (MappingTriple(p, x, k) & reference_point(p)) ).

ReferenceDirectionTemplate(x, dx, dy, dz) <->
    ObjectWithReferenceDirection(x) &
    DirectionScaleTemplate(x, dx, dy, dz) .
```

The templates **CoordinateSystem** and **ListOfReals3Template** have the same semantics of the templates presented at the alternative 1. The template **ReferenceDirectionTemplate** uses the template **DirectionScaleTemplate** (Table 9), that connect the three real values using the template **ListOfReals3Template** which has the same semantics presented at the alternative 1.

```
DirectionScaleTemplate(x, dx, dy, dz) <->
    ObjectWithDirection(x) & exists k ( direction(k) &
        exists c(CoordinateSystem(k,c) & ListOfReals3Template(c,dx,dy,dz)) &
        exists p ( MappingTriple(p, x, k) & direction_scale(p)) ).
```

Table 9. Signature of DirectionScaleTemplate

Order	Rule	Type
1	hasPossessor	ObjectWithDirection
2	hasPositionX	RealNumber
3	hasPositionY	RealNumber
4	hasPositionZ	RealNumber

Bellow is presented a graphic example of templates instantiations (Fig. 5), that uses the same diagram language and values of the circle instantiation proposed at the alternative 1: a circle with a radius which the value is 3, with the position(1,2,3) and the direction expressed by the coordinate (1,0,0).

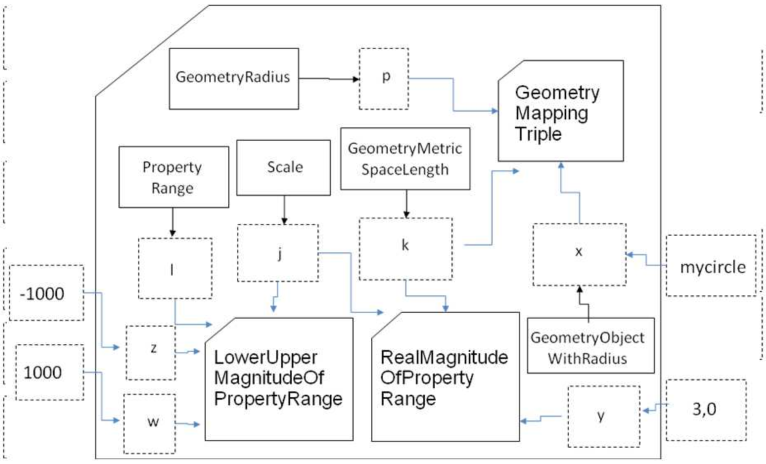

Figure 5. Instantiation example of alternative 2

Alternative 3: Trying a Balanced Approach. This approach presents two alternatives of intermediate granularity using a high level template. It aims to propose a model whose properties are explicit, to simplify the process of constructing queries in alternative 1 and simplifying the model construction in alternative 2, using all of its elements. In our first attempt, called **CircleTemplateAlternative3_1**, a higher level of abstraction is presented to construct instances of the model in the alternative 1. It uses all of its elements with a new template that constructs all the necessary instances, as a procedure that encapsulates all the process. This template signature is present in Table 10. The **CircleTemplateAlternative3_2** presents a more compact process of individuals construction which does not affect query construction negatively. Its template signature is shown in Table 10. The signature is the same for both alternatives in this approach, differing in its axioms, as follows.

