Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2011
August 28-31, 2011, Washington, DC, USA

DETC2011-47965

BUSINESS OBJECT MODELS FOR INDUSTRIAL DATA STANDARDS

Keith A. Hunten, P.E.
Advanced Development Programs
Lockheed Martin Aeronautics Company
Fort Worth, Texas 76101
Email: keith.a.hunten@lmco.com

Allison Barnard Feeney
Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland 20899
Email: abf@nist.gov

ABSTRACT

Business object models, as proposed for inclusion in the ISO 10303 family of industrial data standards developed in ISO TC 184/SC4 (SC4), are a layer over the ISO 10303 architecture that is intended to simplify and make the complex standards more accessible to a wider audience, ease implementation, and improve implementation performance. This paper discusses the motivation for developing business object models in SC4, proposes a process for developing business objects, provides example business objects at different levels of complexity, and describes issues facing the two SC4 projects currently developing business object models.

Keywords. *business object, data exchange, ISO 10303, STandard for the Exchange of Product data (STEP), product lifecycle management (PLM), web services, Product Life Cycle Support (PLCS)*

INTRODUCTION

ISO 10303, most commonly known as the Standard for Exchange of Product model data (STEP), is an international standard designed to exchange digital information, enabling an ever-widening range of engineering software systems to interoperate. Each software system has its own format for storing and writing data, making it nearly impossible for organizations using different systems to communicate product model data without translation. STEP, developed by a global consortium of standards bodies, governments, and industry, provides a robust neutral file format that has the potential to save $928 million (2001$) per year by reducing interoperability problems in the automotive, aerospace, and shipbuilding industries alone [1].

Business object models, as proposed for inclusion in the STEP family of industrial data standards developed in ISO TC 184/SC4 (SC4) [2], are layered over a complex data model and are intended to present that complex model in a form that is more understandable by application experts and easier to implement. As the STEP business object models are a new layer in the SC4 architecture, they provide an opportunity to use more mainstream methods for implementation.

Business object models may be specified in languages such as EXPRESS [3], the information modeling language used for STEP standards, or in more widely used languages such as the Unified Modeling Language (UML) [4] or eXtensible Markup Language Schema (XML Schema) [5]. Business objects may aggregate lower level objects from the primary requirements model to hide data complexity or may be augmented with additional data to complete the high-level concepts that the business objects are to be based on.

In this paper we discuss the motivation for developing business object models in SC4, propose a process for developing the business objects that compose the business object models, provide some examples of business objects at different levels of complexity, and describe issues facing the two SC4 projects currently developing business object models.

MOTIVATION FOR STEP BUSINESS OBJECT MODELS

There are several motivators to include business object models in STEP standards. One motivator is that the business object model is a domain-specific model that is documented in the vernacular most familiar to the target audience. Including such a model makes STEP standards more accessible to readers

who are typically not familiar with the complex formal data models in the STEP family of standards.

A second motivator is that business object models provide a separate requirements model that is designed to take advantage of mainstream information technology, such as web services. The model may be specified in widely used languages and leverage advanced implementation technologies, broadening the implementation base for STEP standards and reducing the cost of STEP implementation.

Another motivator is to increase ease of implementation over the traditional implementation forms of complex STEP data standards by providing Application Programming Interfaces (APIs) of reduced complexity. Furthermore, the business object model internal structure can be developed to ensure high levels of performance by careful design and optimization based on the use cases that the business object model is required to support. It may be that different use cases drive different business object models that all support the same API.

Finally, business object models will preserve and enhance the integration of STEP standards.

STEP Architecture Considerations

The architecture of STEP is designed to support the development of standards for product data exchange and product data sharing [6]. The architecture is governed by the following concepts: the scope of what is standardized and what is conformance tested is set at the level of "an application," application requirements are based on a model of a business activity, application requirements are standardized using an EXPRESS Application Reference Model (ARM) in domain terminology, and a mapping defines how the ARM requirements are satisfied using an EXPRESS Application Interpreted Model (AIM) that is specified using generic data concepts shared across STEP standards. The parts of the STEP family of standards that are designed for implementation are called Application Protocols (APs) [7].

