

VIRGINIA POLYTECHNIC INSTITUTE & STATE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

THE BUILDING OF INFORMATION MODELS IN STEP

Vinícius Medina Kern, Jan Helge Bøhn, and Ricardo Miranda Barcia

Article submitted to the Second International Congress of Industrial Engineering, Piracicaba, SP, Brazil, October 7-10, 1996.

Contact:

Vinícius Medina Kern
vmkern@cme.nist.gov

Copyright © 1996 Virginia Tech

THE BUILDING OF INFORMATION MODELS IN STEP

Vinícius Medina Kern

PPGEP-UFSC and NIST

vmkern@cme.nist.gov

Jan Helge Bøhn

Virginia Tech, Department of Mechanical Engineering

Blacksburg, VA 24061-0238, USA

Ricardo Miranda Barcia

Programa de Pós-Graduação em Engenharia de Produção

UFSC-CTC-EPS - Caixa Postal 476 - 88040-900 - Florianópolis-SC, Brasil

Abstract

STEP, or ISO 10303, is a unique and innovative solution to the problem of product data representation and communication. This paper focuses on information modeling in STEP. The data description language EXPRESS is presented, along with a discussion on the modeling of information rather than just data. The process for the construction of Integrated Resources and Application Protocols, STEP's information models, is outlined.

Keywords

information modeling, STEP, EXPRESS, Integrated Resource (IR), Application Protocol (AP), Product Data Exchange (PDE), conceptual schema

1. Introduction

The Standard for the Exchange of Product model data (STEP, ISO 10303) is a unique and innovative solution to a very complex problem: the representation and communication of product data between different CAx systems. STEP is divided into several documents organized in classes that conform to a database architecture.

The use of STEP depends on two main kinds of documents: application-specific information models and the implementation methods used in STEP implementations. This paper focuses on the development of STEP information models. STEP's data description language, EXPRESS, is presented, along with a discussion on the modeling of information rather than just data. The generic, non-implementable information models, the Integrated Resources (IRs), are presented. The building of application-specific information models, the Application Protocols (APs), is described. The integration of Application Protocols using Application Interpreted Constructs is outlined, and the relationship of STEP information models to the rest of the standard is summarized.

2. STEP background

STEP (STandard for the Exchange of Product model data, ISO 10303) was preceded by other product data exchange (PDE) specifications. The most widely known are Autodesk's DXF and the standards IGES (United States), SET (France), and VDA/FS (Germany). These specifications focus mostly on the exchange of geometric data. STEP is an international effort that goes beyond geometry and that aims to represent product data throughout a product's life cycle. Laurance observes that

“a central tenant in the STEP credo is that IGES is deficient in that it does not contain the semantics of the information, only the data. Hence, the job of STEP is to capture the semantics. STEP does this by focusing on the use of the information in the product design process: that is, on the context in which the information is to be used.”
[1]

Data vs. information

The terms *information modeling* and *data modeling* are used interchangeably at times, and their difference may be subtle. Wilson [2] defines information and data modeling as:

- Information model is “an implementation independent specification of the entities defining individual pieces of information necessary for some enterprise, the relationships among the entities, and the constraints on and between the entities and relationships.”
- Data model is “an implementation dependent specification of the entities defining the data for some enterprise, the relationships among the entities and the constraints between the entities.”

STEP development architecture

From the beginning of STEP's development it was acknowledged that the standard was going to be very large. STEP is being published as several *Parts*, grouped in classes of documents according to the ANSI/SPARC database framework [3]. The ANSI-SPARC three-level (*external, conceptual, internal*) architecture is paralleled in STEP, respectively, by:

- Application layer: Information models specific to an application area, developed by experts. These are the Application Protocols (APs), being the 200 part series, and the recently introduced Application Interpreted Constructs (AICs), being the 500 part series.
- Logical layer: A library of product information models called Integrated Resources (IRs), being the 40 and 100 part series, dedicated to describe all domains of interest in a unique, unambiguous way.
- Physical layer: A series of implementation methods, being the 20 part series, dealing with the mapping of the application schemas onto a specific computer technology.

