NISTIR 4528

STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic & Syntactic Rules

William F. Danner

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Building and Fire Research Laboratory
Gaithersburg, MD 20899

David T. Sanford

The Boeing Company

Yuhwei Yang

Product Data Integration Technology

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

NISTIR 4528

STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic & Syntactic Rules

William F. Danner

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Building and Fire Research Laboratory
Gaithersburg, MD 20899

David T. Sanford

The Boeing Company

Yuhwei Yang

Product Data Integration Technology

March 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Abstract

STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic and Syntactic Rules, documents the rules that are used in the integration of STEP draft resource models.[1] The rules are applied in the development of the STEP Integrated Resource that satisfies application requirements for STEP.

Keywords

Conceptual Model, Integration, PDES, Product Data Exchange using STEP, Standard for the Exchange of Product Model Data, STEP

1 This document is N80 of the International Organization for Standardization (ISO) Technical Committee on Industrial Automation Systems (TC 184) Subcommittee on Manufacturing Data and Languages (SC4).

Contents

List of Figures

Definitions

Definitions concerning conceptual modeling [1] useful in the discussion of STEP resource integration include:

abstraction	the result of a conceptualization process which includes generalization/specialization, aggregation, association, and classification.
aggregation	an abstraction in which relationships between low level types of objects can be considered as parts of a higher level object
association	an abstraction in which a relationship between similar types of objects is considered as a higher level set object type
classification	an abstraction in which a type of object is defined as a set of instances
concept	an abstraction derived from the observation of particular instances
conceptual model	information requirements in terms of concepts that are specified as formal structures using the syntax of a modeling language
construct	a logical grouping of conceptual model elements that conveys a semantic idea
draft resource model	a conceptual model that has been approved by a data-modeling project that serves as an origin of resource constructs
generalization	an abstraction in which differences among similar objects are ignored to form a higher level type of object that emphasizes similarities
interpretation	the use of resource constructs to specify context specific relationships and constraints that satisfy application requirements
resource	a construct that has been integrated, and is available for use in the specification of context specific relationships and constraints that satisfy application requirements
specialization	an abstraction in which differences among similar objects are emphesized to form lower level types of objects that maintain the similarities identified in a higher level type of object

STEP (Standard for the Exchange of Product Model Data)
Resource Integration: Semantic & Syntactic Rules

INTRODUCTION

Several principles have been used in the development of these rules.[2] They are stated briefly here, and are elaborated upon in subsequent sections.

1) STEP must contain a cohesive and functionally adequate integrated resource for application protocol [2] interpretation that has an architecture which reduces the impact of change in a phased release environment. It is important to produce a successful STEP Version 1.0 with the ability to add and modify constructs in future versions.

2) STEP will be a collection of Parts each of which is an individual standard with its own scope and unique content. The content of Parts containing semantic constructs is to be conceptual in nature.

3) Constructs are to be within the scope of product data.

4) Constructs are provided for the purpose of supporting application requirements.

5) Constructs are to be functionally adequate for the stated purpose.

6) Constructs are to be functionally unique (i.e., non-redundant).

7) Constructs are to be stable, complete, and correct.

8) Constructs can build upon (i.e., specialize) the semantics of other more generic constructs.

9) Constructs included in a version of STEP are to have an explicit place and role within the schema architecture of the STEP Integration Framework [3].

10) Constructs included in a version of STEP are to be thoroughly integrated with one another.

---

2 This document does not represent an official position of ISO TC184/SC4. Rather, it is a working document of the authors for consideration by WG4, Qualification and Integration, and WG5, STEP Development Methods.

Integration must ensure that the constructs within the STEP Parts form a cohesive whole. Consideration shall be given to all work produced in STEP.

Proper training, thorough understanding of the STEP Integration Framework and strategy, and participation in the Integration Project activities (which involves a detailed understanding of most if not all ongoing work in STEP) is required for membership on the Integration Project team. Ad hoc integration conducted by resource model development teams in isolation of the Integration Project team can be a serious hindrance to thorough integration. These efforts often evolve into individually and independently integrated constructs that are inconsistent with the schema architecture of the STEP Integration Framework and are not readily amenable to the integration practices and rules.

This paper describes the methods and rules that shall be followed to ensure that STEP is thoroughly integrated. The rules apply to constructs that make up the STEP integrated resource, namely contained within the 40 and 100 series STEP Parts. The rules are categorized into semantic and syntactic integration rules.