The STEP document architecture was modularized to address some deficiencies of the original architecture and to speed development, facilitate implementation, and increase interoperability of applications of STEP [8]. One of the side effects of the STEP modular architecture is that the ARMs were generalized for reuse and they no longer provide the rich semantics of the application domain in the end user's vernacular. The generalization also resulted in fragmentation of the higher-level ARM constructs, further exacerbating the problem. This loss of semantic richness was recognized as a deficiency of the modular approach at the time of its development, and the ability to specify an optional layer of business terminology over the modular ARM was included in the architecture. However, no project has yet exercised that capability. Including a business object model in the AP enriches the document and makes it more accessible to users unfamiliar with the complex STEP data models.

Implementation Ease

The complexity of the STEP AIMs requires considerable study and experience to properly implement. This restricts the implementation of STEP standards to a small community of expert implementers that can only be afforded by large enterprises. This greatly reduces the potential implementation base of the STEP standards.

Two standards that are based upon STEP standards, the Object Management Group's (OMG) Product Lifecycle Management Services (PLM Services) [9] and Organization for the Advancement of Structured Information Standards (OASIS) Product Life Cycle Support (PLCS) [10], have evolved approaches to simplify implementation primarily by mapping the EXPRESS-based STEP standards to UML and then providing a path to widely utilized implementation methods. The PLM Services used ISO 10303-28 [11] to automate the implementation in an XML Schema format from EXPRESS. The OASIS PLCS standard defines templates to accomplish simplification (aggregation of lower-level concepts into higher level concepts) and also maps EXPRESS to UML and XML Schema. Thus these two approaches both provide a path from EXPRESS-based STEP standards to alternate implementation forms that utilize the widely used UML and XML Schema standards.

In addition to the benefits of UML, XML Schema, and templates shown by the PLCS and PLM Services methodologies, business object models for STEP must intentionally be simpler than the traditional implementation methods of STEP. One approach for simplification considered in this paper is to partially flatten the complex graph in an ARM by replacing an aggregation of multiple objects used to represent related information with one business object. It is anticipated that enhancement in implementation ease will accrue from automations of instantiation of the full graph of instances from the simplified representation in a business object. This approach is partially used in the layered PLCS templates. These efficiencies will apply to traditional STEP EXPRESS-based implementations along with those in UML and XML Schema.

As a STEP business object model is a full information model, and as it is developed and maintained by SC4 as an integral part of an AP, it will allow an AP project to more easily support multiple disparate implementation methods. Currently only clear text, database and XML Schema EXPRESS-based implementation methods are specified within STEP standards. These implementation methods are applied to the AIM of the AP. Business object models are intended to further these methods with implementation paths to automate the use of technologies such as web services.

This becomes more important as STEP standards evolve toward a future architecture as defined in the SC4 document, *Industrial Data Integrated Ontologies and Models (IDIOM) architecture specification* [12]. In this document a new architecture is presented as a way to move forward from EXPRESS-defined data models to those more widespread

technologies such as UML and the Web Ontology Language (OWL) [13]. It is thought that the adoption of more modern and widely implemented data modeling and implementation technologies that will provide a broader implementation base for STEP standards.

Integration Across STEP APs

The business object models are designed in such a way as to preserve the integration of the STEP standards in areas of commonality across the APs.

For example, the team developing ISO 10303-242, *Managed Model Based 3D Engineering* (AP242), is specifying business objects that map the similar, but not identical, product structure AP242 ARM objects to OMG PLM Services objects. Because the PLM-oriented ARM objects in AP242 are identical to PLM-oriented ARM objects in ISO 10303-239, *Product life cycle support* (AP239) [14], there is an opportunity for traceable mappings between the OASIS PLCS implementation of the PLM-oriented objects of its underlying STEP standard AP239, and the PLM-oriented objects implemented in the OMG PLM Services.

BUSINESS OBJECT MODEL DEVELOPMENT

The development process for the business object model uses an approach that leverages application specific vernacular most familiar to the users and implementers. Currently, two SC4 projects are developing business object models for inclusion in STEP standards: the second edition of ISO 10303-209, *Multidisciplinary Analysis and Design* (AP209), and AP242. The projects have different requirements for developing business object models. AP209 has the requirement of simplifying the complex EXPRESS data model in the engineering analysis domain into an API based on the business object model in order to provide greater access to potential implementers. AP242 has the requirement of maintaining compatibility with the OMG PLM Enablers services based on ISO 10303-214, *Core data for automotive mechanical design processes* (AP214) [15] a precursor to AP242. Both projects also have the requirement to provide a more user-oriented model to use as a teaching tool for explaining the standard to users.