In addition to these documents, there are other classes of STEP documents dedicated to introduction, description methods, and the testing of implementations. This article focuses on information modeling, which comprises the logical and application layers, and the EXPRESS language which is a description method.

EXPRESS is:	EXPRESS is not:
a Data Description Language (DDL)	a Data Manipulation Language (DML)
technology independent	a methodology
object-flavored (but not restricted to object-oriented systems)	a programming language
human-readable, computer-processable (but non-executable)	

Table 1 - EXPRESS characteristics

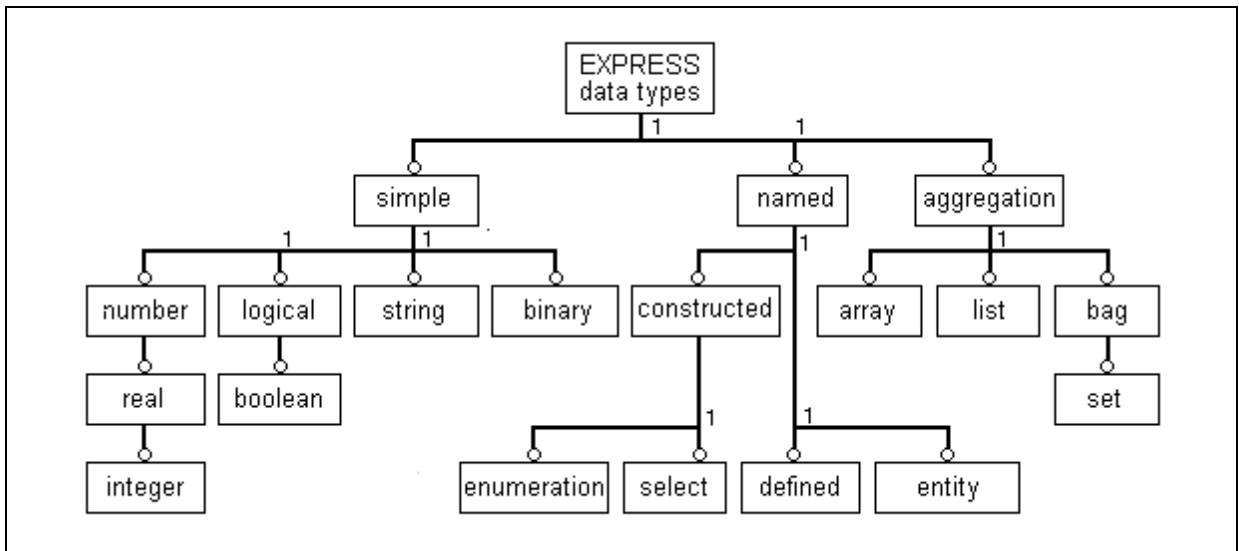


Figure 1 - Classification of EXPRESS data types

```

ENTITY path
  SUPERTYPE OF (ONEOF(open_path, edge_loop, oriented_path))
  SUBTYPE OF (topological_representation_item);
  edge_list : LIST [1:?] OF UNIQUE oriented_edge;
WHERE
  WR1: path_head_to_tail(SELf);
END_ENTITY;

```

Figure 2 - Example of an entity in an IR

EXPRESS

EXPRESS, or STEP Part 11 [4], is a Pascal-like, declarative, object-flavored information modeling language built in accordance with Wirth's syntax [5]. It was designed to be STEP's modeling language because no other language seemed to be appropriate to represent the richness of product data models. Table 1 outlines EXPRESS characteristics.

EXPRESS has a rich data type system which Kiekenbeck et al. [6] classify as generalized, simple, aggregated, constructed, and named. Figure 1 presents a classification of EXPRESS data types in EXPRESS-G, the graphic version of EXPRESS (generalized data type is not shown).