1 SEMANTIC INTEGRATION RULES

The following rules shall be applied to the draft resource models as they are semantically integrated. Following a detailed presentation and discussion of a draft resource model, the rules are typically applied in the sequence presented.

1.1 MODULARIZATION OF CONSTRUCT

Model elements that convey the logical grouping of a semantic concept form a conceptual construct. Each construct is a "module" within the schema architecture of the STEP Integration Framework and shall be specified in EXPRESS as a schema.

construct	a logical grouping based on meaning (i.e., semantics)
module	an architectural element of the STEP Integration Architecture that contains a construct
schema	the EXPRESS syntax used in the specification of a module

Construct, module, and schema are used throughout this paper based on this definition of modularization. Using an analogous approach to structured programming, the modularization rule partitions product data concepts into manageable groups.

Modularization of constructs provides the following benefits:

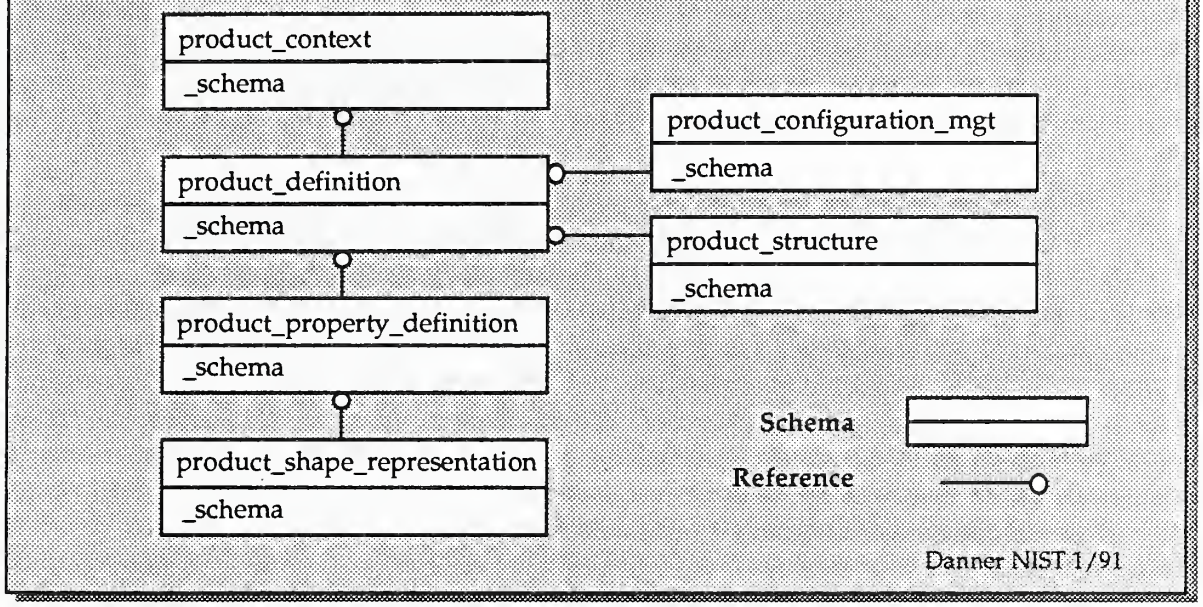
A) The modules are concise to promote readability and understandability.

B) Changes in a phased release environment requires a sound strategy for maintainability. The modularization rule improves the manageability of the specification. This is particularly true because STEP has such large scope and complex concepts.

C) Construct changes can be effected exclusively within modules so long as interfaces between modules are maintained.

Examples include the modules of the Generic Product Data Resources (GPDR) [4] which form the foundation of the integration architecture, and modules that reference the GPDR (e.g., Product Structure & Configuration Manaement).

Figure 1. Example of Resource Modularization



1.2 GENERIC AND CONCEPTUAL NATURE OF CONSTRUCT

Constructs shall be generic and conceptual in nature. Constructs shall convey semantics that logically describe product data concepts. Constructs shall not include ideas or mechanisms that are motivated by convenience in practices, computer technologies, or efficiency requirements for implementation.

generic	shareable among multiple product types, application domains, and life cycle phases; free of context constraints
conceptual	demonstrates consistency (i.e., invarience) of meaning for implementations in heterogeneous computing environments

1.3 PRODUCT DATA SCOPE OF CONSTRUCT

The STEP resource data constructs shall contain semantics that are used to describe a product within the established scope of STEP. STEP should not contain constructs of data necessary to develop an application system that creates or manages the product data. Conceptual constructs of data that are needed for the control, management, and production of a computer system are not to be included as part of the integrated STEP resource specification. Examples of such data semantics are: settings of variables which control the user interface in a CAD system , information necessary to administer data in a DBMS , mechanisms for grouping data that is to be managed in a certain fashion, and control variables in navigating a process or a data structure.