Documentation of Business Object Models

The creation of a business object model begins with analysis of the ARM concepts. When the desired business object complexity warrants simplification, an aggregation of the ARM concepts is specified. Three different options for documentation to automate implementation are described below.

EXPRESS Model of the Business Object. This option provides some implementation automation when used with an implementation form such as XML Schema with a specified ISO 10303-28 configuration option.

An API Signature for a Business Object. This option provides a higher level of functionality than EXPRESS plus XML Schema in that methods are provided. This option addresses issues related to interface development based on the AIM. The information in the API signature is directly instantiated in the AIM. The signature is built on the AIM concepts that correspond to the ARM concepts that are related to the business object, along with any further processing requirements in a format suitable for an API in pseudo code.

A UML Model of the Business Object. This option provides a higher level of functionality than EXPRESS plus XML Schema in that methods are provided. This option provides implementation automation under the assumption that the associated XML Metadata Interchange (XMI) [16] would be used to automatically generate implementation forms such as web services.

Developing Business Object Models from ARMs

Two processes for developing business object models from existing ARMs utilizing different levels of abstraction of ARM objects have been identified as beneficial. A third class of objects, such as those for geometric representations of surfaces and B-rep solids, is being examined to determine the potential benefit of abstraction.

Examples illustrating the first two processes are provided below. The examples are intended to provide a first order look at the relative complexities and content, and are not intended for examination for correctness and completeness of content.

Defining Business Objects by Simple Re-Mapping

The first level of abstraction can be illustrated with the area of Item identification that forms the root of the product structure in STEP. This approach is much like what is being discussed by the AP242 project where the AP214 and PLM services concept of Item is mapped to the ISO 10303-203, *Configuration controlled 3D design of mechanical parts and assemblies* (AP203) [17] and the harmonized PDM Schema [18] concept of Product providing a simple mapping that expresses the object in a vernacular more familiar to the end user.

The following example of business object documentation illustrates the definition of an Item Identification business object. For each business object, the ARM concepts, the aggregation of the ARM objects, the AIM objects and processing requirements, the API signature definition, the graphical EXPRESS model and the UML model will appear in the documentation. The example text and figures that follow, outlined to set them apart from the text of this paper, appear how they are proposed for inclusion in STEP standards.

ARM Concepts
Item identification is the root object of a product structure that identifies a product or of a type of product. It is a collector of data common to all revisions of the product.

Aggregation of ARM Objects
There is only one ARM object associated with the Item business object:
Product.

Associated AIM Objects and Processing Requirements
There is only one AIM object associated with the Item business object:
product.

Item Business Object API Signature Definitions
The Public Function specified here is a 'constructor'. The function signature arguments are first named for ARM Objects, and then AIM objects are added as necessary. In this case no further AIM objects are required for completeness. See the module ISO 10303-1017 Product_identification for definitions.

Public Function for Item
id (string)
name (string)
description (OPTIONAL string)

Private Function for Item
There are no private functions necessary.

Item Business Object EXPRESS-G Definition
Figure 1 illustrates the EXPRESS-G of the Item Business object.

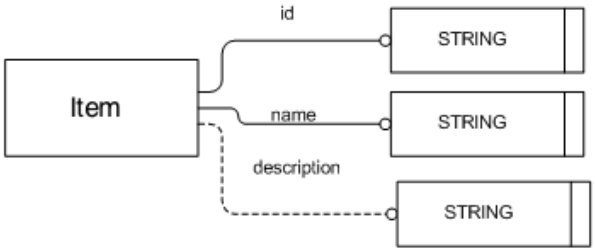

Figure 1. ITEM IDENTIFICATION EXPRESS-G

Item Business Object UML Definition
Figure 2 illustrates the equivalent UML of the Item business object.