The concept of entity type in EXPRESS is similar to that in an entity-relationship model, and to a class in an object-oriented model. The inheritance mechanism has similarities with that of object-oriented programming (OOP), and multiple inheritance is allowed.

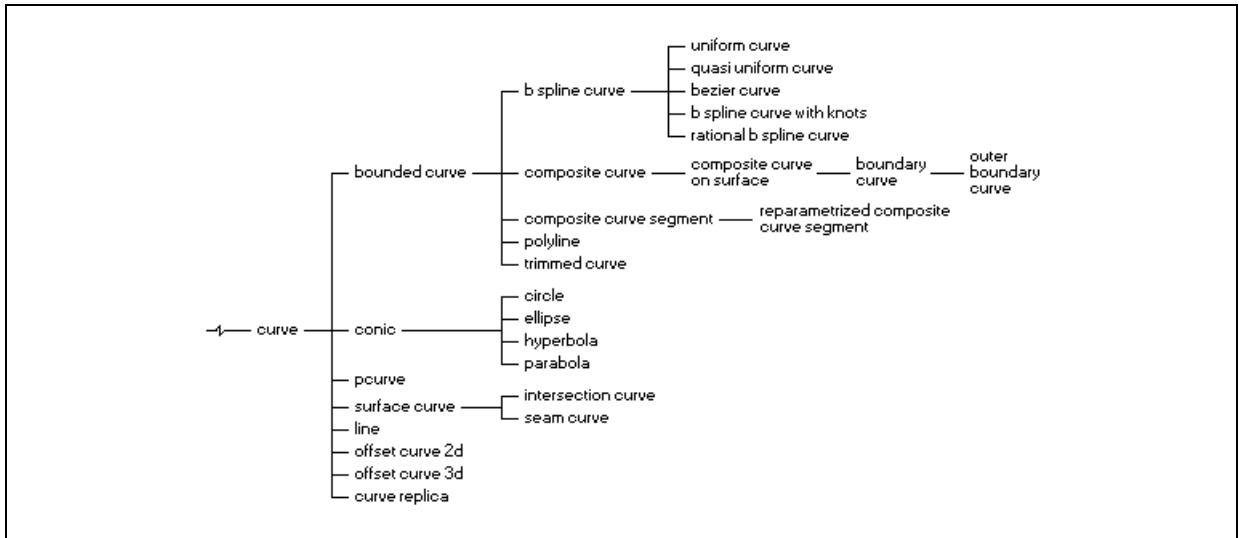


Figure 3 - Some geometry entities in Part 42

3. Information models in STEP

Information models in STEP are built using a bottom-up approach. According to Yang [7], STEP makes use of abstraction as the principle integration mechanism. This results in a hierarchical conceptual schema that permits data constructs to be “re-used” in different contexts.

STEP product data models are normalized, which gives them a fine level of granularity. Hardwick et al. consider that

“Normalizing the object models in STEP is necessary but expensive--necessary because manufacturing data needs to be shared by as many applications as possible, and expensive because normalization often makes a model harder to process.” [8]

Integrated Resources

IRs, the first class of information models in STEP, are divided into *Generic* (40 part series, general purpose) and *Application* (100 part series, concerning a range of applications). The collection of all IRs is a set of reusable, interconnected, unambiguous schemas. A single concept is represented only once within the IRs.

Figure 2 shows an example of a general-purpose geometric entity present in the IR *Geometric and Topological Representation, Part 42* [9]. Here the entity `path` inherits properties from the entity `topological_representation_item`, and it is a supertype for three other kinds of specialized `paths`. It has a list of edges as attribute, and the clause `WHERE` specifies a rule that makes the end vertex of each edge equal to the initial vertex of the next edge.

