NISTIR 5972

Developing Application Protocols (APs) Using the architecture and methods of STEP (STandard for the Exchange of Product data)

Fundamentals of the STEP methodology

William F. Danner

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

NIST

NISTIR 5972

Developing Application Protocols (APs) Using the architecture and methods of STEP (STandard for the Exchange of Product data)

Fundamentals of the STEP methodology

William F. Danner

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

January 1997

U.S. DEPARTMENT OF COMMERCE
William M. Daley, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

BLANK PAGE

Abstract

STEP has provided an architecture and methods for the development of application protocols (APs). An AP is a standards document (a part of ISO 10303) that provides for communication of information in a well defined application context. The use of an AP ensures that the information conveyed is that which was intended. It also ensures that the information conveyed is adequate for specific uses of product data identified as the purpose of the AP.

This report presents a tutorial for the development and use of APs using the architecture and methods of STEP. It provides definitions, rationales, and examples of the principle components of the STEP architecture as well as their use for a sample population. The sample population is presented using both clear text encoding and an example relational database implementation. The presented definitions, rationales, and example AP provide a foundation for a strategy to develop and use interrelated application protocols (IAPs).

Keywords: Standard for the exchange of prodcut data, STEP, application protocol, AP, interrelated application protocol, IAP, data exchange, data sharing, information exchange, information sharing.

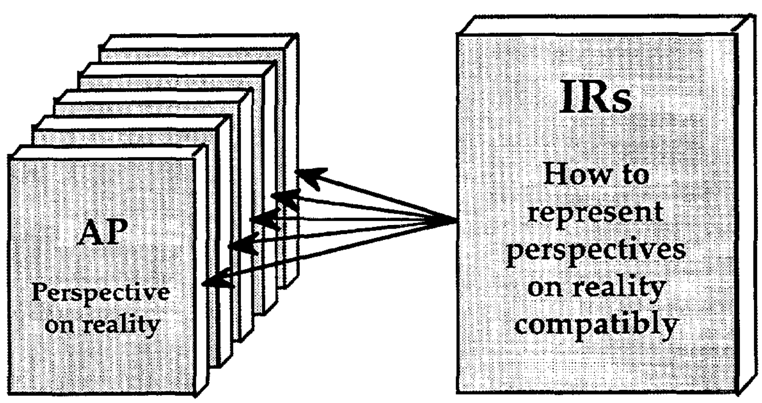
BLANK PAGE

Foreword

> The rules of the game: learn everything, read everything, inquire into everything . . . . When two texts, or two assertions, or perhaps two ideas, are in contradiction, be ready to reconcile them rather than cancel one by the other; regard them as two different facets, or two successive stages, of the same reality, a reality convincingly human just because it is complex.

These words of Marguerite Yourcenar [1] end a book on complexity by John L. Casti [2]. Casti describes complexity as being directly related to the number of different ways of looking at the same thing. Modeling a product is complex in precisely this way. There exist an indefinite number of ways to look at any given product. Each provides a context for description, a perspective that is useful for some purposes but not others. These perspectives differ not only in terms of their utility but also in terms of their scope and granularity of abstraction (e.g., products as systems, as aggregations of components, or as materials comprised of chemical compounds). The totality of what is known about a product is the confluence of all perspectives and this is continually changing.

The complexity inherent in our descriptions of products is reflected in the product data architecture of STEP (STandard for the Exchange of Product model data). The STEP product data architecture has two principal elements, the integrated resources (IRs) and application protocols (APs). The IRs comprise a standard way of representing how we look at products. The IRs contain abstract generic constructs that accommodate different ways of looking at the same products through life cycle dependent product definitions. APs capture particular ways of looking at one or more products for specific purposes. An AP contains a standard description of its perspective based on the use (i.e., the application) of specific product data. It covers a limited portion of a product's life cycle. An AP also contains a standard reprsentation of its perspective that employs the constructs of the IRs. This reprsentation portrays its perspective as a compatible member of an indefinite number of possible perspectives about a product.



Principal elements of the STEP product data architecture.

BLANK PAGE

Table of contents

BLANK PAGE

Introduction

The primary criterion of STEP is the utility of communicated information among end users. Every effort has been made to provide a means to capture the information needed by users to perform their tasks. STEP enables computer systems to be used collaboratively based on a common understanding of information utility. Information has utility when the meaning of the information (i.e., the semantics) and the background knowledge necessary to draw proper inferences about the information (i.e., the context) are understood [3]. The semantics and context of information together describe an ontology that users employ when they talk about a product. An ontology describes, in the context-dependent terminology of the user, those things considered to exist, the characteristics of those things, and the relationships among those things.

A product used in the building industry might be thought of as a beam using the context of structural design or as a reinforced concrete element using the context of concrete prefabrication (Fig. 1). The relevant properties of the product when viewed as a beam might include its load capacity, while when viewed as a reinforced concrete element might include the type of concrete. Each description of the product contributes to what is known about the product, and each description uses a different ontology in which the semantics and context make the information understandable and useful.

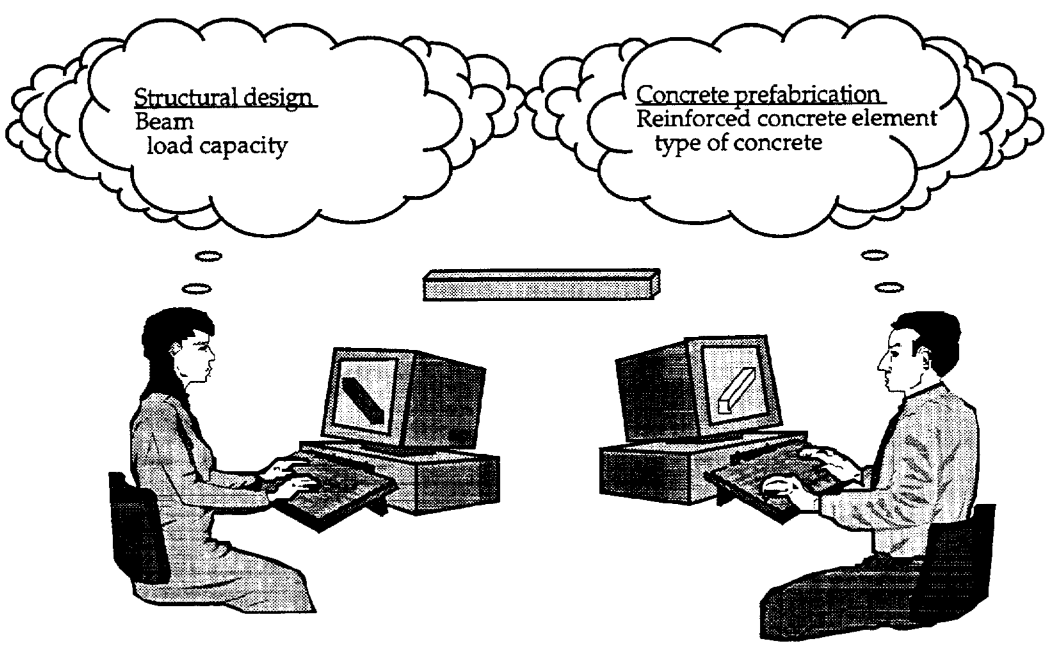


Figure 1. Different ontologies employed to describe the same product.

An application protocol (AP) contains a description of a domain ontology (i.e., an ontology suitable for the applications covered by the AP). The semantics and context of the ontology are described using four elements of an AP (Fig. 2). The context of the ontology is established through a description of its application context. The application context is presented graphically as an application activity model (AAM). Those portions of the AAM specifically supported by the AP constitute the application scope. The semantics of the ontology are specified through an analysis of the in-scope elements of the AAM to define the specifics of the AP's information requirements.

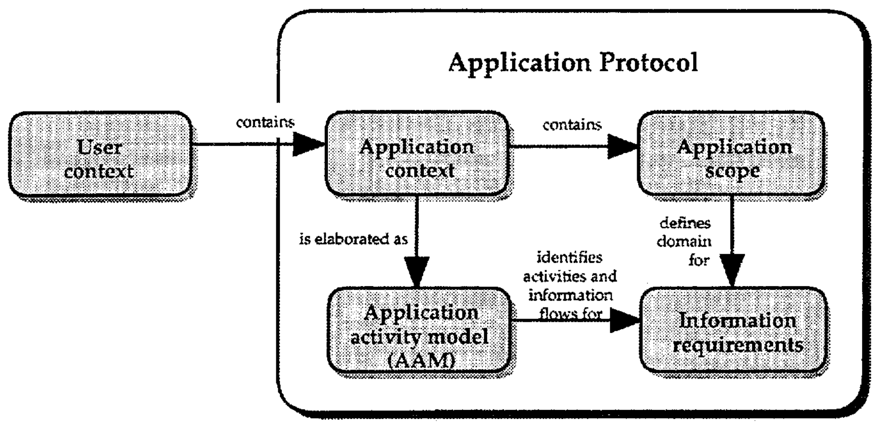


Figure 2. Elements of an AP that describe the semantics and context of its domain ontology..

The context of a user may include one or more application contexts covered by APs depending on the tasks performed. An application context of an AP contains descriptions of the functionalities, technologies, types of product, disciplines, industry sectors, and life cycle stages that comprise the background knowledge of users as they perform specific tasks. The activities and flows of information among activities within the application context are presented in an AAM. The activities and flows of information that are specifically addressed by an AP are identified as being within its application scope. The information flows in scope are analyzed to identify the information requirements of users addressed by an AP. The information requirements include natural language definitions of the application objects (the things considered to exist and their characteristics) and application assertions (the relationships among the things) using the terminology of the users. The information requirements are presented graphically as an application reference model (ARM) to aid visualization.

In order to ensure that a useful collection of information is available to the user for specific tasks, conformance classes are established (Fig. 3). Conformance classes group specific information requirements based upon usage scenarios that describe how the information is expected to be used. A usage mapping identifies the usage scenario, the activities and information flows of the AAM relevant to the scenario, and the information requirements used to satisfy the usage scenario.

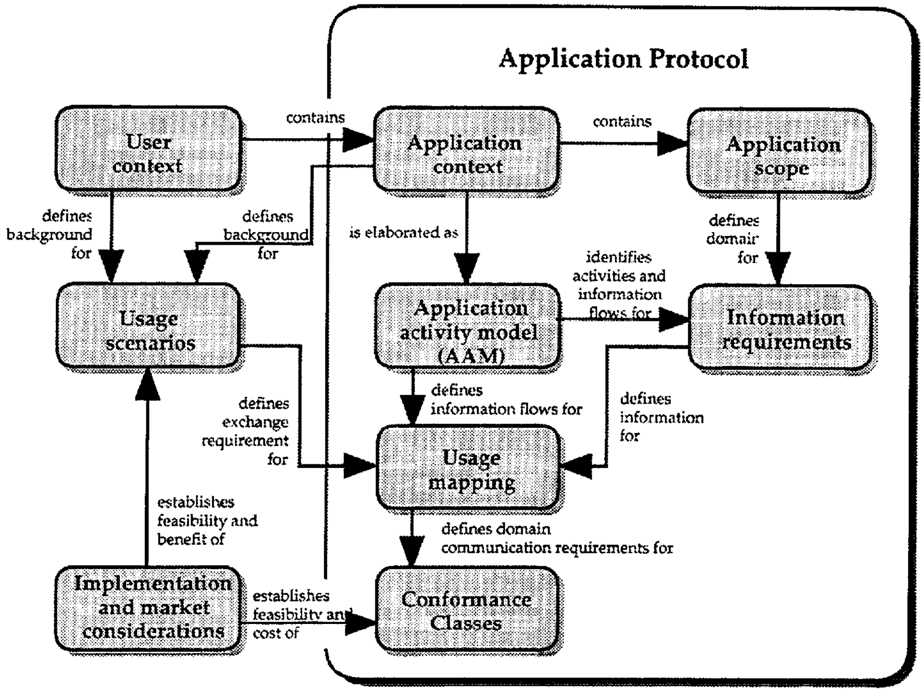


Figure 3. Specification of conformance classes in an AP.

Both the usage scenarios and the conformance classes that are established to satisfy the scenarios are influenced by implementation and market considerations. Implementation considerations address the feasibility of the scenario in terms of existing and future computer systems as well as the ability of these systems to adhere to the resulting conformance class. Market considerations include the benefits of satisfying the usage scenario and the costs of providing systems that meet the requirements of the conformance class.

A conformance class based on the information requirements of a domain ontology is the culmination of the primary focus of STEP on the utility of information. The second focus of STEP is to specify the conformance class of an AP in terms of a formal representation of the domain ontology that includes both information requirements and application context. The domain ontology is considered as an application context dependent way of looking at a product. As such it contributes in a coherent fashion to the totality of what is known about the product. This entails representing all ontologies in a consistent manner that provides compatibility among ontologies used to describe the same product. To achieve this goal, APs formally specify their information requirements and selected aspects of their application contexts using the constructs of the integrated resources (IRs).

The IRs describe a generic ontology for product data. IR constructs are used to provide a formal normative representation that explicitly associates AP information requirements (i.e., the semantics of the AP domain ontology) with elements of the AP application context (i.e., the context of the AP domain ontology) (Fig. 4). The constructs within the IRs must be able to represent all information requirements and all application contexts of AP domain ontologies. In the event that the available resources are inadequate, additional constructs are added to the IRs. STEP is, therefore, extensible with respect to its representation of domain ontologies.

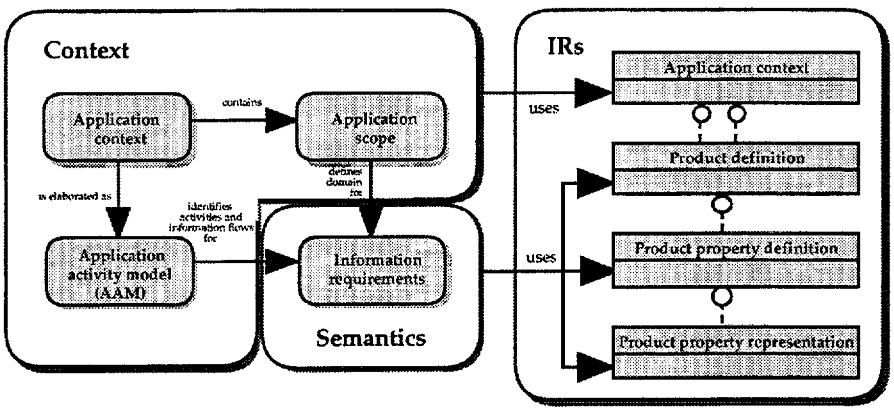


Figure 4. Elements of the IRs used to represent the ontology of an AP.