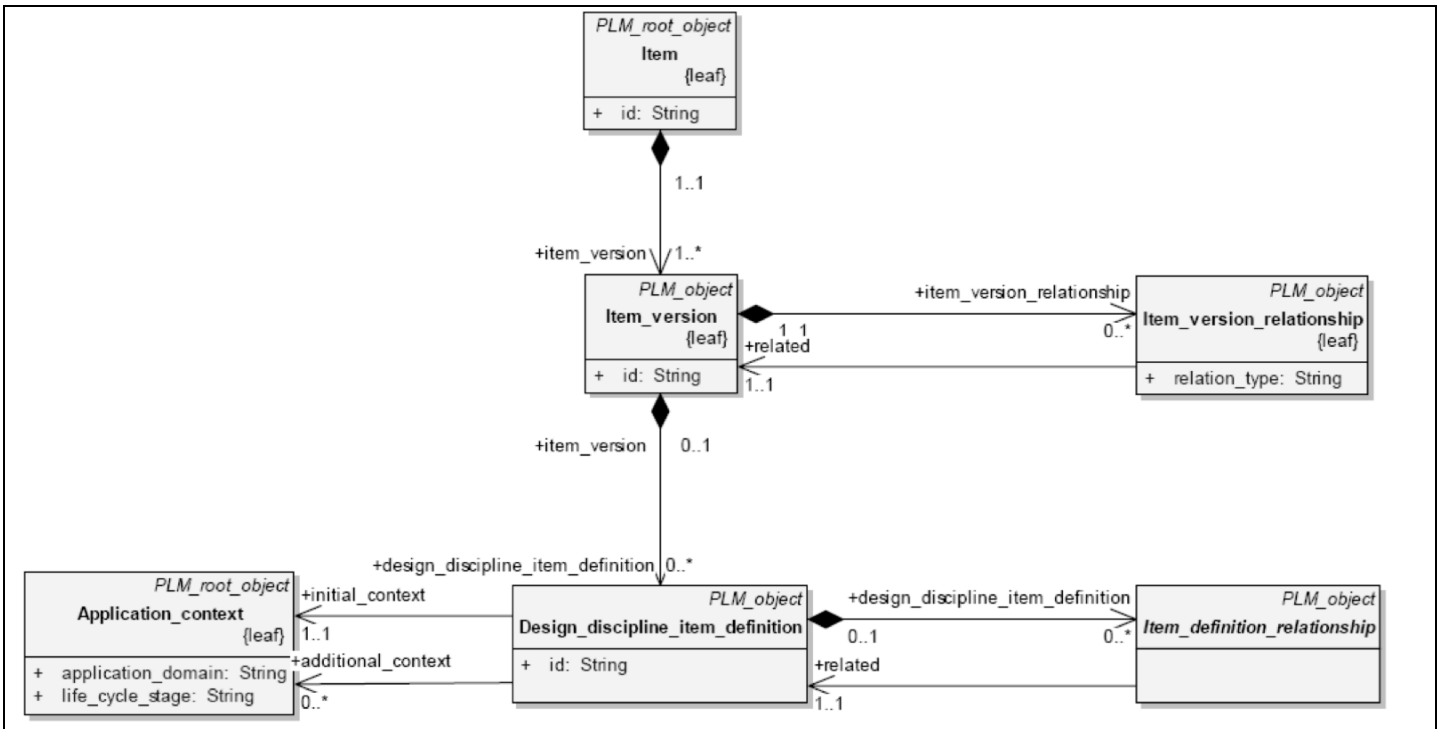

Figure 2. PRODUCT IDENTIFICATION UML FROM THE OMG PLM SERVICES

Defining Business Objects through Simplification

An approach for simplification is to partially flatten the complex graph in the standard by replacing an aggregation of multiple objects used to represent related information with one business object. It is anticipated that implementation performance enhancements will accrue from automating instantiation of the full graph of instances from the simplified representation in a business object.

The second level of abstraction can be illustrated by the node information in the area of finite element analysis based the second edition of ISO 10303-209. In ISO 10303-209 a node is a position in space that finite elements are connected to, and that has several attributes and references. The Node business object provides a higher level of complexity as it is intended to aggregate and automate some of the related ARM and AIM concepts such as coordinate frames and geometric founding. Automation requirements are illustrated through the private methods specified for the Node business object that functions to instantiate the full AIM graph associated with a fea_node_representation from the partially flattened Node business object signature. The simplification process is significantly more complex than the re-mapping approach. The following example (necessarily more detailed and complex) of business object documentation specifies a Node business object.