1.4 APPLICATION REQUIREMENTS FOR CONSTRUCT

Constructs that demonstrate support for application requirements shall be given the highest priority for inclusion as part of the integrated STEP resources. Integration will focus on constructs that are required by application protocols.

Many draft resource models have been developed in the absence of formal application requirements or of the recently established application protocol development framework. Consequently, while the models may convey semantics of product data, they are not traceable to an application requirement. The application protocols will ensure that data in the standard is traceable to a requirement, and is therefore testable and implementable.

However, documentation and distribution for wide review of draft resource models that lack specific application requirements is not prohibited. A different status of publication is necessary to motivate progress, expand scope, and stimulate application protocol development that require additional constructs. These publications should be able to formally solicit review and comments without the negative effects of upward compatibility concerns and without constraints on improvements that would result if formal application requirements suggest different constructs.

1.5 FUNCTIONAL UNIQUENESS OF CONSTRUCT

A construct shall not be a functional duplicate of another construct. This is not to be confused with structural duplication (i.e., use of templates). Integration applies this rule in the identification of commonality of construct and elimination of duplication. A new construct is developed that is a synthesis of the duplicate constructs (i.e., it fulfills all identified requirements). The new construct is then placed appropriately in the integration architecture based on semantic considerations.

1.6 FUNCTIONAL ADEQUACY OF CONSTRUCT

A model development project shall have well defined goals and scope. Each of the constructs in a draft resource model shall be traceable to an element of the established goals and fit within the established scope. Every construct shall fulfill a defined purpose. Inclusion of specific concepts will be based on the semantic necessity for the concept. Examples and test cases shall be employed to evaluate the usability (functional adequacy) of the construct against its declared purpose (e.g., collection and population of representative data).

1.7 CONCEPTUAL STRUCTURE OF CONSTRUCT

The stability, completeness, and correctness of the conceptual structure of a construct must be examined for integration. The intent is to promote the goal of integration: to develop the best quality in STEP exemplified by cohesiveness and unification. The following criteria shall be used as guiding principles for the joint review and potential restructuring of a candidate construct by the Integration project team and the technical experts who developed the model.

1.7.1 Structure Stability

The structure of a construct shall be considered stable by both the integration and technical expert teams using heuristic criteria. There shall not be an outstanding issue regarding stability raised by either team. The intent of the stability rule is to ensure that the structure of a construct has had sufficient review and that broad agreement is obtained before a construct is documented in a STEP Part.

1.7.2 Structure Completeness

The structure of a construct shall completely accommodate the goal for which it is intended. Constructs shall have demonstrated relationships among entities (i.e., do not provide a "container" for miscellaneous entities). The integration team, based on requirements for shareability across disciplines, will determine if the construct serves common goals within the overall STEP architecture.

1.7.3 Structure Correctness

The structure of a construct shall be technically correct in its modelling approach to the defined goals and within the defined scope. Template structures shall be used when similarity of semantics warrant there applicability. Integration applies this rule to examine issues of generality, consistency, and compatibility with other constructs in the integrated resource.

1.8 CONSTRUCT SPECIALIZATION AND EXTENSION

A construct that specializes or extends the semantics of generic entities shall reference those entities from the more generic modules. The Generic Product Data Resources (GPDR) and Generic Product-Data Management Resources (GPMR) [5] constructs are the most generic. Constructs shall reference the GPDR or GPMR in order to specialize or extend concepts.

All entities in the integrated 40 series STEP Parts shall be unique. An entity or construct within a draft resource model, identified as necessary to convey an understanding of the STEP architecture, shall be moved to Part 41, the GPDR and GPMR. The construct may require modification in order to be generic (i.e., context independent) and shareable. Syntactic rules 2.1 and 2.2 address the mechanism by which this is accomplished.

The construct specialization rule is applied to address the overview function of the generic constructs contained within the GPDR and GPMR modules. It is applied to establish control of the amount of detail in the generic and extending constructs. By controlling the amount of details in a given module, the references across module boundaries can be minimized and better managed. Minimization of such references (i.e., interface points) between modules facilitates the accommodation of change.