The most crucial constructs within the IRs with respect to associating the context and semantics of an AP's domain ontology are those used to specify product descriptions [4]. Characteristics ascribed to a product when employing user terminology are represented as property representations associated with property definitions that are associated with a product definition that is meaningful in a specific product definition context. The product definition context specifies the type of product perspective and life cycle stage for which the product definition is appropriate. An indefinate number of product definitions may be useful over an entire product life cycle. Each product definition is therefore associated with an identified product that is meaningful in one or more product contexts. Each product context specifies the industry sector and discipline for which the existence of the product is meaningful.

An identified product only has property definitions through its associated product definitions. For interrelated APs this means that a product and its characteristics (as described in user terminology) are represented by different but compatible life cycle dependent product definitions. These product definitions are associated with the same product. The product is, in turn, identified as having meaning for different technical disciplines. The product definition construct of the IRs is a primary basis for reconciling the different domain ontologies within most interrelated APs that address different perspectives of the same product.

A representation mapping (Fig. 5) establishes the correspondence between the information requirements and elements of the appplication context of the AP ontology, the constructs used from the IRs to represent these information requirements and contexts, and the resulting representation of the AP ontology as an application interpreted model (AIM) [5]. The AIM is the formal specification for communication when satisfying a conformance class of an AP. Since there exists a mapping between the information requirements of an AP and its conformance classes, and between the information requirements and the constructs within the AIM, a conformance class is stated normatively in an AP using mapping tables [6] and AIM constructs. A conformance class can be thought of as a subset of AIM constructs to be used in a manner specified by the mapping tables to satisfy specific functional requirements.

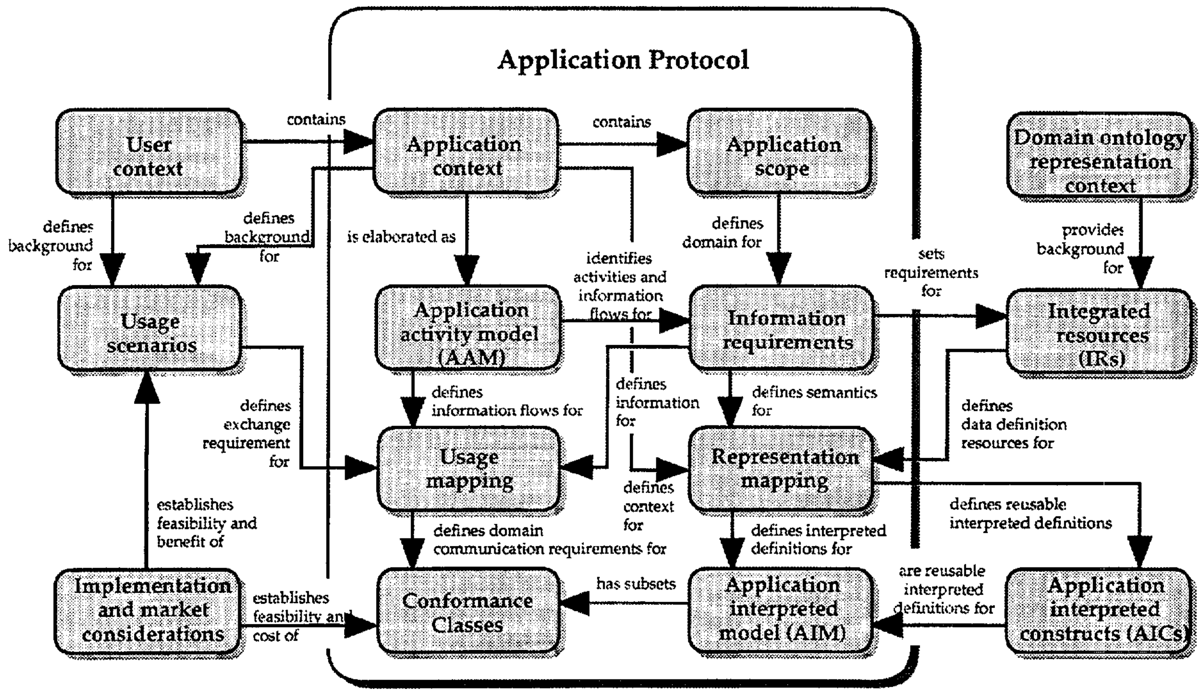


Figure 5. Representation of AP application context and information requirements in terms of standard constructs of the IRs.

Groups of AIM constructs that are used to represent common semantics in more than one AIM are specified as a reusable application interpreted construct (AIC) [7]. The appropriate use of AICs is critical to the common understanding of the semantics of information needed for the coordinated use of interrelated APs (IAPs). An AIC satisfies information requirements that are common to more than one AP. Whether or not an AIC is appropriate for information sharing depends on the details of the common information requirements and upon a shared context. As the essential first step, the information must be within both the appplication context and scope of both APs for information to be commonly understood. The application context must include common functionality, technology, type of product perspective, and life cycle stage that comprise the background knowledge for use of the information. More than one industry sector and multiple disciplines may share this knowledge about the product.

The objectives of an AP are (1) the description of a domain ontology which includes both its semantics (i.e., information requirements) and context (i.e., application context), (2) the formal representation of the described domain ontology (i.e., AIM), (3) the specification of the mapping between the description and representation of the domain ontology (i.e., mapping tables), and (4) the specification of subsets of AIM constructs that comprise conformance classes to ensure that specific functional end user requirements are met. The conformance classes of an AP ensure that conforming implementations provide the utility needed to fulfill identified information usage scenarios.

The objective of the IRs is to provide the fundamental constructs that are used for the normative representation of domain ontologies within all APs. IR constructs are used to develop AIMs that represent the domain ontologies of APs in such a way that the information requirements and selected elements of the application contexts are specified consistently and compatibly among ontologies that deal with the same products. Mapping tables specify the correspondence between the information requirements of a domain ontology description and its representation in an AIM. They also specify how the AIM is to be used by defining reference paths and constraints within the AIM data structure that must be followed. AICs provide common representations of information requirements that when coupled with a common application context and overlapping scopes provides for a common understanding of information. This is the basis of information sharing.

The following sections describe how an AP is developed in terms of the architecture and methods of STEP. It provides simple but technically complete example architecture components of an AP. It also provides a sample population that is presented both using the STEP format for physical files and an example relational database implementation. Finally, it provides a general strategy for using interrelated APs in an integrated application context environment where the use of information can be coordinated over a products entire life cycle.

1. Describing a consensus domain ontology

A consensus domain ontology that employs the terminology of the user is developed as the basis for unambiguous and useful communication. The method used by STEP is conceptualization (Fig. 6). Usage scenarios that reflect how information is expected to be used act as the input to the activity. User expertise serves as the control to ensure accuracy and completeness for the purposes identified within the usage scenarios.

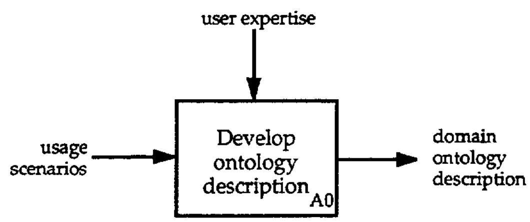


Figure 6. The conceptualization method.

Conceptualization has three elements (Fig. 7), each having usage scenarios as their inputs and controlled by user expertise. The first is to describe the context of the ontology. This results in an application context description, an AAM, and a description of the application scope as outputs. The in-scope information flows of the AAM serve as inputs to describing the ontology semantics. This results in the definition of information requirements and an application reference model (ARM) as outputs. The ARM is a graphical presentation of the information requirments as an aid to visualization. The application objects of the information serve as the input to the description of the conformance classes of the AP. The application objects are grouped as units of functionality (UoFs). The conformance classes are defined in terms of information requirements that must be met to satisfy one or more usage scenarios. The outputs of all three activities constitute the domain ontology description of the AP.

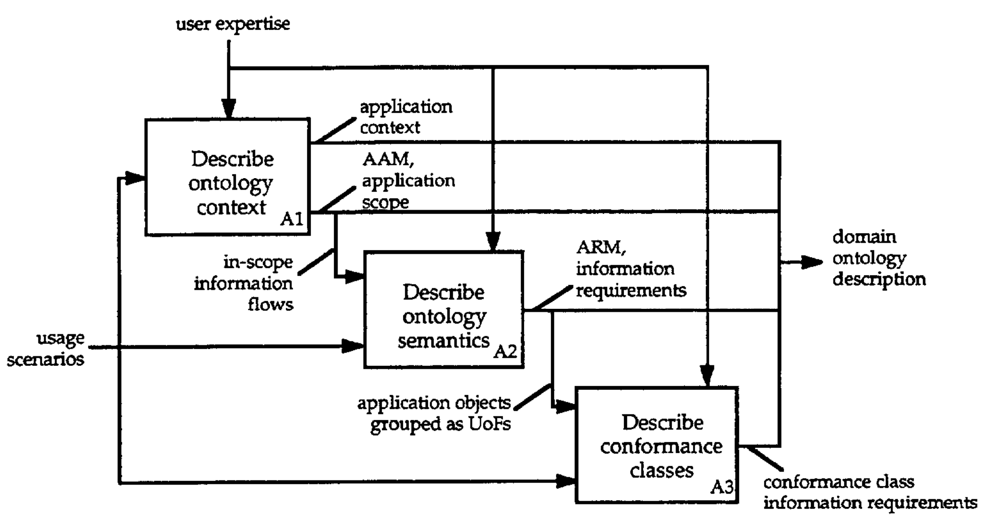


Figure 7. Decomposition of the conceptualization method.

1.1 Describing the context of an AP

The context of a domain ontology is described by three components of the STEP product data architecture. They are the application context, an application activity model, and the application scope.

1.1.1 Describing an application context

An application context identifies background knowledge that is necessary to make proper inferences about the use of communicated information. This includes an identification of the functionality, technology, industry sector, discipline, type of product perspective, and life cycle stage for which an AP is developed. AP 1 (Table 1) is an example AP developed for the building industry with the particular focus of as-required performance halls. It addresses the use of viability analysis techniques as employed by economists during the economic design requirements specification stage of the life cycle.

Application context	Example
application protocol	AP 1
functionality / technology	economic viability analysis
industry sector	building industry
discipline	economics
type of product perspective	as-required
life cycle stage	specify economic design requirements

Table 1. An example application context

STEP does not prescribe descriptions for application context elements. An exception is that actual APs are given numbers in the 200s (e.g., AP 201). That practice is not followed here to avoid any possible confusion with actual APs. As STEP continues to develop, it is possible that the freedom with which the application context is described may be constrained in order to provide greater consistency among APs. Such constraints could be developed in such a way that they do not limit domain experts from making distinctions that are important in their description of an AP's domain ontology.

Two aspects of the application context are relevant to the description of an AAM. They are the functionality / technology and the life cycle stage. The functionality / technology in this example identifies the purpose (i.e., economic viability analysis) for which one or more domain perspectives are relevant. The life cycle stage (i.e., specify economic design requirements) identifies the particular activity that provides the context for one or more specific domain perspectives within the AP. The information flows of this activity are critical to the AP and to those APs with which it may be interrelated.

1.1.2 Specifying an application activity model (AAM)

An application activity model identifies the activities of an application domain in which information is created and used. STEP uses IDEF0 [8] to develop activity models. STEP does not specify, however, a standard general activity model for use in the development of APs. A general activity model for the purpose of example could include such activities as plan product, specify design requirements, develop design specification, construct product, and operate and maintain product (Fig. 8).

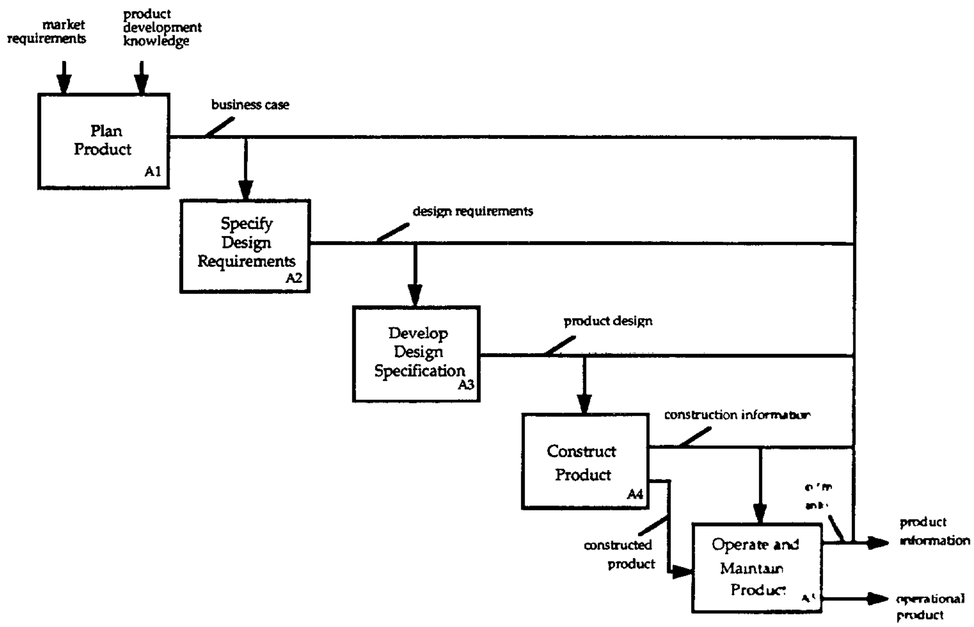


Figure 8. Example general activity model of a product.

In this general model, market requirements and product development knowledge are controls on the activity plan product. A business case is the output of plan product and a control on the activity specify design requirements. Design requirements are an output of this activity and a control to develop design specification. Product design is an output of develop design specification and a control to construct product. Construction information is the output of construct product and a control to operate and maintain product. Constructed product is also an output of construct product and is an input to operate and maintain product. Operation and maintenance information is an output of operate and maintain product as is operational product.

Product information is a collective output of all five activities that includes the business case, design requirements, product design, construction information, and operation and maintenance information. All five activities can be thought of as a single activity called effect product (Fig. 9) that has market requirements and project development knowledge as controls. Product information and the operational product are outputs. Product information created by this activity is within the scope of STEP.

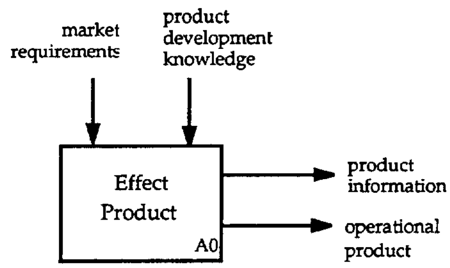


Figure 9. Generalized activity that produces product information.