ARM Concepts

Nodes are collected into a Node_shape based upon the members of a Node_shape having a common coordinate system.

Nodal location coordinate systems may be the base asserted coordinate system, or some coordinate system related to it.

Nodes may have two other coordinate systems or directions (vectors) associated with them: one for surface normals used in shell element formulations (a vector, part of an axis2_placement_3d, is rare), and the other for results calculation and output (a coordinate frame, is quite common).

The ARM objects include:

Node;
Detailed_geometric_model_element;
Point_model;
Node_shape;

Node_shape_relationship;
Nodal_results_coordinate_system;
Substructure_node_relationship;
Node_description;
Fea_group;
Fea_model.

Aggregation of ARM Objects
To simplify the signature of an API function to read or write Nodes and related nodal information, the following aggregations apply:

The Node, a reference to a Detailed_geometric_model_element specifying the location of the Node, and the Nodal_results_-coordinate_system shall be aggregated into one function. In addition, there would be an additional optional Detailed_-geometric_model_element in that function that would specify an element direction (vector).

The Node_shape and Point_model objects shall be aggregated into one private function that would be used each time a Node is created. These objects are a part of the Node business object.

The following shall each have their own separate function as no aggregation is necesssary:
Node_shape_relationship;
Substructure_node_relationship;
Node_description;
Fea_group.

Associated AIM Objects and Processing Requirements
The final set of arguments for the Functions will be an aggregation of the attributes and references in the set of ARM objects and their attributes, and then AIM long form entities as needed, grouped under each function.

Note: The API write functions will instantiate AIM objects according to the arguments of the API signature. Most are direct writes; however there will be some aggregation of instances necessary for objects that contain set references.

The Node Function
node;
node_with_vector;
node_with_solution_coordinate_system;
dummy_node;
geometric_node;
fea_axis2_placement_3d;
direction_node;
point_representation;
node_set;
fea_model_3d.

The Node_geometric_relationship Function
analysis_item_within_representation;
node_geometric_relationship.

The Node_description Function
node_definition.

The Node_group Function
node_group.

The Substructure_node_relationship Function
substructure_node_relationship.

Node Business Object Signature Definitions
The Public Function specified here is a 'constructor'. The function signature arguments are first named for ARM objects, and then AIM objects are added as necessary.

Public Function for Node
See 10303-1383 Finite_elements and ISO 10303-104 Finite element analysis for definitions.
node_identification (supply an ID. Can be alpha-numeric, but preferably a unique integer)
location (supply an x,y,z, plus flag for <CARTESIAN, SPHERICAL, CYLINDRICAL>)
results_coordinate_space (-> fea_axis2_placement_3d instance)
shell normal (OPTIONAL -> fea_axis2_placement_3d or fea_axis2_placement_2d instance)
model_ref (-> fea_model instance)
description (OPTIONAL supply a description)
dummy (flag true/false if node is a subtype 'dummy_node')
geometric (flag true/false if node is a subtype 'geometric_node')

Private Function for Geometric Founding
The following pseudo-code describes the operations required for the Geometric_founding private method:
Create a point of <CARTESIAN, SPHERICAL, CYLINDRICAL>)
Check to see if node is in a node_set
If not, create a new node_set
Create a new point_representation whose .items points at the new node set
If true, add to existing node_set
Only if in the same coordinate frame (specified by application input format)
Else create new node_set

Private Function for Results Coordinate Frame
The following pseudo-code describes the operations required for the Results_coordinate_frame private method:
Check to see if the results coordinate frame exits
If not then create a fea_axis2_placement_3d instance
Set a pointer to the fea_axis2_placement_3d instance

Private Function for Shell Normal
The following pseudo-code describes the operations required for the Shell_normal private method:
If this OPTIONAL argument is provided, the create a node_with_vector instance
Check to see if the shell normal coordinate frame exists
If not then create a fea_axis2_placement_3d instance
Set a pointer to the fea_axis2_placement_3d instance

Private Function for FEA Model Instance Generation
The following pseudo-code describes the operations required for the FEA_model_instance_generation private method:
Check to see if a fea_model exists for the current set of application input data
If not, then create a fea_model_3d instance
Information for mandatory text attributes to be provided by using application
fea_axis2_placement_3d offset location to be specified by querying the application input
Set a pointer to the fea_model instance

Private Function for Node Description Generation
The following pseudo-code describes the operations required for the Node_description_instance_generation private method:
Check to see if a node_definition exits that matches the string
If not then create a node_definition instance and the associated entities to link it to a node_representation
Set a pointer to the node_definition instance

Node Business Object EXPRESS-G Definition
Figure 3 illustrates the equivalent EXPRESS of the Node business object.