1.9 PLACEMENT IN THE STEP INTEGRATED SCHEMA ARCHITECTURE

A construct that is to be integrated shall have a logical place in the schema architecture of the STEP Integration Framework. If this is not possible, then either the construct or the architecture must change. Over time, because of the impact on the already integrated constructs, it is going to be easier to consider changes in a proposed additional construct than in the integration architecture. This is not meant to preclude change in the integration architecture where it is warranted.

Resource constructs, as they are placed in the architecture, must be understandable in that broader context. Thus, in applying this rule, integration will raise issues of understandability, correctness, and completeness of a construct in an integrated context, which may be different than the context in which the draft resource model was developed.

It is anticipated that the application of this rule will have significant impact on some projects that have developed models prior to the establishment of the STEP Integration Framework. To prevent similar problems in the future, draft resource model developers are encouraged to begin communication with the Integration Project as early as possible in the development of their models.

1.10 THOROUGHNESS OF INTEGRATION

A construct shall be integrated within the entire scope of STEP. Individual draft resource model development project teams shall not attempt integration of their work without the participation of the Integration Project team. An integrated resource shall have consistent levels of abstraction and usage of specification languages. A STEP Part shall document only those constructs from a draft resource model that have been fully integrated. It must accommodate an unambiguous method of interpretation for application requirements.

2 SYNTACTIC INTEGRATION RULES

The following rules have been developed to provide guidelines for syntactic integration. They define the mechanisms which are used to implement the semantic integration rules. Where possible the relationships between the syntactic rules and the semantic rules are identified. There is no priority implied by the numbering of the syntactic rules.

2.1 CONSTRUCT MODULARIZATION BY SCHEMA SPECIFICATION

A construct that becomes a module in the STEP Integration Architecture, as a result of the integration process, shall be specified as a schema. It shall be specified as a subschema of the STEP Integrated Resource Schema.

> A referencing module shall specialize or extend at most one generic construct.

A draft resource model that is a specialization or extension of multiple constructs or contains disjoint ideas, will be split into multiple schemas. The modules will be placed in semantically appropriate STEP Parts.

> A more particular construct references a more generic construct to which it adds (i.e., extends) or specializes semantics.

Modules in the 40 series STEP Parts shall use the EXPRESS "REFERENCE" keyword to specialize or extend concepts from the GPDR modules. They must specialize or extend the semantics of a single GPDR construct (i.e., the module must fit at a single specific location in the GPDR structure).

An example can be found in the draft resource model for Part 43. The original model had only one schema. It included such definitional items as shape_aspect and such representational items as shape_aspect_representation. These entities are now separated into two modules that deal with shape aspect definition and shape aspect representation.

2.2 CONTROLLED REFERENCES BETWEEN AND WITHIN MODULES

References between entity types are a primary integration issue. Both inter- and intra-module references shall be controlled to ensure consistency of semantic interpretation and manageability of the specification. (Semantic Rules 1.1 & 1.9)

Attribute references from one entity type to another, where the two entities span module boundaries, are controlled by the integrated schema architecture.

> Reference direction between modules shall be determined on the basis of existence dependence and definitional dependence.
>
> Reference direction from one entity to another within the same module shall be determined on the basis of existence dependence.

2.2.1 Existence Dependence

Existence dependence involves a relationship between entities where an instance of one entity cannot be present without the presence of an instance of another entity. Existence dependence is most easily identified by the cardinality constraints of the relationship between two entities.

A is existence dependent on B when:

1) A is related to one and only one B (1,1), AND
   B is related to zero, one, or many As (0,n), OR
   B is related to zero or one A (0,1);

or

2) A is related to one or many Bs (1,n), AND
   B is related to zero, one, or many As (0,n), OR
   B is related to zero or one A (0,1).

The EXPRESS specification of relationships involving existence dependence shall use an attribute in the dependent entity, that has as its type, the entity upon which it is dependent.