Part 42 has three main sections--geometry, topology, and geometric models. EXPRESS modeling power allows for the representation of information beyond pure data structures:

“Although many modeling languages can describe the data structures needed to represent topology, EXPRESS is one of the few languages that describes the rules needed to make sure that a topology data set is correct.” [10]

```

SCHEMA associative draughting;
USE FROM aic_advanced_brep;           -- ISO 10303-514
USE FROM aic_draughting_annotation;  -- ISO 10303-504
(...)
USE FROM date_time_schema            -- ISO 10303-41
    (date_and_time);
USE FROM geometric_model_schema      -- ISO 10303-42
    (edge_based_wireframe_model,
     geometric_curve_set,
     shell_based_wireframe_model);
(...)
ENTITY draughting_pre_defined_colour
    SUBTYPE OF (pre_defined_colour);
WHERE
    WR1: SELF.name IN ['black', 'red', 'green', 'blue',
                       'yellow', 'magenta', 'cyan', 'white'];
END_ENTITY;
(...)
END_SCHEMA;

```

Figure 4 - AIM short listing in AP 202 [13]

Figure 3 presents a hierarchical view of part of the geometry section in Part 42: the entity *curve* and its specializations.

IRs serve as the basis for the building of Application Protocols, the second class of information models in STEP.

Application Protocols

There are more than 20 STEP APs currently under development by ISO. The largest document will probably be AP 214 [11], which is a very broad specification dedicated to automobile industry applications.

APs are the most important and, by far, the largest class of STEP parts [12]. They are the STEP information models that are intended to be implemented. APs define all the information required for a specific application domain, and they are ideally independent of each other [2].

The concept of APs was developed to avoid arbitrary implementations of only parts of the standard, as it is the case with IGES. Without APs, CAx vendors would be free to implement translators comprising non-standardized subsets of STEP.

To build an AP, the structures in the IRs are reused and customized to fit the needs of a specific application domain. The building of an AP is comprised of four stages:

- Define scope
- Specify information requirements
- Generate the AIM (Application Interpreted Model)
- Establish conformance requirements

```

ENTITY date_and_time;
  date_component : date;
  time_component : local_time;
END_ENTITY; -- date_and_time

```

Figure 5 - Entity `date_and_time` in the extended AIM schema in AP 202 [13]

Application elements	AIM element	Source	Rules	Reference path
2D_CARTESIAN_COORDINATE_SPACE	[geometric_representation_context]	42	6, 16	
	[global_unit_assigned_context]	41		
2D_ELEMENTARY_GEOMETRIC_CURVE_SET	elementary_2d_geometric_curve_set_shape_representation	202	41	elementary_2d_geometric_curve_set_shape_representation <= shape_representation
2D_GEOMETRIC_CURVE_SET	geometrically_bounded_2d_wireframe_representation	503	41	geometrically_bounded_2d_wireframe_representation <= shape_representation

Table 2 - Mapping table for AIM in AP 202 (detail) [13]

In the first stage, the scope of the model is defined in a document called the Application Activity Model (AAM). It provides an initial evaluation of the standard.

In the second stage, the processes identified in the AAM have their information requirements defined in a document called the Application Reference Model (ARM). The ARM specifies the context-specific information to be communicated via the AP. The IRs are not considered at this stage.

The third stage consists of mapping the IR and ARM constructs to produce the Application Interpreted Model (AIM). An AIM is an interpreted subset of the general-purpose IRs for use in the specific application context of the AP. The AIM includes a short form of an EXPRESS schema. It references the constructs of the IRs that are used, the schema in which each construct is defined, and additional constraints. Figure 4 presents an excerpt from the short listing in AP 202 [13]. Figure 5 illustrates how the entity `date_and_time`, referenced from IR 41 in figure 4, is represented in the AIM extended schema. A mapping table shows how each requirement in the ARM is satisfied by EXPRESS constructs in the AIM. Table 2 displays an example from the mapping table in AP 202: the `Application element` column shows application object names; `Source` is the number of the corresponding STEP part; `Rules` refers to numbered rules that apply to the current AIM element or reference path; and `Reference path` “documents the role of an AIM element relative to the AIM element in the row succeeding it” [13], allowing for the specification of a reference path through several related AIM elements. The connector ‘<=’ in table 2 can be read as “is subtype of.”