Any AP could use the first two levels of a general model simply by replacing the word product with the kind of product addressed by the AP (e.g., Effect performance hall). The outputs at these levels are too large for APs which address specific limited scopes. Therefore, further decomposition of the activities is undertaken. The nature of the decomposition is typically dependent on the industry sector (i.e., an aspect of the product context) and type of product perspective (i.e., an aspect of the product definition context) identified within the application context. For the example application context of AP 1, specify design requirements may be decomposed into specify economic requirements, specify staging requirements, and specify acoustical requirements. Only specify economic requirements would be within the scope of example AP 1.

1.1.3 Defining an application scope

An application scope describes the activities and flows of information that are specifically covered by an AP. For AP 1, the decomposition of the activity called specify design requirements would indicate the elements that are in scope (shaded and bold face type) and those that are out of scope (italics) (Fig 10).

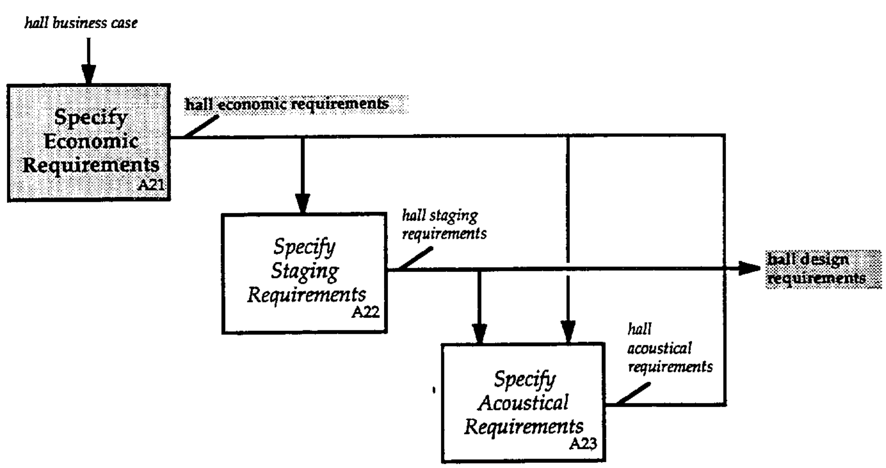


Figure 10. Decomposition of specify design requirements for AP 1.

In this example, the activity specify economic requirements is in scope as is its output hall economic requirements. This output serves as a control to specify staging requirements which is out of scope. The hall economic requirements are a part of the information flow hall design requirements which is a control to develop design specification. Having identified the information flow that is critical to AP 1, an information analysis is undertaken to determine the specific information requirements of the AP.

1.2 Defining the semantics (information requirements) of an AP

An in-scope information flow, such as hall economic requirements, is analyzed with particular attention to groupings of information requirements that must be considered collectively to capture specific user concepts completely and unambiguously. These groupings are referred to as units of functionality (UoFs). The information requirements within UoFs contain definitions of application objects (the things of the domain ontology and their characteristics) and application assertions (the relationships among the things). As the information requirements approach consensus, they are grouped as conformance classes that ensure specific utility of information based on an analysis of specific usage scenarios.

1.2.1 Defining application objects

Application objects name things within the domain ontology and the properties of those things. An analysis of hall economic requirements results in two application objects, Performance_hall and Target_audience_capacity.[1] The properties of the Performance_hall are its Name and Id. The properties of the Target_audience_capacity are the Seating_capacity and the Standing_capacity. The application objects and their properties are defined using natural language.

Hall economic requirements:

Perfomance_hall
A large room for the presentation of entertainment to audiences.

Id
An identifier (not necessarily human interpretable) used to designate a performance hall.

Name
A label (human interpretable) used to designate a performance hall.

---

[1] Application object definition can easily be influenced by data modeling technology. This example identifies an object called Target_audience_capacity. A domain expert might have chosen to define instead a third characteristic of performance hall called target capacity that had two ordered values. The first value being assumed to be the seating capacity and the second value the standing capacity. This might be standard practice in the discipline in which the domain expert works.

Target_audience_capacity
The minimum number of persons that must be accommodated by a performance hall for economic viability.

Seating_capacity
The number of persons that may be seated in a performance hall during a performance.

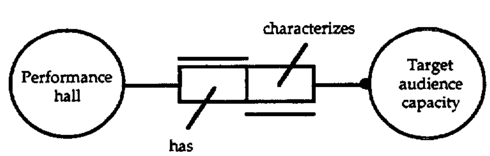
Standing_capacity
The number of persons that may stand in a performance hall during a performance.

1.2.2 Defining application assertions

Application assertions describe relationships among the application objects of the domain ontology and the constraints that apply to those relationships. Their description uses well-formed natural language statements. The application assertions are presented graphically as an aid to understanding the described ontology. Graphical presentations can use NIAM [9], IDEF1x [10], and EXPRESS-G [11]. NIAM graphical presentations can be derived directly from well-formed natural language statements of application assertions. The graphical presentations can also be used to generate well formed natural language statements. Therefore, NIAM is used for example AP 1.[2]

Example AP 1 contains two application objects, Preformance_hall and Target_audience_capacity. The description of the application assertion first states the two way binary relationship among these application objects and then restates the relationship specifying the constraints.

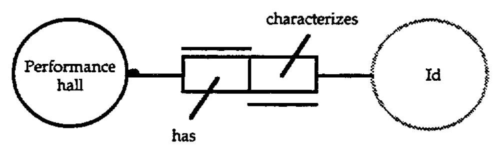
Performance_hall has Target_audience_capacity
/ Target_audience_capacity characterizes Performance_hall
Each Performance_hall has at most one Target_audience_capacity and
Every Target_audience_capacity characterizes exactly one Performance_hall



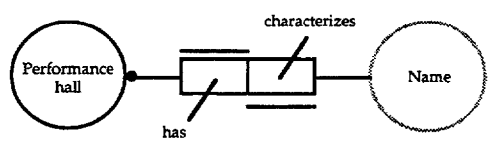
Application assertions can also be used to describe the relationships between the application objects and their characteristics.

[2] IDEF1X and EXPRESS-G are used more often than NIAM in STEP APs. However, the correspondence between natural language statements and their presentation using NIAM is used in an effort to make the examples as clear as possible (see Annex A for a description of NIAM graphical elements).

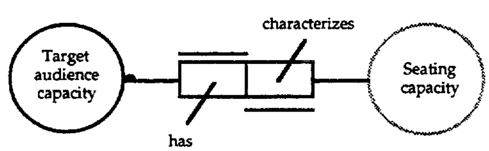
Performance_hall has Id
/ Id characterizes Performance_hall
Every Performance_hall has exactly one Id and
Each Id characterizes at most one Performance_hall



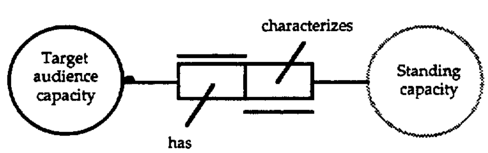
Performance_hall has Name
/ Name characterizes Performance_hall
Every Performance_hall has exactly one Name and
Each Name characterizes at most one Performance_hall



Target_audience_capacity has Seating_capacity
/ Seating_capacity characterizes Target_audience_capacity
Every Target_audience_capacity has exactly one Seating_capacity and
Each Seating_capacity characterizes at most one Target_audience_capacity



Target_audience_capacity has Standing_capacity
/ Standing_capacity characterizes Target_audience_capacity
Every Target_audience_capacity has exactly one Standing_capacity and
Each Standing_capacity characterizes at most one Target_audience_capacity



The distinction between a relationship between an application object and one of its characteristics and a relationship between two application objects is a result of how the users think about the information. In this example, the users think of a performance hall as having three characteristics. A performance hall always has an Id and a Name but optionally may have a Target_audience_capacity depending on the stage in the product's life cycle. The Target_audience_capacity is comprised of two characteristics of a performance hall that are always dealt with as a pair. These distinctions will become significant when the information requirements are transformed into an AIM.

For the purpose of this paper, application assertions include the relationships among application objects and their properties and the constraints on these relationships. These constraints are specified descriptively as mandatory or optional characteristics of the application objects.

The application assertions can be presented collectively (Fig. 11) as an application reference model (ARM) when they include both kinds of relationships (i.e., both between application objects and between application objects and their characteristics)[3]. The ARM is used extensively during development of consensus for the information requirements and as an introduction to the domain ontology of an AP by those using the standard.

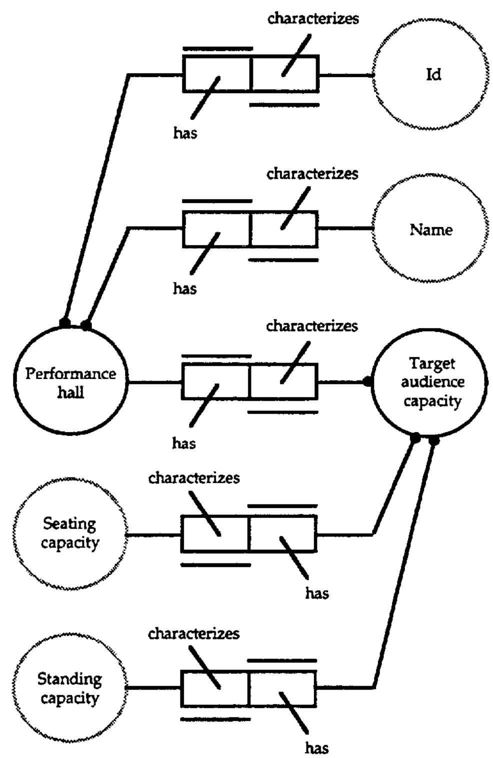


Figure 11. NIAM graphical presentation of application assertions as an ARM.

The use of a NIAM graphical presentation for an ARM allows for a direct correspondence with well-formed natural language description of the application assertions as well as application object characterization in an AP. The application assertions and descriptions of application objects including their characteristics using natural language comprise the information requirements of an AP.

---

3 NIAM diagrams are more clear for relistic ARMs when application assertions (i.e., relationships between application objtects are presented separately from application object descriptions (i.e., relationships between application objects and their characteristics). This would result in three presentations for example AP 1 (i.e., one for the relationship between Performance hall and Target audience capacity, one for relationships between Performance hall and its Id and Name. and one for relationships between Target audience capacity and its Seating capacity and Standing capacity).

1.3 Establishing conformance classes

Establishing conformance classes is among the principal objectives of AP development. They identify the information requirements that ensure the utility of communicated information as described in specific usage scenarios.

1.3.1 Describing usage scenarios

A usage scenario is a description of one or more events that involve the communication of information about a product among users of that information. The presentation of events for a usage scenario identify the flow of information at a particular time in the life cycle of a product (Fig. 12).

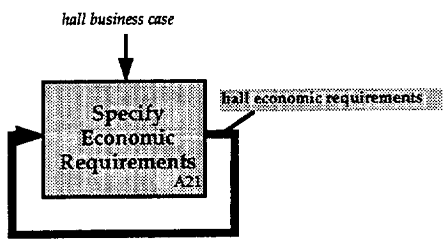


Figure 12. Event one of the usage scenario for example AP 1.

Event one of the usage scenario for AP 1 involves the flow of information (bold arrow) as an output and an input of the same activity specify economic requirements. Upon decomposition, detailed activities and information flows may be identified. An AP reflects whether such detail is warrented (greater specificity) or not (greater flexibility). One group of users may create some of the hall economic requirements which may be modified or added to by others.

Event two involves the flow of hall economic requirements from specify economic requirements to specify staging requirements (Fig. 13).

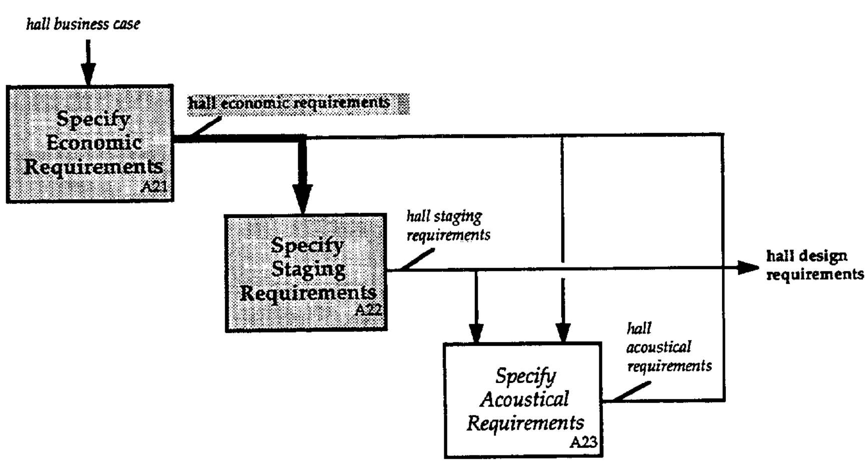


Figure 13. Event two of the usage scenario for example AP 1.

A third event involves the flow of the same information from the activity specify economic requirements to the activity specify acoustical requirements (Fig. 14).

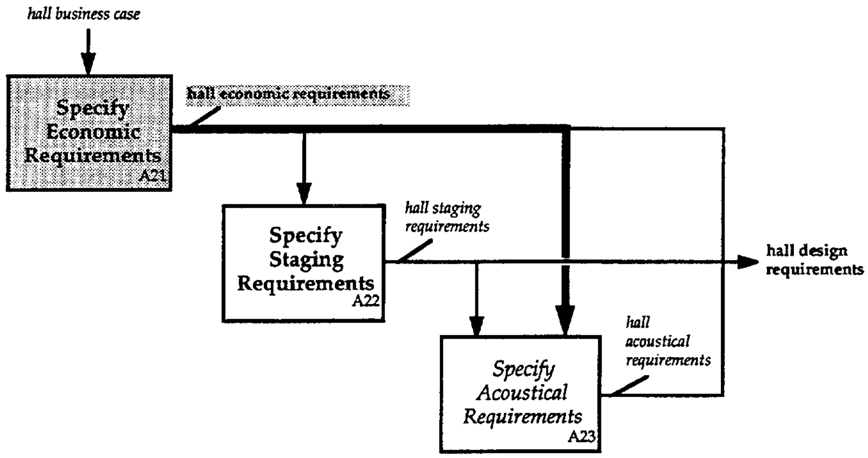


Figure 14. Event three of the usage scenario for example AP 1.

The fourth event of the usage scenario for example AP 1 is the flow of this same information between the activity specify economic requirements and the activity develop design specification (Fig. 15). The hall economic requirements are an output of specify economic requirements and as a part of hall design requirements, are a control to develop design specification.