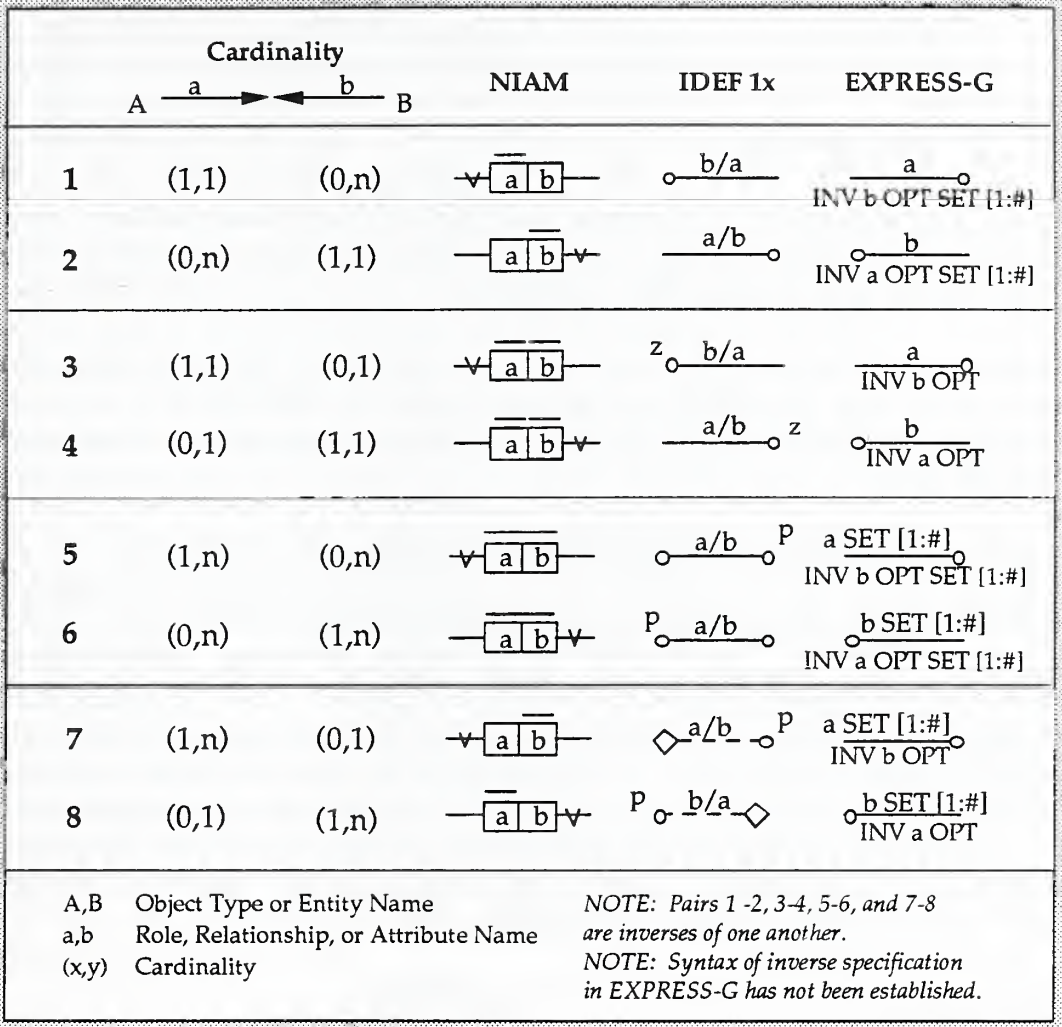
```
ENTITY entity_a;                    -- specification of A
   attribute_of_a:  entity_b;       -- reference to B [3]
END_ENTITY;
```

The application of the existence dependence rule results in a consistent manner by which EXPRESS specifications are made and a consistent way of translating between NIAM and IDEF1x representations of these relationships.

---

[3] The cardinality of the reverse relationship between B and A must also be specified.

Figure 2. Specification of Existence Dependence



An example from the GPDR involves the relationship between a "product" and a "product_version." An intuitive approach to this relationship suggests that a product may have many versions (i.e., a cardinality of(0,n)). The reverse of this relationship is that a given product version can be of only one product (i.e., a cardinality of (1,1)). The existence of a product version is dependent on there being a product. Therefore, the EXPRESS specification of this relationship has an attribute in the "product_version" entity that has as its type "product."

```
ENTITY product_version;
   of_product: product;
END_ENTITY;
```

The resulting EXPRESS specification may seem to be counter-intuitive. Unfortunately, rules for a computer sensible specification can not be based on often inconsistent human intuitions. A consistent use of this rule minimizes many upward compatibility issues. Along with the practice of this rule, the inclusion of EXPRESS-G which graphically presents references with explicit inverse information will improve the human understandability.