In the fourth stage, tests based on the ARM and AIM are developed in order to verify the conformance to constraints of the instances in the AIM.

The information model generated by the above procedure is the normalized conceptual model in the AIM. It may therefore be tested by a conformance testing tool. AIMS tend to be very large documents and often overlap. This creates a need for AP integration which will be discussed next.

Application Interpreted Constructs

An AIC is a specification for a narrow domain area shared by two or more APs. It provides semantic integration for common functional requirements. For instance, geometric representation (B-rep, surface models, etc.) may be used in several different application domains such as automotive design, sheet metal tooling, and shipbuilding [12]. An AIC can be developed and used to fulfill this functional requirement, as observed by Fowler:

“The existence of an AIC identifies the *potential* for reuse of implementation code, and for the sharing of data between applications. This latter point is particularly important: since the data specification for surface models is common across the three applications identified above, data instances may be shared between them.” [12]

4. Conclusions

In this paper, we discussed the construction of information models in STEP. The development of the concept of APs, the testing methodology, and the design and use of a powerful modeling language, EXPRESS, are the best improvements of the standard in relation to previous specifications. Rather than being limited to plain data, EXPRESS allows for the representation of product information, including data, constraints, rules, functions and procedures to be applied on data. The APs define an interpretation of the IRs for a specific application domain. Coupled with standardized testing procedures, they favor sound implementations of STEP translators.

The normalized, technology-independent STEP information models are conceptual specifications that can be used by implementors to build applications that share data regardless of database approach, network or operating system.

References

- [1] Neal Laurance, “A High-Level View of STEP,” *Manufacturing Review*, 7, 1994, pp. 39-46.
- [2] Peter R. Wilson, “A View of STEP,” In: *Geometric Modeling for Product Realization*, P.Wilson, M.Wozny, and M.Pratt (editors), Elsevier, 1993, pp. 267-296.
- [3] Dennis Tsichritzis and Anthony Klug (editors), “The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems,” *Information Systems* 3, 1978, pp. 173-191.
- [4] ISO TC184/SC4/WG5 Document N65(P2), Product Data Representation and Exchange Part 11, *EXPRESS Language Reference Manual*, ISO International Standard, November 1, 1994.
- [5] Niklaus Wirth, “What Can We Do about the Unnecessary Diversity of Notation for Syntactic Definitions?,” *Communications of the ACM* 20 (11), 1977, pp. 822-823.
- [6] Juergen Kiekenbeck, Annette Siegenthaler and Gunter Schlageter, “EXPRESS to C++: A Mapping of the Type-System,” In: *EXPRESS User's Group (EUG'95) Conference*, 1995.
- [7] Yuhwei Yang, “The STEP integration information architecture,” In: *Engineering Data Management: Key to Success in a Global Market*, Proceedings of the 1993 ASME International Computers in Engineering Conference and Exposition, San Diego CA, 1993, pp. 39-47.

- [8] Martin Hardwick, David Spooner, Tom Rando and K.C. Morris, "Sharing Manufacturing Information in Virtual Enterprises," *Communications of the ACM* 39 (2), February 1996, pp. 46-54.
- [9] ISO TC184/SC4/WG3 Document N125a, Product Data Representation and Exchange Part 42, *Integrated Resources: Geometric and Topological Representation*, ISO proposed Committee Draft, July 31, 1992.
- [10] STEP Tools, Inc., "How to select an Express compiler," *STEP Tools News* 1 (1), July 1992.
- [11] ISO TC184/SC4/WG3 Document N436, Product Data Representation and Exchange Part 214, *Application Protocol: Core Data for Automotive Mechanical Design Process*, ISO Committee Draft, August 8, 1995.
- [12] Julian Fowler, *STEP for Data Management, Exchange and Sharing*, Technology Appraisals Ltd., Twickenham UK, 1995, 214 pp.
- [13] ISO TC184/SC4, Product Data Representation and Exchange Part 202, *Application Protocol: Associative Draughting*, ISO Draft International Standard, August 8, 1995.