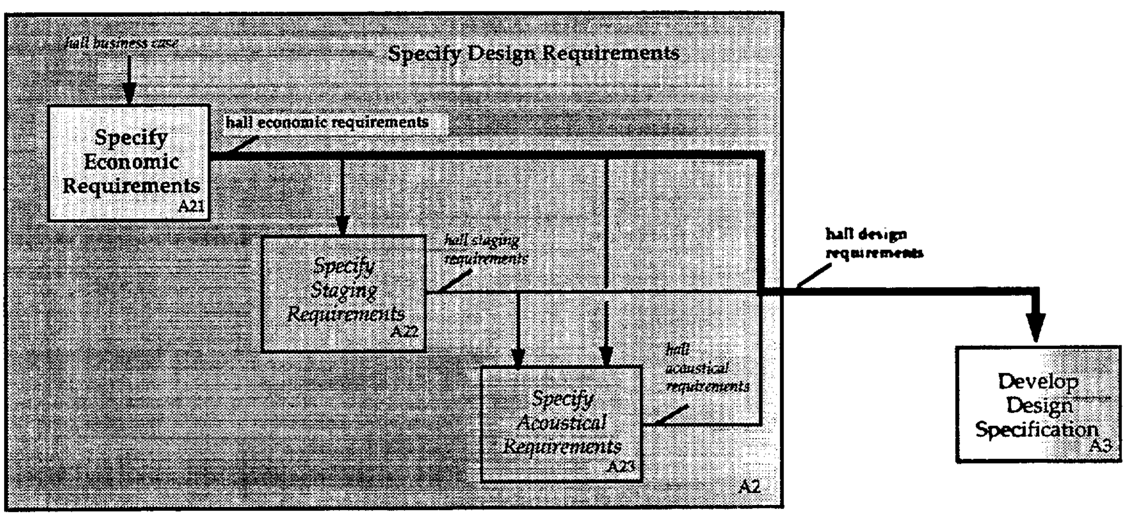


Figure 15. Event four of the usage scenario for example AP 1.

In the usage scenario of example AP 1, the identical information is involved in each of the events. This is not necessarily the case for all usage scenarios. Events may involve different or overlapping information all of which will be included in identifying the information requirements of the usage scenario.

1.3.2 Identifying conformance class information requirements

The information requirements of a conformance class are identified by listing the application objects that are covered by the conformance class (Table 2). Application assertions that relate included application objects are assumed to be part of the conformance class. Information requirements are selected based on a balance between benefits to industry and the feasibility and cost of satisfying the conformance class.

	Conformance Classes	
Application Objects	CC 1	CC n
Performance_hall	x	x
Target_audience_capacity	x	x
(other application objects)		x

Table 2. Identification of the application objects of a conformance class for example AP 1.

In the usage scenario for example AP 1, all of the application objects that are part of the hall economic requirements are included in a single conformance class for AP 1. The simplicity of the example prevents defining additional conformance classes that could include additional information. Often conformance classes are developed based on what is generally achievable with today's systems, what is achievable by some of the more advanced systems, and what are the goals of the users for systems yet to be developed.

2. Representing a consensus domain ontology

The consensus domain ontology of an AP includes the description of an application context, information requirements, and conformance classes. The formal representation of an AP's domain ontology is as a collection of mapping tables and an application interpreted model (AIM) using the STEP method of interpretation (Figs. 16 and 17). Interpretation involves developing and using the constructs of the integrated resources (IRs) to represent the application context (i.e., the context of the ontology), the information requirements (i.e., the semantics of the ontology), and the conformances classes. An explicit association is established between the context and the semantics of the domain ontology. The semantics are represented in such a way that the domain ontology is compatible with other related domain ontologies. Interpretation is possible as a direct result of the domain ontology representation capabilities of the IRs.

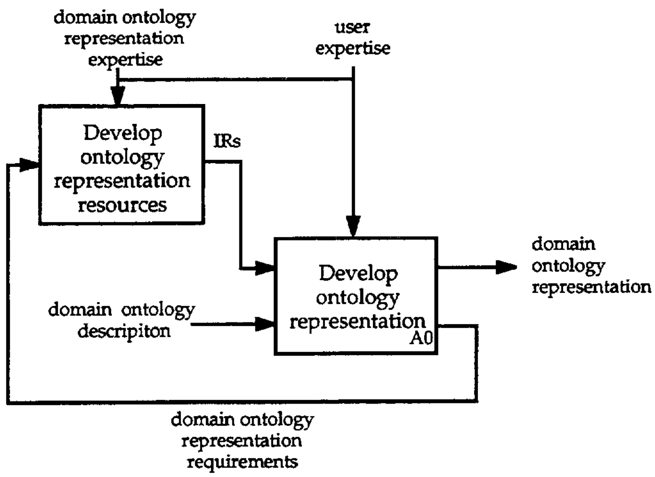


Figure 16. The resource integration and interpretation methods.

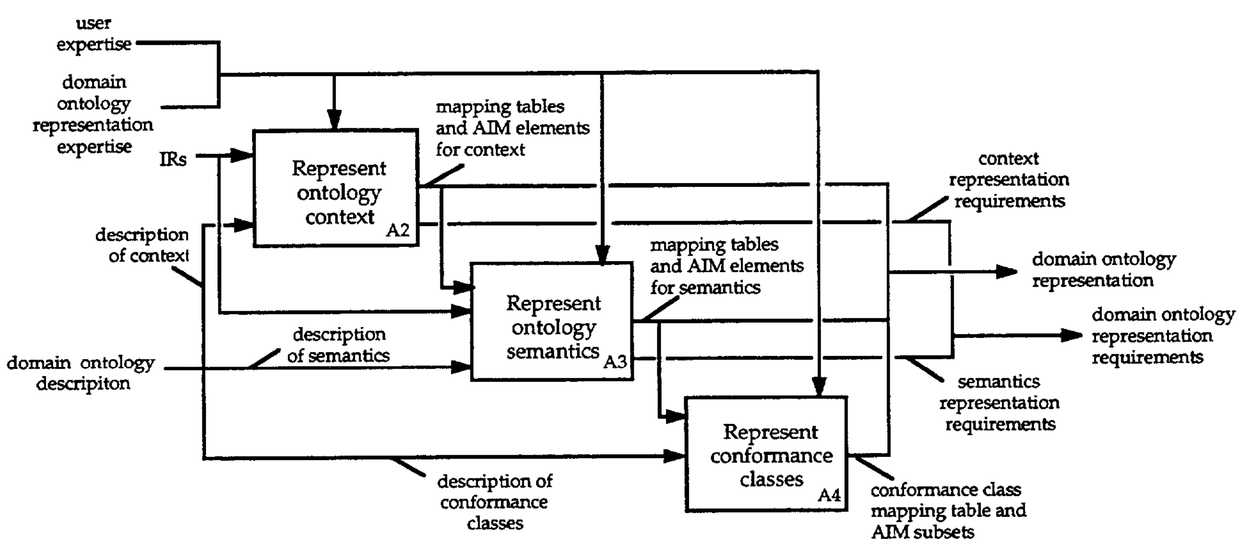


Figure 17. Decomposition of the interpretation method.

2.1 Developing domain ontology representation resources

The IRs are developed to represent the domain ontologies of all APs. They are developed in response to domain ontology representation requirements. Principal among the requirements addressed by the IRs is that constructs for representing both the context and semantics of a domain ontology be available with explicit relations between them. Another requirement is that the constructs be generic in nature. That is, that they represent concepts that are common to all domain ontologies at a level of detail that provides the ability to explicitly represent that which is often implicit in the way users think, talk, and process information about a product.

The IRs contain core generic resource constructs applicable to the representation of all domain ontologies and additional resource constructs that may be used as needed to represent specific domain ontologies. The IRs are modular and extensible so that as new representation requirements are identified, they can be accommodated. Core generic constructs that illustrate the representation of domain ontologies are contained within seven IR modules (Fig. 18). IR modules are specified as schemas using EXPRESS [11]. They include the application context schema, the product definition schema, the product property definition schema, the product property representation schema, the representation schema, the measure qualification schema, and the measure schema [4,12,13].

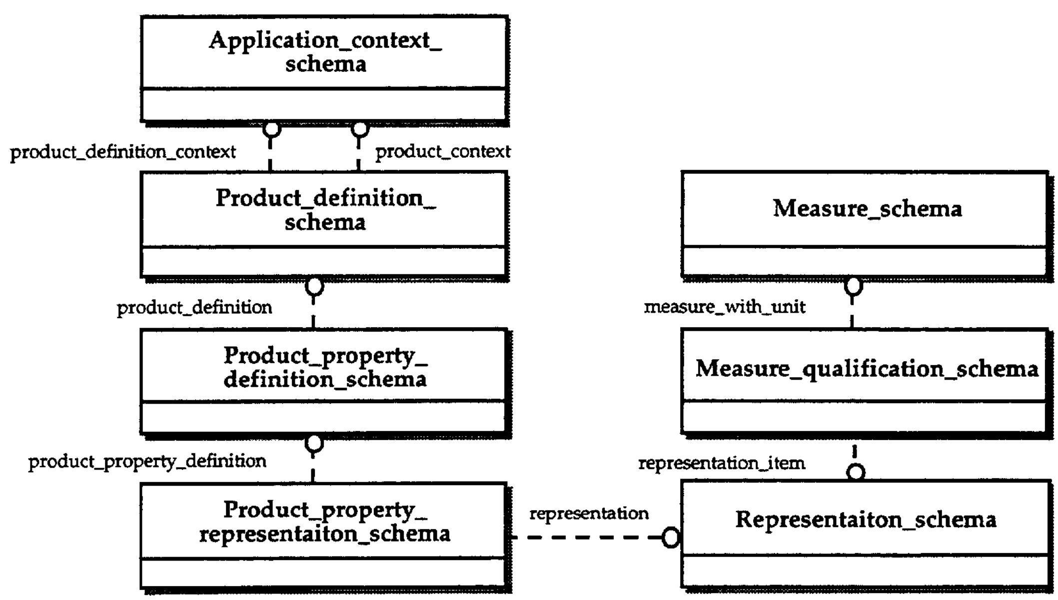


Figure 18. Selected modules of the integrated resources (IRs)

The first four of these schemas (i.e., the application_context_schema, the product_definition_schema, the product_property_definition_schema, and the product_property_representation_schema) are called generic product description resources. They contain, along with the representation_schema, atomic constructs for product description. The measure_qualification_schema and measure_schema are used to specify details of a particular kind of representation that involves values and units used here to illustrate the principles of domain ontology representation.

References among the modules provide integration interfaces. The constructs, therefore, comprise a single integrated model that is used in the representation of domain ontologies.

2.2 Using generic constructs to represent a domain ontology

Interpretation is the method employed to transform an AP domain ontology from a natural language description using the terminology of the user to a formal normative representation as mapping tables and an AIM. The generic constructs of the IRs are used to specify the context and semantics of the ontology in such a way that there is explicit representation of the ontology as a member of an indefinite number of perspectives on the products within scope.

The modules of the IRs contain generic constructs that are used to represent and associate the context (i.e., the application context) and semantics (i.e., the information requirements) of AP domain ontologies. The context of an AP is represented using the constructs of the application_context_schema. The semantics are represented using constructs from the other IR modules.

2.2.1 Representing the context of a domain ontology

Application context

The application context schema contains constructs that are used to represent the context of an AP domain ontology. The constructs can be described as resource objects, characteristics of the objects, and assertions that describe relationships among the objects.[4] Since the IRs are specified using EXPRESS, both resource object characteristics and assertions appear as entity attributes. Therefore, the description of resource objects includes the assertions (in italics) following the characteristics of a resource object.[5]

[4] The descriptions presented in this document are not taken from the standards documents of STEP (i.e., ISO 10303 parts [13]). Rather, they reflect the views of the author about the meaning of the constructs stated as natural language domain ontology representation requirements (similar to AP information requirements).

[5] Assertions are bidirectional. Each direction has constraints on the relationship. EXPRESS divides assertion into two components. A resource object contains that portion of the assertion that references another object. This is either as an inverse attribute or if not present as an inverse, is implicit and assumed unconstrainted. SUBTYPE and its inverse SUPERTYPE are presented as assertions defining a special relationship (i.e., is_a) that involves inheritence of properties and assertions from the object playing the role of supertype to the object playing the role of subtype.

Resource objects and assertions:

application_context_schema
A representation of the contexts of AP domain ontologies.

application_context
An identification of the context of a domain ontology that provides background knowledge needed for proper inferences about the use of information about a product.

application
A description that identifies the functionality/technology covered by a domain ontology.

application_protocol_definition
An identification of the AP in which the application context of a domain ontology is described.

status
The standing of that AP as assigned by the ISO.
application_interpreted_model_schema_name
The human interpretable label ascribed to the AIM of the AP.
application_protocol_year
The year of the AP.
application
Every application_protocol_definition references exactly one application_context as application

application_context_element
An identification of an aspect of the context of a domain ontology that provides background knowledge needed for proper inferences about the use of specific information about a product.

name
A description that identifies the industry sector (for a product context) or type of product perspective (for a product definition context) covered by a domain ontology.
frame_of_reference
Every application_context_element references exactly one application_context as frame_of_reference
SUPERTYPE OF (ONEOF (product_context, product_definition_context))
Every application_context_element may be either a product_context or a product_definition_context[6]

[6] The application_context_schema specifies that an application_context_element is either one of its subtypes or is not a subtype at all. However, in AIMs, an application_context_element is always one of its subtypes.

product_context
An identification of an aspect of the context of a domain ontology that provides background knowledge needed for proper inferences about the use of information concerning the existence of a product.

discipline_type
The identification of a discipline for which the existence of a product is meaningful.
SUBTYPE OF (application_context_element)
Every product_context is an application_context_element

product_definition_context
An identification of an aspect of the context of a domain ontology that provides background knowledge needed for proper inferences about the use of information concerning a definition of a product.

life_cycle_stage
The identification of the life cycle stage in which a definition of a product is created.
SUBTYPE OF (application_context_element)
Every product_definition_context is an application_context_element

The resource assertions are presented as an EXPRESS-G diagram (see Annex) as an aid in visualizing the relationships among resource objects (Fig. 17). Characteristics of resource objects are not presented graphically.

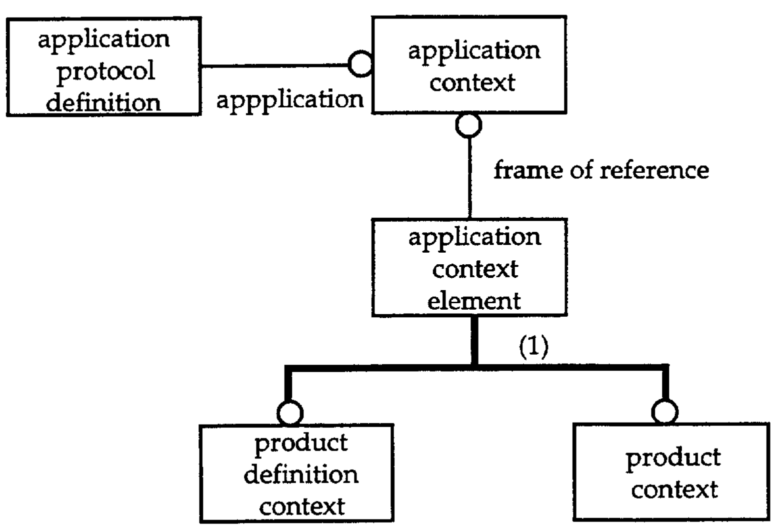


Figure 17. Assertions of the application_context_schema.