```
CircleTemplateAlternative3_1(q,rd,rl,ru,px,py,pz,dx,dy,dz)<->
    circle(q) & RadiusTemplate(q, rd, rl, ru) &
    MetricSpacePointTemplate(q, px, py, pz) &
    DirectionTemplate(q, dx, dy, dz).
CircleTemplateAlternative3_2(q,rd,rl,ru,px,py,pz,dx,dy,dz)<->
    circle(q) & RadiusTemplate(q, rd, rl, ru) &
    AxialReferencePlacementTemplate(q,px,py,pz,dx,dy,dz).
```

Table 10. Signature of CircleTemplateAlternative3_1 and CircleTemplateAlternative3_2

Order	Rule	Type
1	hasPossessor	Circle
2	hasRadius	metric_space_length
3	hasLowerBound	RealNumber
4	hasUpperBound	RealNumber
5	hasPositionX	RealNumber
6	hasPositionY	RealNumber
7	hasPositionZ	RealNumber
8	hasDirectionX	RealNumber
9	hasDirectionY	RealNumber
10	hasDirectionZ	RealNumber

4. Conclusions

The effort in the development and application of the ISO 15926 standard contributed with a new paradigm of information management for the Oil e Gas industry, that will reduce the costs in this area [Gallaher et al. 2004]. For the development of computer systems that are compliant with the standard across the industry, it shall know how to define, to manage, to extend the information models to store the data in a neutral format. There are many documents about the ISO15926 standard, but is difficult to organize the knowledge and to understand how to model the concepts without a methodology. It creates barriers for the deployment of the standard. Collaborating on this challenge, this work presents three alternatives that can be adopted at the modelling process. The two first alternatives have different levels of information granularity and one should be adopted depending of the queries to the Endpoints to retrieve the information. The last alternative uses high level templates to encapsulate the process of linking the elements at the instantiation of the others templates.

In future works, the main objective is to develop the standard researching subjects as the implementation of tools to help domain experts use the ISO 15926 standard, i.e. software to model and verify ISO 15926 templates, as well as an environment to create and to manage distributed data bases built upon the ISO 15926 proposed paradigm, building on the accumulated experience of the iRING User Group etc.; Implementation of the models using Web Ontology Language using the ISO 15926-Part8, envolving studies correlated with present day ontology challenges such as: how to store the ontology, how to manage the RDF triple store, how to make an efficient query across distributed RDF databases on the web; Design of an architecture to support format neutral exchange of 2D and 3D documents, based on SPARQL Endpoints providing federated management of process plant item symbology and Engineering document templates.

Acknowledgements

The authors thank TecGraf/PUC-Rio Computer Graphics Technology Laboratory and CNPq, for supporting this work, the PCA Geometry Special Interest Group and most especially Mr. Onno Paap for his advice and encouragement.

References

Association, P. C. (2008). RDS/WIP. http://rdl.rdlfacade.org. Acessed: February 2012.

Batres, R., West, M., Leal, D., Price, D., and Naka, Y. (2007). An upper ontology based on ISO 15926. *Computers & Chemical Engineering*, 31(5-6):519–534.

Consortium, T. W. W. W. (2004). Web ontology language overview. http://www.w3.org/2004/OWL/. Acessed: April 2012.

d'Aquin, M. and Noy, N. F. (2012). Where to publish and find ontologies? a survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11:96–111.

Gallaher, M. P., O'Connor, A. C., Jr., J. L. D., and Gilday, L. T. (2004). Cost analysis of inadequate interoperability in the US capital facilities industry. Technical report, National Institute of Standards and Technology.

IRING User Group (2012). IRING user group. http://iringug.org/wiki/index.php?title=Main_Page. Acessed: February 2012.

ISO 10303-1 (1994). Overview and fundamental principles.

ISO 10303-104 (2000). Integrated application resource: Finite element analysis.

ISO 10303-42 (2003). Integrated generic resource: Geometric and topological representation.

ISO 15926-1 (2004). Overview and fundamental principles.

ISO 15926-2 (2003). Data model.

ISO 15926-3 (2007). Reference data class.

ISO 15926-4 (2007). Initial reference data.

ISO 15926-7 (2011). Implementation methods for the integration of distributed systems: Template methodology.

Kim, B. C., Teijgeler, H., Mun, D., and Han, S. (2011). Integration of distributed plant lifecycle data using iso 15926 and web services. *Annals of Nuclear Energy*, 38(11):2309–2318.

Lee, S., Han, S., and Mun, D. (2012). Integrated management of facility, process, and output: data model perspective. *Science China-Information Sciences*, 55(5).

Pawsey, N. (2010). Iso 15926 & interoperability. In *PCA Meeting*.

POSC Caesar Association (2012a). PCA geometry special interest group. https://www.posccaesar.org/wiki/SigGeometry. Acessed: April 2012.

POSC Caesar Association (2012b). POSC caesar trac. https://www.posccaesar.org. Accessed: April 2012.

Silva, G. M. H. and Lopes, G. B. M. (2011). An approach about the modelling process to geometric objects with the ISO 15926 standard. In *Annals of CIB-W102 – Information and Knowledge Management*.