2.2.2 Definitional Dependence

Definitional dependence exists between two modules when one module, the dependent or referencing module, must have access to a type name in another module, the referenced module, for compilation.

When references are required between modules that are of the same level of conceptualization (e.g., the GPDR and its extensions) the REFERENCE keyword will be used in whichever module has the definitional dependence on an entity of another module.

An example is the references required between the PSCM configuration management schema and the GPDR product definition schema. Through the application of the existence dependence rule, the "configuration_design" entity remains in the PSCM configuration management schema because of its dependence on the "generic_configuration_item" entity. The "configuration_ design" entity is also existence dependent on the "product_version" entity of the GPDR product definition shcema. However, there is a logical division of concept between the product definition schema and the configuration management schema. The former is an extension of the more generic semantics. Therefore, the configuration management schema must reference the product definition schema for access to the "product_version" entity.

```
SCHEMA
    product_definition_schema;

    ENTITY product;
        --
    END_ENTITY;

    ENTITY product_version;
        of_product: product;
        --
    END_ENTITY;

    ENTITY product_definition;
        version: product_version;
        --
    END_ENTITY;

END_SCHEMA;
```

```
SCHEMA
    configuration_management_schema;

    REFERENCE FROM
        product_definition_schema
        (product_version);

    ENTITY generic_configuration_item;
        --
    END_ENTITY;

    ENTITY configuration_design;
        design: product_version;
        config: generic_configuration_item;
        --
    END_ENTITY;

END_SCHEMA;
```

Another example that is more historical in nature, rather than due to Integration Project modularization, exists between topology and geometry. The topology schema uses the REFERENCE keyword to gain access to entity type names of geometry.

Such references are an integration issue, since this kind of inter-module dependence encompasses semantic modularity and upward compatibility issues. References based on definitional dependence need to be controlled. Such an inter-module reference should only exist when the entity being referenced is necessary to the semantics of the construct of which it is a part, and the referencing module specifies a construct that is adding semantics to that entity. Inter-module references that do not comply with this principle are most probably an artifact due to arbitrary schema boundaries based on committee organization and discipline interest.

The solution in these cases is to move entities from one schema to another to achieve appropriate definitional and existence dependence. If two schemas have many low level references between them, it is often an indication that their scopes are overlapping or their boundaries (i.e., scopes) are not clearly defined. Where the modules overlap, common entities should be included in only one schema, or in a third new schema created to be shared by the two if a shared semantically independent construct is identified.

Ideally, by referencing the entities from a more generic schema, an extension schema should be self sufficient in its definition. Interfaces are controlled by generic constructs being definitionally independent, with dependent extensions adding semantics. This approach creates a very open system, in which subsequent versions of STEP can, for the most part, simply add new dependent constructs to the more generic constructs which serve as the foundation of Version 1.0.

Since many of the draft resource models defined their boundaries long before the establishment of an integrated architecture, pre-existing boundaries may demand references between modules that are not as theoretically well founded. Definitional dependence applied without existence dependence is a temporary solution. Definitional dependence based on existence dependence is the preferred mechanism toward which the Integration Project is moving. This situation is particularly true in the case of the shape models. Proper use of the rules for controlled references between and within modules will result in fewer interfaces (i.e., inter-module references) in the integrated resources.

2.3 PLACEMENT OF SUPERTYPE & SUBTYPE ENTITIES

SUBTYPE / SUPERTYPE declarations which span schema boundaries shall be specified such that the supertype entity exists in the more generic module

without the explicit SUPERTYPE declaration (supports the construct specialization semantic rule, 1.8). The referencing schema (containing a REFERENCE) shall declare an entity as a SUBTYPE of a referenced entity.

An example of the use of this rule is the placement of the "shape_model" entity in the GPDR Shape Representation Schema. The "shape_model" entity is the supertype of all representation methods. The specializing schema references the "shape_model" entity from the GPDR Shape Representation Schema to create SUBTYPE specializations .

```
SCHEMA
    product_shape_representation_ schema;

    ENTITY shape_model;
    --
     END_ENTITY;

END_SCHEMA;
```

```
SCHEMA geometric_shape_schema;

 REFERENCE FROM
    product_shape_representation_schema
    (shape_model);

 ENTITY surface_model
    SUBTYPE OF (shape_model);
    --
 END_ENTITY;

END_SCHEMA;
```

SUBTYPE / SUPERTYPE relationships between two specializing schemas are not allowed. This is an indication that the supertype entity is of a more generic nature and, therefore, should be moved to a more generic module, the GPDR or GPMR schemas defined in Part 41.