There is a correspondence between the elements of an application context description and constructs of the application context schema (Table 3). The values of status and year attributes of the application protocol definition are provided at the time an AP is actually used. The example AP 1 application context description is presented in parentheses.

application_context_schema	Application context description
application_protocol_defintition status aim_schema_name year	 - AP AIM schema name (AP1_aim_schema) -
application_context application	 functionality / technology (economic viability analysis)
product_context name discipline_type	 industry sector (building industry) discipline (economics)
product_definition_context name life_cycle_stage	 product type (as-required) life cycle stage (specify economic design requirements)

Table 3. Using the constructs of the application_context_schema.

The correspondence between the application context description and its representation using constructs of the application_context_schema are presented formally as a representation mapping (Table 4). The development of a representation mapping (or mapping table) for the description of the application context of an AP domain ontology is not currently part of interpretation in STEP. This mapping table applies the same principles used for the development of mapping tables for information requirements which is part of STEP interpretation method.

The representation of the application context description involves constraints on attribute values for entities of the application_context_schema. The application interpreted model schema name of an application protocol definition must be ap1_aim_schema. The application of an application context must be economic viability analysis. The name and discipline type of a product context must be building industry and economics. The name and life cycle stage of a product definition context must be as-required and specify economic design requirements.

The representation mapping is used to identify the constructs of the IRs that are needed in the AIM of the AP. For the example AP 1 the AIM contains the entities application_context, application_context_element, product_context, and product_definition_context. An EXPRESS schema is defined for the AIM that employs the USE FROM statement.

Application element	AIM element	Source	Rules	Reference path
AP	application_ protocol_ definition	41		{application_protocol_definition application_protocol_definition. application_interpreted_model_ schema_name= 'ap1_aim_schema'}
Functionality / technology	application_ context	41		{application_context application_context.application= 'economic viability analysis'}
Industry sector / discipline	product_context	41		{application_context_element <- [application_context_element application_context_element.name= 'building industry'] product_context product_context.discipline_type= 'economics'}
Type of product perspective / life cycle stage	product_ definition_context	41		{application_context_element <- [application_context_element application_context_element.name= 'as-required'] product_definition_context product_context.life_cycle_stage= 'specify economic design requirements'}

Table 4. Mapping table for example AP 1 application context description.

The preliminary AIM satisfying the requirements of the application context description (i.e., the context of the AP domain ontology) would use these entities.[7]

```
SCHEMA  ap1_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

END SCHEMA;
```

The formal description of the application context description of an AP domain ontology includes both the representation mapping (i.e., a mapping table) and the AIM. This is also true for the information requirements that use other modules of the IRs. The reference path constaints of the representation mapping are an essential part of how a domain ontology is represented using STEP.

[7] Only entities that are to be independently instantiated need be included. This is an AP development decision. In this example, only application_context_element is not explicitly identified since only instances of its subtypes, product context and product definition context, will be instantiated.

2.2.2 Representing the semantics of a domain ontology

Product definition

The product definition schema contains constructs that are used to represent the semantics of an AP domain ontology with respect to the existence of a product relevant to one or more technical disciplines and life cycle dependent definitions of that product (Fig. 17). The life cycle dependent definitions of a product serve as a point of aggregation for life cycle dependent properties but are independent of these properties. Properties are represented using the constructs of the product property definition schema. The product definition schema contains additional constructs some of which are described here.

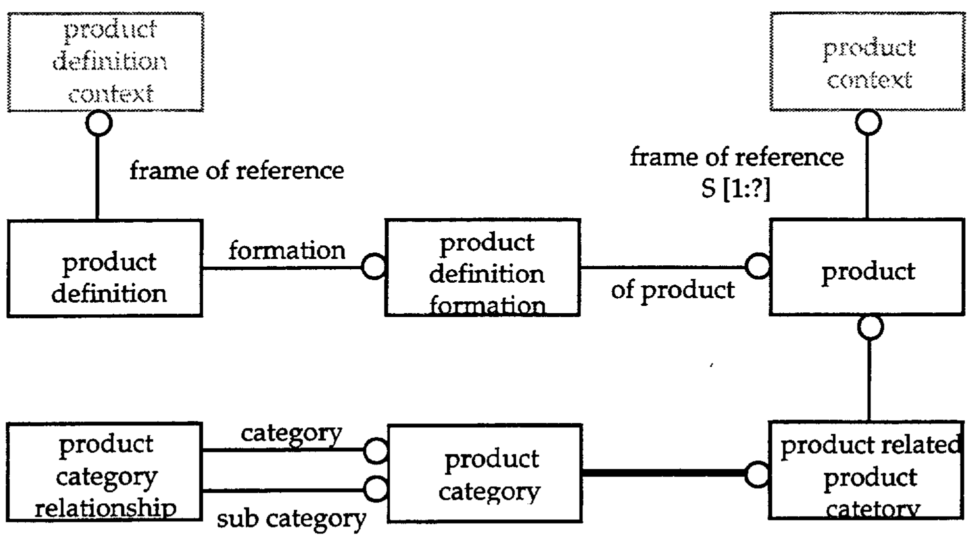


Figure 17. Elements of the product_definition_schema.

product_definition_schema:

A representation of the semantics of AP domain ontologies with respect to the existence of a product and life cycle dependent definitions of a product.

product

An identification of the declared existence of a thing thought of as relevant to one or more technical disciplines.

id

An identifier for the thing declared to exist.

name

A label that is human interpretable for the thing declared to exist.

description

Text used to designate the nature of the thing declared to exist.

frame_of_reference

Every product references at least one product_context as frame_of_reference product_category

An identification of a class applicable to products.

name

A label that is human interpretable for the class.

description

Text used to designate the nature of the class (optional).

product_related_product_category

An identification of a class that is associated with one or more products.

products

Every product_related_product_category references at least one product as products

SUBTYPE OF (product_category)

Every product_related_product_category is a product_category

product_category_relationship

An identification of an association among two classes where one plays the role of class to the other plays the role of subclass.

name

A label that is human interpretable for the ordered association.

description

Text used to designate the nature of the ordered association.

category

Every product_category_relationship references exactly one product_category as category

sub_category

Every product_category_relationship references exactly one product_category as category

product_definition_formation

An identified aggregation of life cycle dependent product definitions for a product that are considered together for some purpose (e.g., a version).

id

An identifier for the aggregate of product definitions.

description

Text used to designate the nature of the aggregation.

of_product

Every product_definition_formation references exactly one product as of_product

product_definition

An identification of a life cycle dependent aggregation of property definitions.

id

An identifier for the aggregation of property definitions.

description

Text used to designate the nature of the aggregation of property definitions

formation

Every product_definition references exactly one product_definition_formation as formation

frame_of_reference

Every product_definition references exactly one product_definition_context as frame_of_reference

An AP (e.g., example AP 1) uses these constructs to represent the semantics of its domain ontology that deal with the definition of a product. The information requirements regarding a performance hall application object and their representation using constructs of the product definition schema are presented formally as a representation mapping (Table 5).

Application element	AIM element	Source	Rules	Reference path
Performance_ hall	product	41		{product <- product_related_product_category.products[i] product_related_product_category <= (product_category product_category.name product_category.name = 'performance hall') (product_category <- product_category_relationship.sub_category product_category_relationship product_category_relationship.category product_category.name = 'performance hall')}
id	product.id	41		product product.id
name	product.name	41		product product.name

Table 5. Mapping table for example AP 1 use of product construct from product definition schema.

A performance hall, as thought of by the user, is represented as a product. The reference path indicates that it is a product that is either associated with a product category that has a name with a value of performance hall or is a product that is associated with some other product category that plays the role of subcategory in a relationship with a category that has a name with a value of performance hall. The reference path can be used to develop queries against an AIM data structure. Such a query would yield information about the application element described in the information requirements. It provides a formal link between the way a user thinks about the information and the way it is represented in an AIM.

The properties of a performance hall (i.e., id and name) are mapped to the id and name attributes of the product entity. These properties are not part of a life cycle dependent product definition, but rather are attributes used to identify the existence of a product throughout its entire life cycle. The target audience capacity, however, is a property of a product definition that is created during the specify economic requirements stage of the life cycle. To capture and maintain the information about a product over the course of its life cycle, a life cycle dependent product definition can be specified in the mapping table that is available for reference by an appropriate property definition (Table 6).

Application element	AIM element	Source	Rules	Reference path
performance_ hall to target_ audience_ capacity (has)	PATH	41		product <- product_definition_formation.of_product product_definition_formation <- product_definition.formation product_definition

Table 6. Mapping table for example AP 1 use of product definition construct from product definition schema.

The assertion performance hall has target audience capacity is interpreted as a PATH that involves product, product definition formation, and product definition from the product definition schema. There is no reference path constraint on the name of the product definition for example AP 1.

The schema of the AIM for AP 1 uses the product and product definition entities from the product definition schema. Since AP 1 stated no requirements for aggregations of product definitions, product definition formation is not called out explicitly in the USE FROM statement. The entity is available to satisfy the path mapping since a product definition formation is referenced by a mandatory attribute of product definition.

```
SCHEMA  ap1_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

USE FROM product_definition_schema (product, product_definition);

END SCHEMA;
```

Representing the target audience capacity as a property of a life cycle dependent product definition requires not only the product definition entity but also requires constructs from the product property definition schema, the product property representation schema, the representation schema, the measure qualification schema, and the measure schema.

Product property definition

The product property definition schema contains constructs that are used to represent the semantics of an AP domain ontology with respect to the identification of property definitions that are aggregated to form a life cycle dependent product definition (Fig. 18). For purposes of this discussion, a property definition references a product definition through two select types (see Annex B). Any number of property definitions may be used to form a characterized product definition.

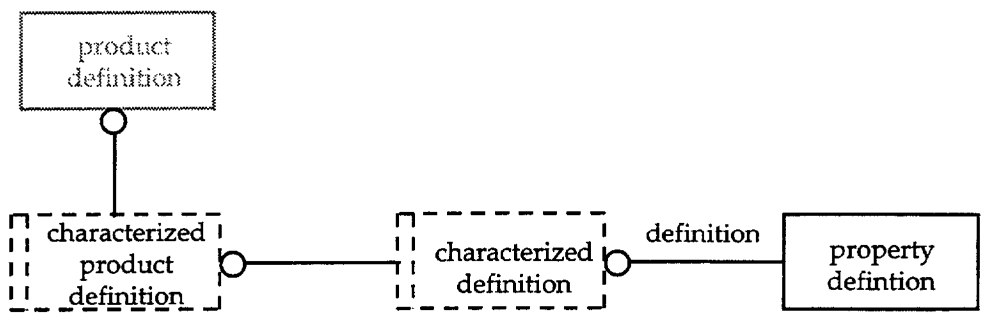


Figure 18. Elements of the product_property_definition_schema.

product_property_definition_schema:
A representation of the semantics of AP domain ontologies with respect to the existence of a defined property that is used for characterization.

property_definition
An identification of the declared existence of a property that may be use to characterize a life cycle dependent product definition (i.e., it may be a member of the aggregation of properties that comprise a product definition).

name
A label that is human interpretable for the defined property.

description
Text used to designate the nature of the defined property.

definition
Every property_definition references exactly one characterized_definition as definition

characterized_definition
A selection of a characterized product definition (or another resource object not in the scope of this discussion).
selection (implicit)
Every property_definition references exactly one characterized_prouct_definition as definition

characterized_product_definition
A selection of a product definition (or another resource object not in the scope of this discussion).
selection (implicit)
Every property_definition references exactly one characterized_prouct_definition as selection

The assertion performance hall has target audience capacity from example AP 1 is interpreted as a PATH using constructs from the product property definition schema (Table 7). The name of property definition is constrained to be audience capacity and its description is constrained to be viability criterion. The property definition references characterized definition which selects characterized product definition which selects a product definition that has the name as-required hall.

Application element	AIM element	Source	Rules	Reference path
performance_ hall to audience_ capacity (as target)	PATH	41		*product <-* *product_definition_formation.of_product* *product_definition_formation <-* *product_definition.formation* *product_definition* characterized_product_definition= product_definition characterized_product_definition characterized_definition= characterized_product_definition characterized_definition <- property_definition.definition property_definition <- {property_definition.name= 'audience capacity' property_definition.description= 'viability criterion'}

Table 7. Mapping table for example AP 1 use of product property definition schema.

This interpretation is a result of a semantic analysis of the application object definition that describes target audience capacity as a way of thinking about audience capacity that identifies the minimum number of persons required for economic viability.

The AIM of example AP 1 will use property definition.

```
SCHEMA ap1_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

USE FROM product_definition_schema (product, product_definition);

USE FROM product_property_definition_schema (property_definition);

END SCHEMA;
```

Product property representation

The product property representation schema contains constructs that are used to represent the semantics of an AP domain ontology with respect to the identification of an association between a product property definition and a representation of that property definition (Fig. 19). Any number of such associations are possible so there may be zero, one, or more representations for the property.

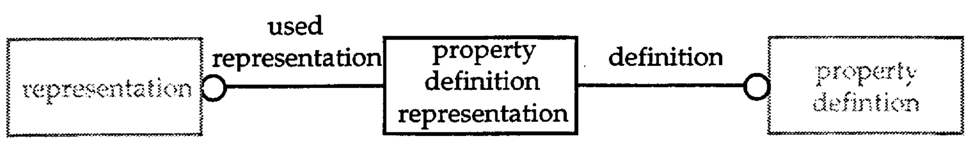


Figure 19. Elements of the product_property_representation_schema.

product_property_representaiton_schema:
A representation of the semantics of AP domain ontologies with respect to the existence of an association between a property definition and any number of representations of the defined property.

property_definition_representation
An identification of an association between a property definition and a representation used for that property.

definition
Every property_definition_representation references exactly one property_definition as definition
used_representation
Every property_definition_representation references exactly one representation as used_representation

The assertion performance hall has target audience capacity is interpreted as a PATH that also uses constructs from the product property representation schema (Table 8).

Application element	AIM element	Source	Rules	Reference path
performance_ hall to target_ audience_ capacity (has)	PATH	41		*product* <- *product_definition_formation.of_product* *product_definition_formation* <- *product_definition.formation* *product_definition* *characterized_product_definition=* *product_definition* *characterized_product_definition* *characterized_definition=* *characterized_product_definition* *characterized_definition* <- *property_definition.definition* *property_definition* <- *{property_definition.name=* *'audience capacity'* property_definition.description= 'viability criterion'} property_definition_representation.definition property_definition_representation property_definition_representation. used_representation -> representation

Table 8. Mapping table for example AP 1 use of product property representation schema.