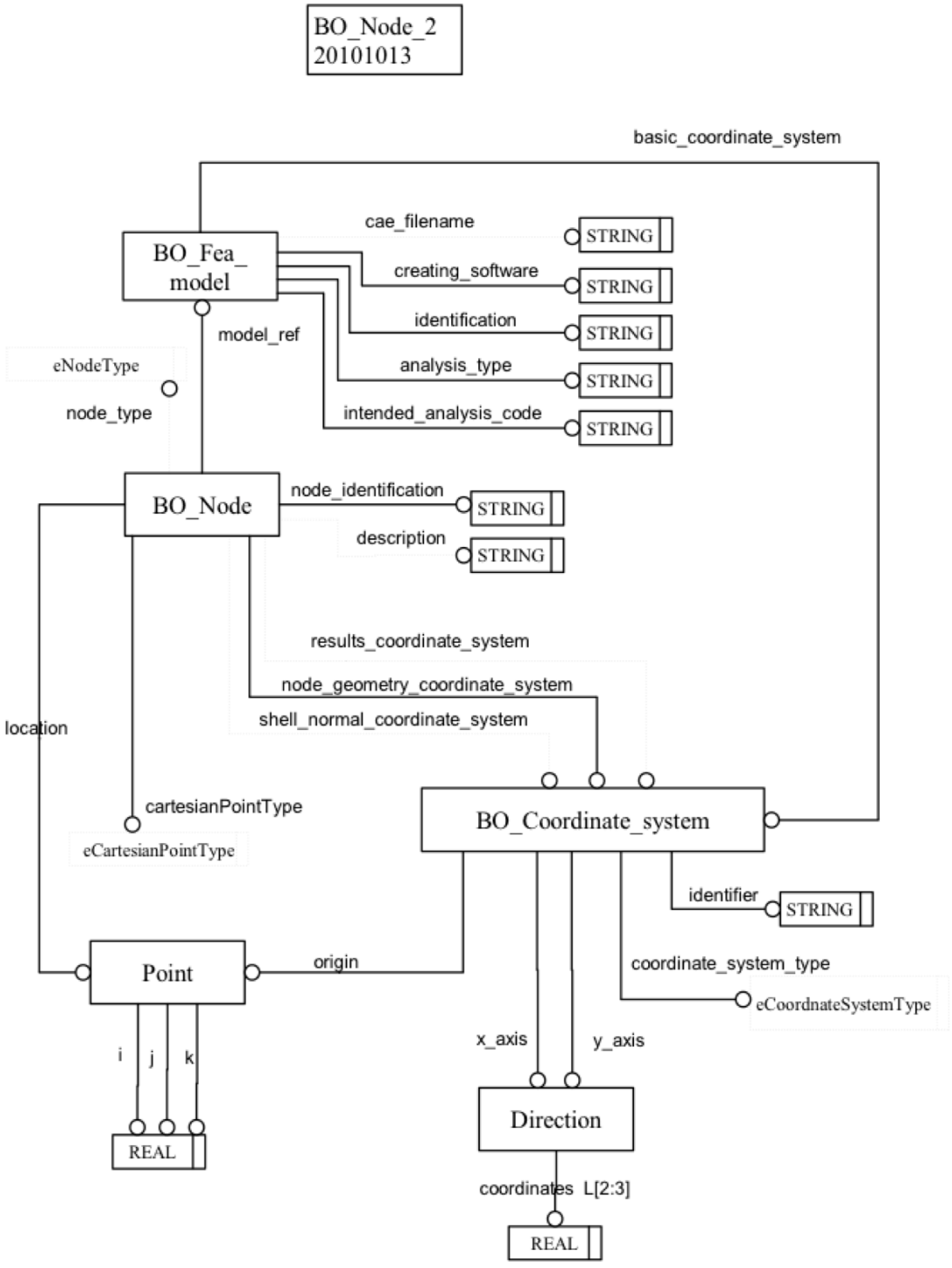

Figure 3. NODE BUSINESS OBJECT EXPRESS-G BASED ON 10303-209

Node Business Object UML Definition

Figure 4 illustrates the equivalent UML of the Node business object

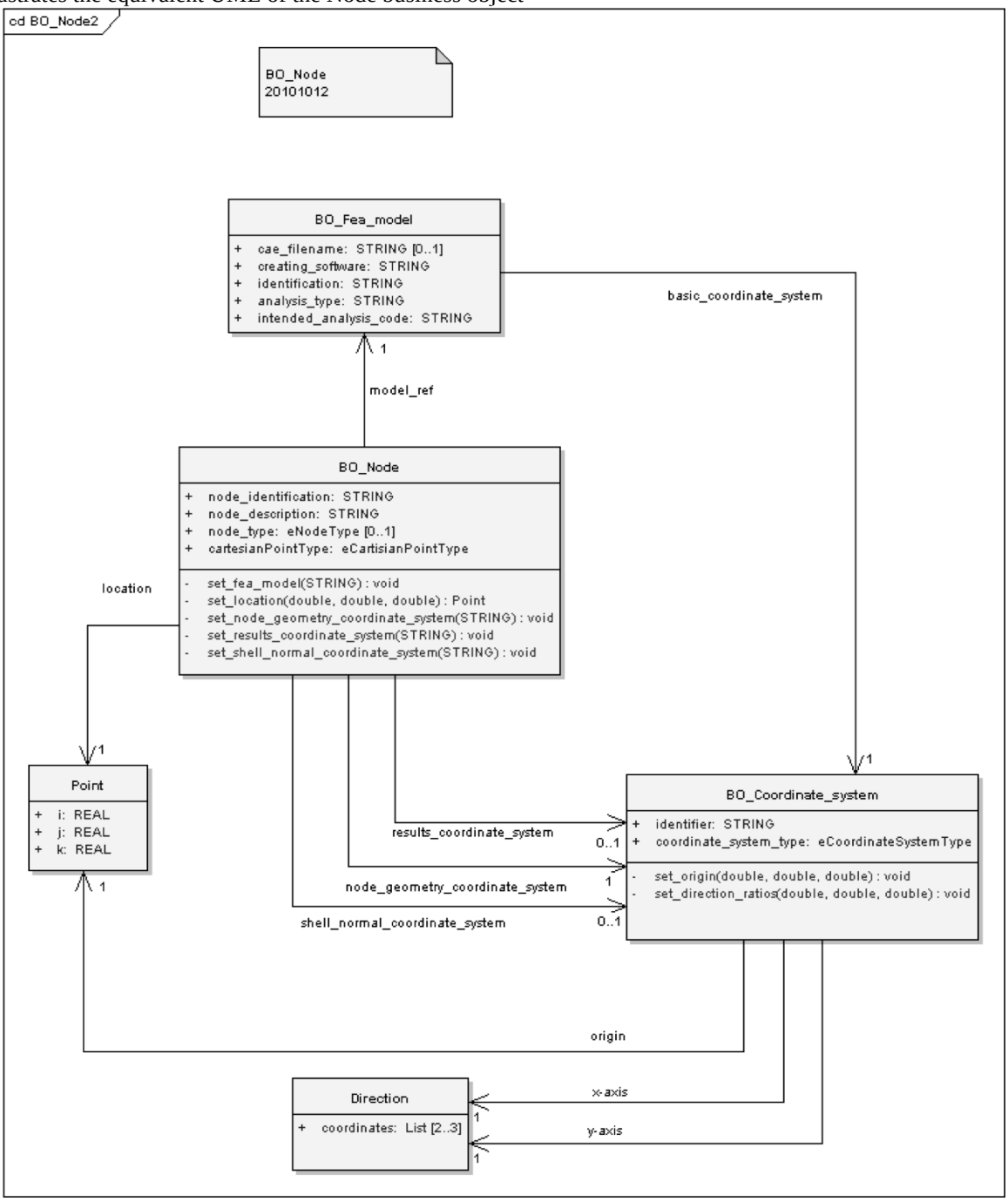

Figure 4. NODE BUSINESS OBJECT UML BASED ON 10303-209

When Business Objects Provide Little or No Benefit

A third class of objects, such as those for geometric representations of surfaces and B-rep solids, is complex and specific enough that simplification is not practical. Therefore a business object would look virtually identical, removing any potential benefits. In STEP standards, most, if not all, of the geometric representation objects fall into this category.