2.4 PLACEMENT OF ENTITY DEFINITION & DESCRIPTION

If an entity is moved to a more generic module, such as one of the GPDR schemas, and is referenced in its original source schema, all text which supports the EXPRESS entity specification definitions, attribute descriptions, constraints, and proposition descriptions will be moved into the more generic schema. If the semantics of the entity are further specialized in the domain of the extension schema, the GPDR schema will provide a generic definition of the entity and the referencing schema shall include additional descriptions of the entity necessary in its context. The definitions and descriptions need not be identical but must be consistent. Also included in the description is an identification of the document in which the referenced schema appears (e.g., ISO 10303 Part 41).

An example is the definition and description text for the entity "shape_aspect" as they appear in both the Product Property Definition Schema of the GPDR and the PSIM (Product Shape Interface Model) Shape Aspect Definition Schema.

3 SUMMARY

STEP Resource Integration is a cooperative effort between the Integration Methods Project of WG5, the Resource Integration Project of WG4, and data modeling projects of WG3. The experience gained through the interaction of these STEP projects, at numerous integration meetings and workshops, has led to the development of the integration rules presented in this paper. The rules will continue to be refined, and additional rules will be added as we work toward the development of an integrated resource for STEP.

Semantic Integration Rules

1. Modularization of Construct
2. Generic & Conceptual Nature of Construct
3. Product Data Scope of Construct
4. Application Requirements of Construct
5. Uniqueness of Construct
6. Functional Adequacy of Construct
7. Conceptual Structure of Construct
   - stability
   - completeness
   - correctness
8. Specialization & Extension of Construct
9. STEP Integration Architecture Placement of Construct
10. Thorough Integration of Construct

Syntactic Integration Rules

1. Construct Modularization by Schema Specification
2. Control of Inter- & Intra Module References
   - existence dependence
   - definitional dependence
3. Placement of SUPERTYPE & SUBTYPE Entities
4. Placement of Textual Entity Definitions and Descriptions

References

[1] Brodie, M.L., Mylopoulos, J., and Schmidt, J.W.; On Conceptual Modelling, Springer-Verlag, NY, 1984.

[2] Palmer, M., Gilbert, M., and Andersen, J.; Guidelines for the Development and Approval of STEP Application Protocols,NISTIR, National Institute of Standards and Technology, Draft January 1991.

[3] Danner, W.F.; A Proposed Integration Framework for STEP (Standard for the Exchange of Product Model Data), NISTIR 90-4295, National Institute of Standards and Technology, April 1990.

[4] Danner, W.F. and Yang, Y.; Generic Product Data Resources (GPDR), NISTIR, National Institute of Standards and Technology, Draft January 1991.

[5] Danner, W.F., Shaw, N., and Yang, Y.; Generic Product-Data Management Resources (GPMR), NISTIR, National Institute of Standards and Technology, Draft January, 1991.

Acknowledgments

The authors wish to thank Dr. Kent Reed, Leader of the Computer Integrated Construction (CIC) Group, for his continued support and encouragement in producing this document.

Appreciation is extended to the members of the Integration Methods, Resource Integration, and product data modeling projects of ISO TC 184/SC4/WG5 (STEP Development Methods), WG4 (Qualification and Integration), and WG3 (Product Data Modeling). Many hours of direct integration experience have led to the establishment of the rules contained in this document.

Special appreciation is extended to Buzz Bloom, who, as the technical representative of the first draft resource to be integrated with the STEP generic product data resources, contributed immensely as the catalyst for the documentation of rules by which integration is to proceed in STEP.

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4528

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
MARCH 1991

4. TITLE AND SUBTITLE

STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic & Syntactic Rules

5. AUTHOR(S)

William F. Danner; David Sanford; Yuhwei Yang

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic and Syntactic Rules, documents the rules that are used in the integration of STEP draft resource models. The rules are applied in the development of the STEP Integrated Resource that satisfies application requirements for STEP.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

Conceptual Model; Integration; PDES; Product Data Exchange using STEP; Standard for the exchange of Product Model Data, STEP

13. AVAILABILITY

- [X] UNLIMITED
- [ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
- [ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
- [X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
20

15. PRICE
A02

ELECTRONIC FORM