The property definition representation construct has a definition reference to a property definition with a name audience capacity. It also has a used representation reference to a representation (from the representation schema).

The AIM of example AP 1 uses property definition representation.

```
SCHEMA ap1_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

USE FROM product_definition_schema (product, product_definition);

USE FROM product_property_definition_schema (property_definition);

USE FROM product_property_representation_schema (property_definition_representation);

END SCHEMA;
```

Representation

The representation schema contains constructs that are used to capture the semantics of an AP domain ontology with respect to the identification of an association between a representation context and one or more representation items that are used to represent a property definition (Fig. 20).

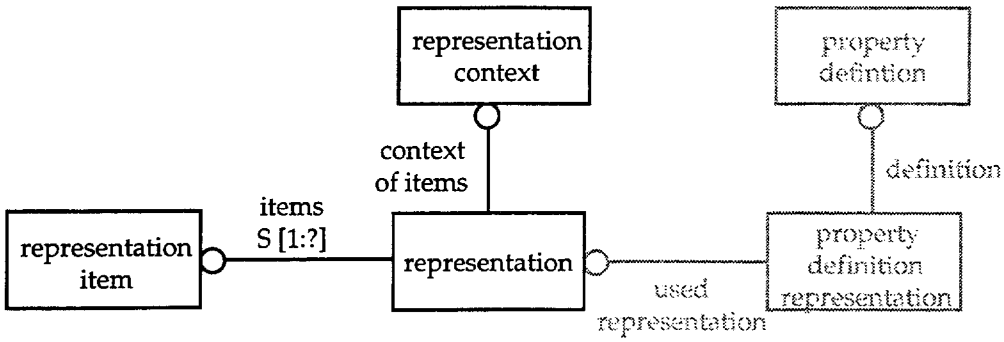


Figure 20. Elements of the representation_schema.

representation_schema:

A representation of the semantics of AP domain ontologies with respect to the existence of a representation that may be associated with a property definition.

representation

An identification of an aggregation of representation items that are related to one another by a common representation context used to represent something (e.g., a property definition).

name

A label that is human interpretable for the defined association.

context_of_items

Every representation references exactly one representation_context as context_of_items

items

Every representation references at least one representation_item as items

representation_context

An identification of a common condition or circumstance for an aggregation of representation items.

context_identifier

An identifier for the common condition or circumstances for an aggregation of representation items.

context_type

A textual description of the kind of common condition or circumstances that serve as the context of a representation.

representation_item

An identification of an element that stands for something else (e.g., a property definition).

name

A label that is human interpretable for an element that stands for something else.

The assertion performance hall has target audience capacity from example AP 1 is interpreted as a PATH that also uses the representation construct from the representation schema (Table 9).

Application element	AIM element	Source	Rules	Reference path
performance_ hall to target_ audience_ capacity (has)	PATH	41		*product <-* *product_definition_formation.of_product* *product_definition_formation <-* *product_definition.formation* *product_definition* *characterized_product_definition=* *product_definition* *characterized_product_definition* *characterized_definition=* *characterized_product_definition* *characterized_definition <-* *property_definition.definition* *property_definition <-* *{property_definition.name=* *'audience capacity'* property_definition.description= 'viability criterion'*}* *property_definition_representation.definition* *property_definition_representation* *property_definition_representation.* *used_representation ->* *representation* {representation.name='minimum seated and standing audience capacity'}
Target_ audience_ capacity	representation	41	41	{representation representation.name=' minimum seated and standing audience capacity'}

Table 9. Mapping table for example AP 1 use of the representation construct from the representation schema.

The application object target audience capacity is interpreted as a representation with minimum seated and standing audience capacity as its name. Others representations of audience capacity are possible. The representation referenced by property definition representation references a representation with this name.

The AIM of example AP 1 will use representation by declaration and representation context and representation item implicitly to satisfy mandatory references.

```
SCHEMA apl_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

USE FROM product_definition_schema (product, product_definition);

USE FROM product_poperty_definition_schema (property_definition);

USE FROM product_poperty_representation_schema (property_definition_representation);

USE FROM representation_schema (representation);

END SCHEMA;
```

Measure

The measure schema contains the construct measure with unit (Fig. 21). It has a value component and a unit component. The unit component can be a context dependent named unit with appropriate dimensions. The value component can be a count measure.

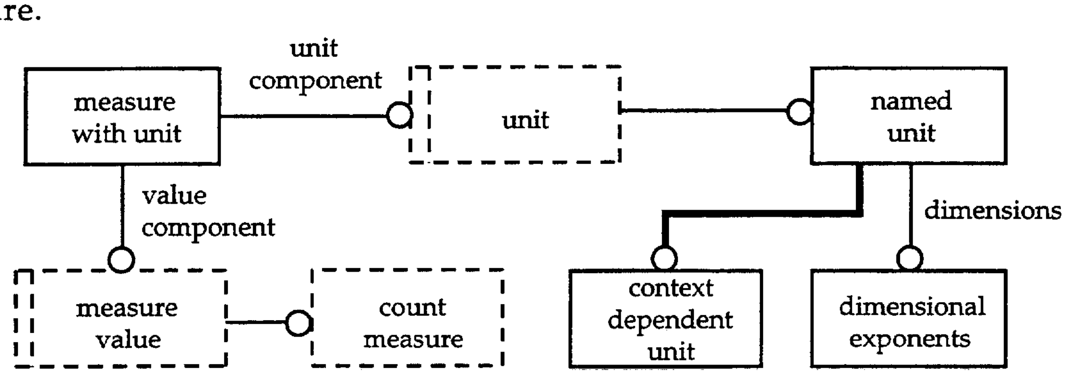


Figure 21. Elements of the product_property_representation_schema.

measure_schema:
A representation of the semantics of AP domain ontologies with respect to the specification of a physical value.

measure_with_unit
An identification of an association between a value component and a unit component used to quantify the extent of something.

value_component
Every measure_with_unit references exactly one measure_value as value_component
unit_component
Every measure_with_unit references exactly one unit as unit_component

named_unit
An identification of a unit that is named.

dimensions
Every named_unit references exactly one dimensional_exponents as dimensions

dimensional_exponents
A specification of the dimensionality of a quantity in terms of the values of seven base quantities as powers.

length_exponent
The power of the length base quantity.
mass_exponent
The power of the mass base quantity.
time_exponent
The power of the time base quantity.
electric_current_exponent
The power of the electric current base quantity.
thermodynamic_termperature_exponent
The power of the thermodynamic temperature base quantity.
amount_of_substance_exponent
The power of the amount of substance base quantity.
luminous_intensity_exponent
The power of the luminous intensity base quantity.

AP information requirements established the need for a construct that is a representation item and a measure with unit called a measure representation item. Measure representation item is an extension to the IRs that appears in the qualified measure schema [14]. It can be used to represent properties like the target audience capacity of example AP 1. The measure with unit is of type measure representation item which is also a subtype of representation item. Two measure representation items are used.

The reference paths indicate that a representation with a name minimum seated and standing audience capacity has measure representation items that include one called minimum seating capacity and another called minimum standing capacity. The application elements seating capacity and standing capacity are each mapped to a count measure (an integer) for the value component of a measure with unit through a measure representation item (Table. 10).

Application element	AIM element	Source	R	Reference path
seating_ capacity	measure_ with_ unit.value_ component	41		representation representation.items[i] -> {representation_item.name= ' minimum seating capacity'} representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.value_component= count_measure}
standing_ capacity	measure_ with_ unit.value_ component	41		representation representation.items[i] -> {representation_item.name= 'minimum standing capacity'} representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.value_component= count_measure}

Table 10. Mapping table for example AP 1 use of the value component of a measure with unit from the measure schema.

An application element number of persons is mapped to measure with unit in the measure schema (Table 11). Persons is the name of a context dependent unit which is the unit component of measure with unit.

Application element	AIM element	Source	Rules	Reference path
persons	measure_ with_ unit. unit_ component	41		representation ({representation_item.name= 'minimum seating capacity'}) {representation_item.name= 'minimum standing capacity'}) representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.unit_component= context_dependent_unit context_dependent_unit.name='persons'}

Table 11. Mapping table for example AP 1 use of the unit component of a measure with unit from the measure schema.

The AIM of example AP 1 will use measure with unit, count measure, and context dependent unit from the measure schema. It will also use measure representation item from the measure qualification schema.

```
SCHEMA ap1_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

USE FROM product_definition_schema (product, product_definition);

USE FROM product_property_definition_schema (property_definition);

USE FROM product_property_representation_schema (property_definition_representation);

USE FROM representation_schema (representation, representation_item);

USE FROM measure_schema (measure_with_unit, count_measure, context_dependent_measure);

USE FROM measure_qualification_schema (measure_representation_item);

END SCHEMA;
```

The interpretation of the information requirements results in both an AIM and a representation mapping (i.e., mapping tables). The representation mapping contains reference path constraints that prescribe values for attributes of entities used to represent the application context and semantics of the AP domain ontology. The increased complexity of the AIM compared to that of the domain ontology description is a result of specifying semantics that were implicit in the user description but that are explicitly represented using the IRs.

The principal constructs of the IRs illustrated in example AP 1 include (see Fig. 22):

- application context schema
  - application protocol definition
  - application context
  - application context element
  - product context
  - product definition context
- product definition schema
  - product
  - product related product category
  - product category
  - product category relationship
  - product definition formation
  - product definition
- product property definition schema
  - property definition
- product property representation schema
  - property definition representation
- representation schema
  - representation
  - representation context
  - representation item
- measure schema
  - measure with unit
  - named unit
  - context dependent unit
  - dimensional exponents
- measure qualification schema
  - measure representation item

2.3 Specifying the conformance classes of a domain ontology

The conformance classes are described as part of the AP domain ontology description. The requirements of the conformance class are specified by identifying the application objects that are needed to accommodate particular usage scenarios. Those portions of the mapping tables and AIM that correspond to the identified application objects comprise the formal specification of the conformance classes. Once elements of the mapping tables and AIM have been identified, other elements that are required to satisfy mandatory references are also included.

For example AP 1, the two application objects are part of a single conformance class. Therefore, all of the mapping tables and the entire AIM together comprise the specification of the AP 1 conformance class. The mapping tables for AP 1 include mappings for the context of the domain ontology (Table 12), for the semantics of the product addressed by the domain ontology (Table 13), for the semantics of the life cycle dependent product definition of the domain ontology (Table 14) and for the semantics of the representation of life cycle dependent property definitions (Table 15). Each of these mapping tables prescribe attribute value constraints on entities used from the IRs to represent the context and semantics of the AP domain ontology.

Application element	AIM element	Source	Rules	Reference path
AP	application_ protocol_ definition	41		{application_protocol_definition application_protocol_definition. application_interpreted_model_ schema_name= 'ap1_aim_schema'}
Functionality / technology	application_ context	41		{application_context application_context.application= 'economic viability analysis'}
Industry sector / discipline	product_context	41		{application_context_element <- [application_context_element application_context_element.name= 'building industry'] product_context product_context.discipline_type= 'economics'}
Type of product perspective / life cycle stage	product_ definition_context	41		{application_context_element <- [application_context_element application_context_element.name= 'as-required'] product_definition_context product_context.life_cycle_stage= 'specify economic design requirements'}

Table 12. Mapping table for the context of the AP 1 domain ontology.

Application element	AIM element	Source	Rules	Reference path
Performance_ hall	product	41		{product <- product_related_product_category.products[i] product_related_product_category <= (product_category product_category.name product_category.name = 'performance hall') (product_category <- product_category_relationship.sub_category product_category_relationship product_category_relationship.category product_category.name = 'performance hall')}
id	product.id	41		product product.id
name	product.name	41		product product.name

Table 13. Mapping table for product semantics of the AP 1 domain ontology.

Application element	AIM element	Source	Rules	Reference path
performance_ hall to target_ audience_ capacity (has)	PATH	41		product <- product_definition_formation.of_product product_definition_formation <- product_definition.formation product_definition characterized_product_definition= product_definition characterized_product_definition characterized_definition= characterized_product_definition characterized_definition <- property_definition.definition property_definition <- {property_definition.name= 'audience capacity' property_definition.description= 'viability criterion'} property_definition_representation. definition property_definition_representation property_definition_representation. used_representation -> representation {representation.name='minimum seated and standing audience capacity'}

Table 14. Mapping table for product definition semantics of the AP 1 domain ontology.

Application element	AIM element	Source	R	Reference path
Target_ audience_ capacity	representation	41		{representation representation.name=' minimum seated and standing audience capacity'}
seating_ capacity	measure_ with_ unit.value_ component	41		representation representation.items[i] -> {representation_item.name= ' minimum seating capacity'} representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.value_component= count_measure}
standing_ capacity	measure_ with_ unit.value_ component	41		representation representation.items[i] -> {representation_item.name= 'minimum standing capacity'} representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.value_component= count_measure}
persons	measure_ with_ unit.unit_ component	41		representation representation.items[i] -> (representation_item.name= 'minimum seating capacity'}) { representation_item.name= 'minimumstanding capacity'}) representation_item => measure_representation_item <= measure_with_unit {measure_with_unit.unit_component= context_dependent_unit context_dependent_unit.name='persons'}

Table 15. Mapping table for representation semantics of the AP 1 domain ontology.

The AIM of example AP 1 specifies the use of the constructs from the IRs as identified by the mapping tables. The EXPRESS-G presentation of AIM assertions is useful in visualizing the structure and functional dependencies of the AIM (Fig. 22).

Central to the STEP representation of AP domain ontologies is the distinction between the identification of the existence of a product as relevant to multiple technical disciplines, and the identification of life cycle dependent product definitions that are aggregates of properties relevant to a limited number of perspectives (i.e., domain ontologies). Any number of product definitions may be developed for a given identified product. These definitions may span the entire life cycle of a product. In STEP, life cycle dependent product definitions are distributed among interrelated APs, as needed. Communication takes place within the context of one or more APs that have an identified need to include a given perspective.