ISSUES

Several open issues have been identified with the development of business objects.

For the EXPRESS plus XML Schema implementation form there is no way to specify methods to automate the implementation of the AIM graph from the flattened and concatenated business object definition. This is fine if it is a simple mapping, i.e., that the business object attributes are the same type and number as the ARM object it is based on.

The creation and query types of business objects will be substantially different, and business objects for queries should be based on use cases. The proposal described in this paper so far only deals with the creation type of business object.

Additional work needs to be done to examine whether there is a requirement for the API approach to instantiate ARM objects instead of, or in addition to, AIM objects.

CONCLUSION

This paper presents an approach for creating business object models for STEP standards. The motivations and benefits are discussed. The two principle drivers for creating business object models are recasting STEP standards to be more understandable by application experts, and easing implementation through use of mainstream technologies as opposed to SC4-specific languages and tools. Examples that illustrate two approaches to creating business objects show that indeed the two strategies of simple mapping and simplification postulated in this paper can result in business objects with beneficial characteristics. It was also shown that there are times where there is little or no benefit to business objects. Finally, several issues are identified that need to be addressed before completing the methodology for developing and implementing STEP business object models.

REFERENCES

[1] Gallagher, M., O'Connor, A., Phelps, T., 2002. *Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries.* RTI Project Number 07007.016.

[2] SC4 *On-Line*. On the WWW, at http://ng.tc184-sc4.org. February 2011.

[3] ISO 10303-11:2004, Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual.

[4] *Unified Modeling Language (UML) 2.3,* Object Management Group. OMG Available Specification. May 2010. See also URL http://www.omg.org/technology/ documents/modeling_spec_catalog.htm#UML

[5] Fallside, David C., ed., 2001, *XML Schema Part 0: Primer*, W3C. See http://www.w3.org/ TR/2004/REC-xmlschema-0-20041028/ primer.html.

[6] ISO 10303-1:1994, Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles.

[7] Kemmerer, S., ed., 1999, *STEP the Grand Experience*, NIST Special Publication 939, US Government Printing Office, Washington, DC.

[8] Barnard Feeney, A., 2002, "The STEP Modular Architecture," ASME J. Comput. Inf. Sci. Eng., **2**(2), pp.132-135.

[8] *OMG Product Lifecycle Management Services, v2.0*, See also URL http://www.omg.org/ spec/PLM/Current.

[10] *OASIS Product Life Cycle Support (PLCS) TC*. See also URL http://www.oasisopen.org/committees/tc_home. php?wg_abbrev=plcs. February 2011.

[11] ISO 10303-28:2007, Industrial automation systems and integration -- Product data representation and exchange -- Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas.

[12] ISO TC 184/SC4 N2615, 2010, *Industrial Data Integrated Ontologies and Models (IDIOM) architecture specification*. See also URL http://ng.tc184-sc4.org/index.cfm?PID=80& FID=62321

[13] Motki, B., Patel-Schneider, P., Parsia, B., eds., 2009. *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax.* W3C Recommendation. See also URL http://www.w3.org/TR/owl2-syntax/.

[14] ISO 10303-239:2005, Industrial automation systems and integration -- Product data representation and exchange -- Part 239: Application protocol: Product life cycle support.

[15] ISO 10303-214:2010, Industrial automation systems and integration -- Product data representation and exchange -- Part 214: Application protocol: Core data for automotive mechanical design processes.

[16] *XML Metadata Interchange 2.1.1*. Object Management Group. OMG Available Specification. December 2007. See also URL http://www.omg.org/technology/ documents/modeling_spec_catalog.htm#XMI

[17] ISO 10303-203:2011, Industrial automation systems and integration -- Product data representation and exchange -- Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies.

[18] Ungerer, M., Rosche, P. eds., 2002. *Usage Guide for the STEP PDM Schema V1.2 (Release 4.3)*. See also URL http://cax-if.de/documents/pdmug_ release4_3.pdf.