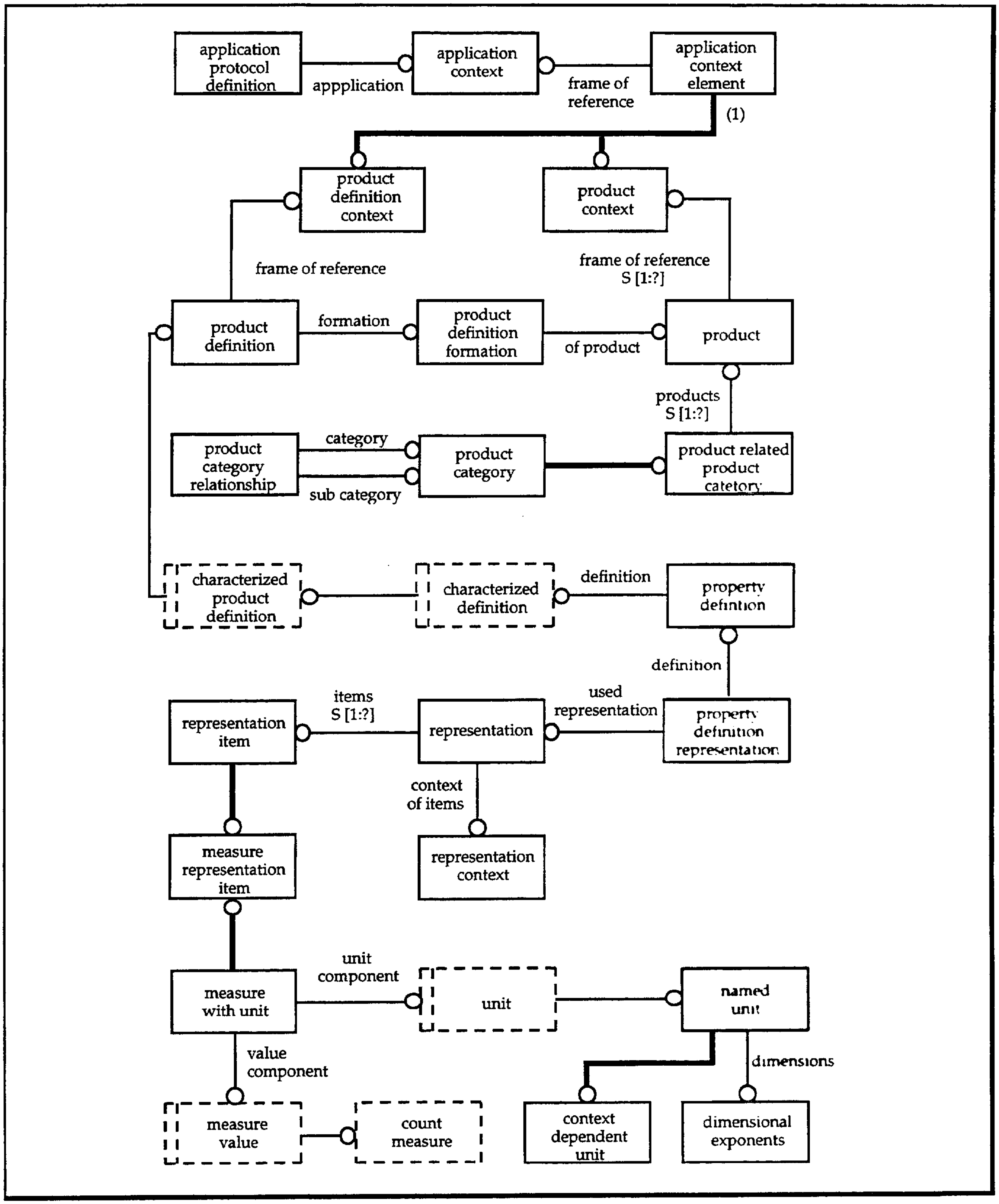


Figure 22. Graphical presentation of example AP 1 AIM assertions.

3. Using conformance classes for communication

A conformance class identifies the application objects of an AP domain ontology that must be supported in order to satisfy an information usage scenario. A usage scenario identifies communication events among activities within a specific application context. The application objects identified in a conformance class of an AP are interpreted using standard semantic representation constructs. Interpretation results in mapping tables and an AIM. The mapping tables identify a correspondence between application objects and elements of the AIM. They also define reference paths that specify assertions among AIM elements that are used to represent both explicit and implicit semantics of the AP information usage requirements. Reference path constraints specify specific attribute values that ensure semantic integrity within the application context of the AP.

Example AP 1 contains two application objects in its single conformance class. They are the performance hall and the target audience capacity. The elements of the mapping tables (Tables 12-14) and AIM (Fig. 23) are used to represent these application objects to define the conformance class of the AP.

```
SCHEMA ap1_aim_schema;

USE FROM application_context_schema (application_protocol_definition, application_context,
product_context, product_definition_context);

USE FROM product_definition_schema (product, product_definition);

USE FROM product_poperty_definition_schema (property_definition);

USE FROM product_poperty_representation_schema (property_definition_representation);

USE FROM representation_schema (representation, representation_item);

USE FROM measure_schema (measure_with_unit, count_measure, context_dependent_measure);

USE FROM measure_qualification_schema (measure_representation_item);

END SCHEMA;
```

Figure 23. AIM for example AP 1.

The mapping tables and the AIM are the basis for the communication of information using STEP. They may also be used as the basis for storing and retrieving information in such a way that the semantic integrity specified for communication is maintained. Example AP 1 is used to illustrate how the mapping tables and AIM can be used for both communication and storing/retrieving information.

3.1 Communicating information

Information can be communicated using the format for clear text encoding specified by STEP [14]. The simplest form of communication involves the transfer of an information state (i.e., a snapshot at the time of transfer) consistent with some portion of a conformance class without control over what is done with the information by either the sender or receiver subsequent to transfer (Fig. 24).

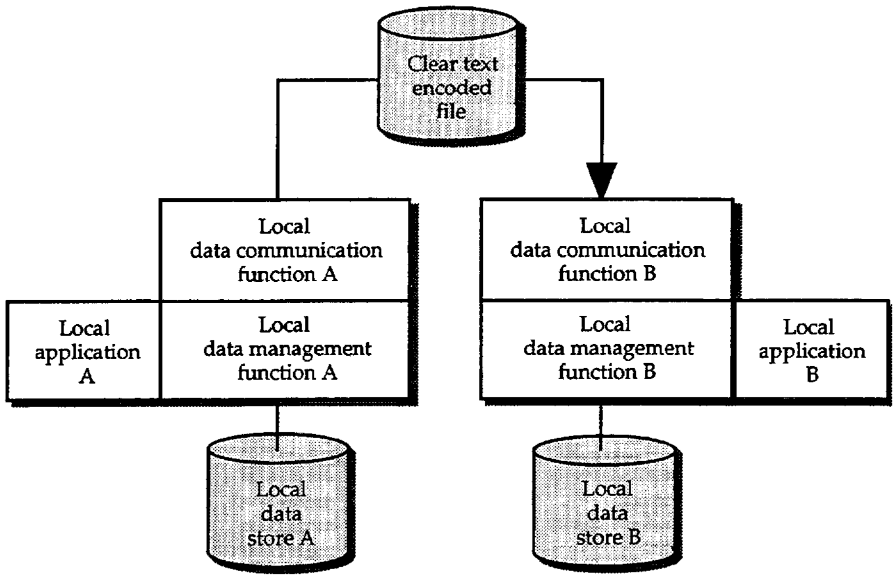
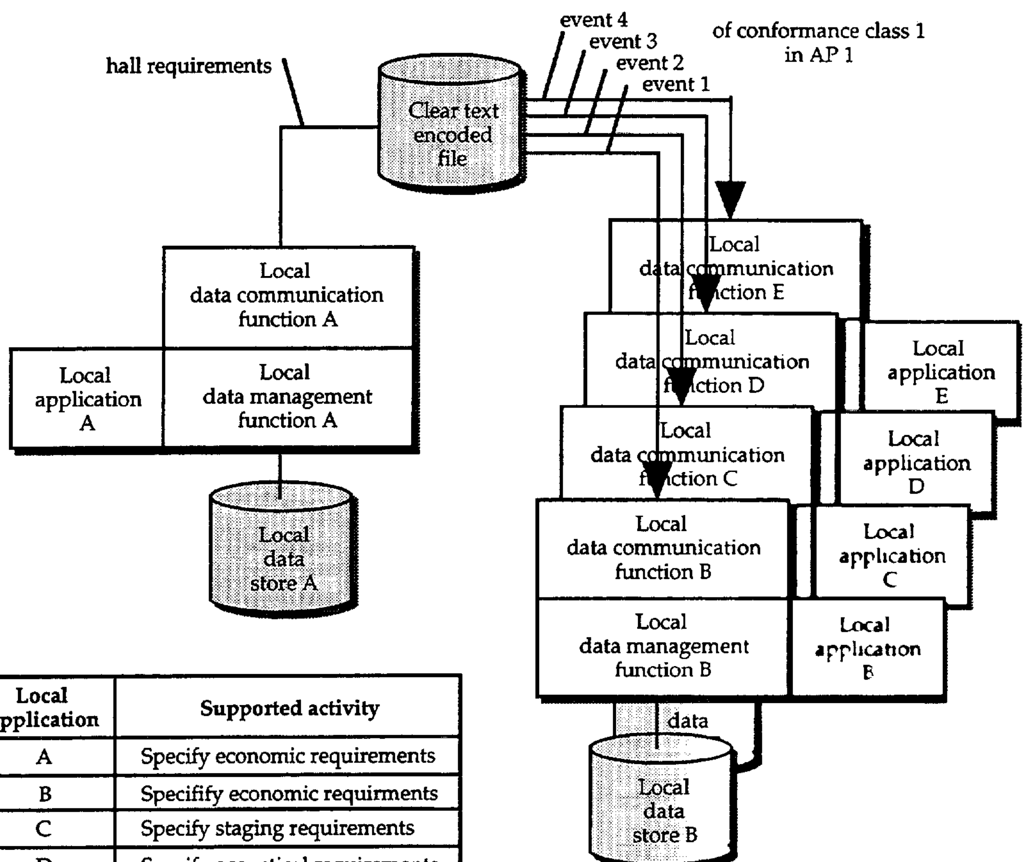


Figure 24. Transfering an information state.

A local data management function A retrieves information from the local data store A and sends it to the local data communication function A. Local communication function A encodes the information as a clear text file consistent with a conformance class of a STEP AP and sends the file to local data communication function B. The file is decoded consistent with a conformance class of a STEP AP and sends the information to local data management function B. Local data management function B stores the information in local data store B.

The local data communication functions, A and B, must use conformance classes of APs that have the information to be exchanged within their scopes. Both local data communication functions A and B can use the same conformance class from the same AP. They can also use conformance classes from different APs if each has within its scope the information that is to be communicated. This common information will use one or more AICs. It also involves common reference path constraints such that the use of the AIM data structures are the same. Two APs for the same or different disciplines (i.e., the product context may involve more than one discipline) that understand a minimum seated and standing audience capacity as created by an application dealing with economic viability analysis can communicate. This communication requires an understanding by each AP of the semantics of the entire reference path from the measure representation items to the application context.

The conformance class of example AP 1 supports a usage scenario that accommodates four events (Fig. 25). Event one is the transfer of hall economic requirements from users involved in specifying economic requirments to other users involved in the same activity. Event two involves the transfer of hall economic requirements from users involved in specifying economic requirements to users involved in specifying staging requirements. Event three involves the transfer of hall economic requirements from users involved in specifying economic requirements to users involved in specifying acoustical requirements. Event four involves the transfer of hall economic requirements from users involved in specifying economic requirements to users involved in developing a design specification.



Local application	Supported activity
A	Specify economic requirements
B	Specifify economic requirments
C	Specify staging requirements
D	Specify acoustical requirements
E	Develop design specification

Figure 25. Transfer of hall requirements for four types of events of a conformance class.

Each of these four event types may be accommodated using local data communication functions at both the sending and receiving ends that employ the same conformance class of example AP 1. A sample data population for such a communication contains data that is in part specified by the reference path constaints of AP 1 and in part by the user defined data (Fig. 26).

```
DATA;
#1 =application_context('economic viability analysis ');
#2 =application_protocol_defintion('wd','ap1_aim_schema ',1996,#1);
#3 =(application_context_element('building industry ',#1)
    product_context('economics '));
#4 =(application_context_element('as-required ',#1)
    product_definition_context('specify economic design requirements '));
#5 =product( 'NIST Blue Auditorium','Room 1 Building 2002',
    'an example product'(#3));
#6 =(product_category('performance hall ','a performance space ')
    product_related_product_category((#5));
#7 =product_definition_formation('v 1.0.0','initial version',#5);
#8 =product_definition('economic requirement definition 1','by BFRL',#7,#4);
#9 =property_definition('audience capacity ','viability criterion ',#8);
#10 =representation_context('audience capacity context 1','valued ');
#11 =representation('minimum seated / standing audience capacity ',#10,(#12,#13));
#12 =(representation_item('minimum seating capacity ') measure_with_unit(3000,#16)
    measure_representation_item());
#13 =(representation_item('minimum standing capacity ') measure_with_unit(200,#16)
    measure_representation_item());
#14 =property_definition_representation(#9,#11);
#15 =dimensional_exponents(0,0,0,0,0,1,0 );
#16 =(named_unit(#15) context_dependent_unit( 'persons '))
ENDSEC;
```

Figure 26. Physical file formatted sample data for example AP 1.

The reference path and reference path constraints of the mapping tables specify how the representation of the domain ontology (i.e., the AIM) is to be used if the intended usage is to comply with a conformance class of a STEP AP. The data predefined by the reference path constraints are presented in italics. The data defined by the user is presented in bold. In this example, 15 elements are predefined by the conformance class of example AP 1 while 12 elements are defined by the user. The confomance class of an AP provides the semantics, context, and to a considerable degree, much of the data that ensures both the semantic integrity and the utility of a communication.

Communicating part of the information covered by a conformance class

The IRs, and therefore the AIMs that use the IRs, contain explicit existence dependency constraints in the resource assertions among resource objects. These constraints appear as mandatory attributes in one entity that creates a dependency upon the existence of another entity. In the STEP IRs, all information is (i.e., in terms of instances) dependent upon an application context (Fig. 22). As an example, a product must have a product context which is an application context element which must have a frame of reference which is an application context. A product is not, however, dependent upon the existence of a product definition formation which is in turn not dependent on the existence of a product definition.

A product definition must be of a particular product definition formation which must be of a particular product. The product is dependent upon application context as described. The product definition is also dependent upon a product definition context that is an application context element which must have a frame of reference which is an application context. A product definition is not dependent upon any product property definition.

The existence dependency constraints explicit in the resource assertions among resource objects ensures the semantic integrity of any available information. It does enable the communication of partial information covered by a conformance class. A given conformance class of a particular AP can be used to communicate the development of information about a product over time.

In the use of the example AP 1, local applications A and B (Fig. 25) can communicate in accordance with the event 1 usage scenario of conformance class 1. Both applications are working at the specify economic requirements phase of the life cycle. The existence dependency of the AIM allows application A to communicate available information about the performance hall upon which they will be working jointly.

```
DATA;
#1 =application_context('economic viability analysis ');
#2 =application_protocol_defintion('wd','ap1_aim_schema ',1996,#1);
#3 =(application_context_element('building industry ',#1)
    product_context('economics '));
#4 =(application_context_element('as-required ',#1)
    product_definition_context('specify economic design requirements '));
#5 =product( 'NIST Blue Auditorium','Room 1 Building 2002',
    'an example product'(#3));
#6 =(product_category('performance hall ','a performance space ')
    product_related_product_category((#5));
ENDSEC;
```

Subsequently, application B can further develop information about the performance hall's target audience capacity (from the end user's point of view) by communication consistent with the reference path and its constraints for the target audience capacity (i.e., as illustrated in the clear text encoding on page 46).

Using a conformance class for communication of information in this way requires management of information accross applications. The information contained within the sending system is independent of how it is contained and used in the receiving system. The instance identifiers (#1-#6 in the first communication and #1-#16 in the second) of the clear text encoding provide for the integrity of the information. The instance identifiers of the clear text encoding format resolve references within each file, but need not be used by the sending and receiving systems before or after the information is communicated. Data management of information across and among activities requires a coordinated approach to instance identification and control by a global data management function [15].

3.2 Managing communicated information globally

The primary requirement for managing communicated information globally, is that instances are unambiguously identified. This is a first necessary step toward an integrated application context environment. It requires that there exist a global data management function that is responsible for assigning and maintaining instance identifiers in cooperation with local data management functions (Fig. 27).

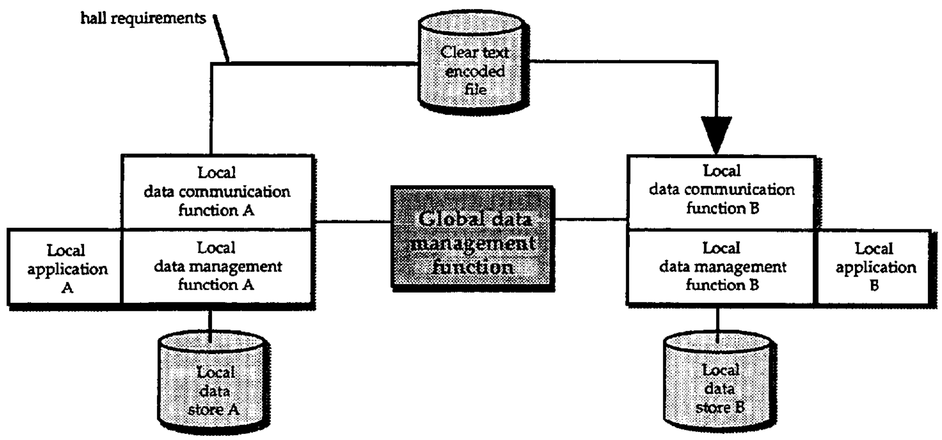


Figure 27. A global data management function among communicating systems.

Each entity instance in a population consistent with an AIM (or any EXPRESS schema) is assumed to have a unique object identifier (OID) in addition to values for the explicitly declared attributes.[8] It is left to implementing systems to provide the means by which unique identification of instances is achieved.[9]

Clear text encoding provides a means of identifying instances and resolving references among instances consistent with an identified AIM. It does not, however, provide a mechanism for associating the internally consistent instance identification of instances with mechanisms used by either a sending or receiving system. Global control of instance identification (i.e., involving two or more systems) requires either that the instance identifiers of the clear text encoding be used either directly or indirectly, or that unique instance identifiers be added (e.g., as the first attribute before explicitly declared attributes) for every entity in an AIM.

In the case of using the clear text encoding identifiers, several forms are possible including sequential number generation as instances are created (as in Fig. 26). For comparison with tables of a relational database in the section on storing information, the normalized element numbers of an AIM multiplied by 1000 are used (Fig 27).

---

[8] EXPRESS does not use the values of declared attributes to identify instances unambiguously. Rather, it assumes the use of unique instance identifiers in any instantiation of an EXPRESS schema. EXPRESS neither specifies the content nor the representation of these identifiers.

[9] This may involve keys that use explicitly identified unique attributes in an AIM rather than OIDs.

The first instance of element 1 is 1001 (presented in bold). The sequential approach is the more feasible in actual systems (though less readable).

```
DATA;
#1001 =application_context('economic viability analysis');
#2001 =application_protocol_defintion('wd','ap1_aim_schema',1996,#1001);
#4001 =(application_context_element('building industry',#1001)
    product_context('economics'));
#5001 =(application_context_element('as-required',#1001)
    product_definition_context('specify economic design requirements'));
#6001 =product('NIST Blue Auditorium','Room 1 Building 2002',
    'an example product',(#4001));
#8001 =(product_category('performance hall','a performance space')
     product_related_product_category((#6001));
#12001 =product_definition_formation('v 1.0.0','initial version',#6001);
#13001 =product_definition('economic requirement definition 1','by BFRL',#12001,#5001);
#17001 =property_definition('audience capacity','viability criterion',#13001);
#18001 =representation_context('audience capacity context 1','valued');
#19001 =(representation_item('minimum seated capacity') measure_with_unit(3000,#27001)
     measure_representation_item());
#19002 =(representation_item('minimum standing capacity') measure_with_unit(200,#27001)
     measure_representation_item());
#20001 =representation('minimum seated and standing audience capacity', #18001,
    (#19001,#190002));
#22001 =property_definition_representation(#17001,#20001);
#25001 =dimensional_exponents(0,0,0,0,0,1,0);
#27001 =(named_unit(#25001) context_dependent_unit('persons'))
ENDSEC;
```

Figure 28. Physical file formatted sample data for example AP 1 managing instances through selection of clear text encoding numbers.

Employing clear text encoding instance numbers to identify instances has both advantages and disadvantages. The most obvious advantage is that it requires no change to any part of STEP. It is an approach that may be taken by any collection of participants to provide common instance identifiers among those participants. A principal disadvantage is that it may not provide identifiers for all logical elements of a database. Accommodation may be required, depending upon the implementation, for complex instances, select types, defined types, and references to aggregations.

Any scheme that employs clear text encoding instance numbers requires all participants that deal with a given product over its entire life cycle to coordinate the issuance of an unambiguous number at the time an instance is created. These numbers can be used to create unique identifiers in terms of a global data management function (i.e., globally unique with respect to all systems using that function). Such a data management function can be established for the purpose of managing information about one or more products over the entire life cycle of the products.

A global data management function requires an agreement about the use of OIDs among participants using the function. It does not, however, ensure control of other aspects of instance management beyond identification. Data management involves such issues as data redundancy control (e.g., controling copies of an instance) and data integrity control among all participating systems (e.g., contolling creation, update, and modification of data). Maintaining a correspondence between the identification of instances on systems and during exchange among systems provides a necessary (but not sufficient) condition for sharing information since sharing requires all of these data management functions.

Using object identifiers (OIDs) to identify instances

Every entity within any EXPRESS schema is assumed to have an OID used for instance identification. The implicit OID of an AIM can be assumed to be the first attribute of any defined entity (Fig 29). All systems using a particular global data management function would maintain these instance identifiers.

```
DATA;
#1 =application_context(CIC1-1-1,'economic viability analysis');
#2 =application_protocol_defintion(CIC1-2-1,'wd','ap1_aim_schema',1996,#1);
#3 =(application_context_element(CIC1-3-1,'building industry',#1)
    product_context(CIC1-4-1,'economics'));
#4 =(application_context_element(CIC1-3-2,'as-required',#1)
    product_definition_context(CIC1-5-1,'specify economic design requirements'));
#5 =product(CIC1-6-1, 'NIST Blue Auditorium','Room 1 Building 2002',
    'an example product'(#3));
#6 =(product_category(CIC1-8-1,'performance hall','a performance space')
    product_related_product_category(CIC1-10-1, (#5));
#7 =product_definition_formation(CIC1-12-1,'v 1.0.0','initial version',#5);
#8 =product_definition(CIC1-13-1,'economic requirement definition 1','by BFRL',#7,#4);
#9 =property_definition(CIC1-17-1,'audience capacity','viability criterion',#8);
#10 =representation_context(CIC1-18-1,'audience capacity context 1','valued');
#11 =representation(CIC1-19-1,'minimum seated and standing audience capacity',#10,(#12,#13));
#12 =(representation_item(CIC1-20-1,'minimum seating capacity')
    measure_with_unit(CIC1-29-1,3000,#16) measure_representation_item(CIC1-30-1));
#13 =(representation_item(CIC1-20-2,'minimum standing capacity')
    measure_with_unit(CIC1-29-2,200,#16) measure_representation_item(CIC1-30-2));
#14 =property_definition_representation(CIC1-220-1,#9,#11);
#15 =dimensional_exponents(CIC1-25-1,0,0,0,0,0,1,0);
#16 =(named_unit(CIC1-27-1,#15) context_dependent_unit(CIC1-26-1, 'persons'))
ENDSEC;
```

Figure 29. Physical file formatted sample data for example AP 1 managing instances through addition of OIDs.

3.3 Storing and retrieving information

In order to maintain the information contained within a conformance class of an AP, an implementation design activity (Fig. 30) can be used to develop a logical data structure that supports the semantics contained within the AIM and mapping tables. For purposes of illustration, a relational database is assumed.

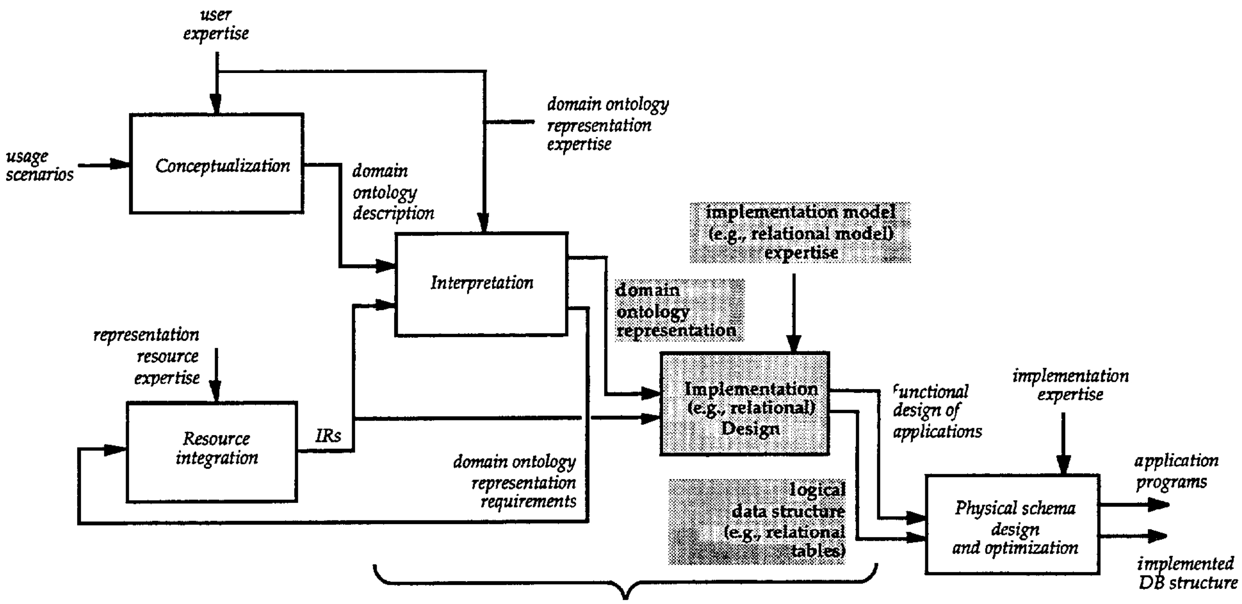


Figure 30. Developing a logical data structure for information storage.

Three data specification activities provide input to the implementation design activity. They are, conceptualization (i.e., developing a domain ontology description), interpretation (i.e., developing a domain ontology representation), and resource integration (i.e., developing ontology representation resources).[10] A domain ontology representation as well as the IRs are inputs to the implementation design process.

The implementation design activity is controled by implementation model expertise. For the purposes of this discussion, the relational data model is assumed. The output of the activity is a logical data structure, in the assumed case, a logical model used to identify relational tables (i.e., relations).

A relational implementation design [16,17] produces a normalized logical data structure of relations derived from the AIM. It is constrained by the rules of the AIM and reference path constraints of the mapping tables. The simplest approach for illustration purposes is to create a table for each normalized logical element of the AIM.[11]

---

[10] Customarily, activities are labeled using verbs. This activity model uses nouns (e.g., conceptualization, resource integration, and interpretation) since they correspond to common usage.

[11] A normalized element (e.g., a relation in third normal form) requires attributes to be atomic, mutually independent, and fully dependent on the element instance identifier.

The logical elements of the AIM for example AP 1 are numbered from one to thirty (Fig. 31).[12] Assumptions have been made with respect to what constitutes a logical element and therefore a relational table in the example database. This is reflected in the numbering of the logical elements and the highlighted changes in the data structure.

1. Each entity is considered a logical element (e.g., element 1).

2. A supertype and subtype are considered as separate logical elements with an "is-a" relationship between them.

   Application_context_element is a product_context OR product_definition_context
   / product_context is a(n) application_context_element
   /product_definition_context is a(n) application_context_element
   Every application_context_element is exactly one product_context OR exactly one product_defintion_context and
   Every product_context is exactly one application_context_element and
   Every product_definition_context is exactly one application_context_element

   Measure_representation_item is a representation_item AND measure_with_unit
   / representation_item may be a(n) measure_representation_item
   / measure_with_unit may be a(n) measure_representation_item
   Every measure_representation_item is exactly one representation_item AND exactly one measure_with_unit and
   Each representation_item may be exactly one measure_representation_item and
   Each measure_with_unit may be exactly one measure_representation_item

   Named unit may be a contextext_dependent_unit
   / context_dependent_unit is a named_unit
   Every named_unit is at most one context_dependent_unit and
   Every context_dependent_unit is exactly one named_unit

   An attribute that is an aggregation (e.g., SET) is considered a logical element (e.g., element 7 product.frame_of_reference becomes an entity product_frame_of_reference).

   product_frame_of_reference
   context
   Every product_frame_of_reference references exactly one product_context as context
   of_product
   Every product_frame_of_reference references exactly one product as of product

3. A select type is considered a logical element with an optional attribute with the name of the selected element appended to the word selection (e.g., element 15 characterized_product_definition.product_definition_selection). A rule is added to the element stating that exactly one of the optional attributes must reference an item (e.g., element 15 must reference a product definition since it is the only possible selection).

[12] The number 14 is not used. This number could be used by an additional associative entity product definition context relationship if a product definition were to be related to more than one context (e.g., as an output of one and a control of another) rather than only the context in which the information is created.

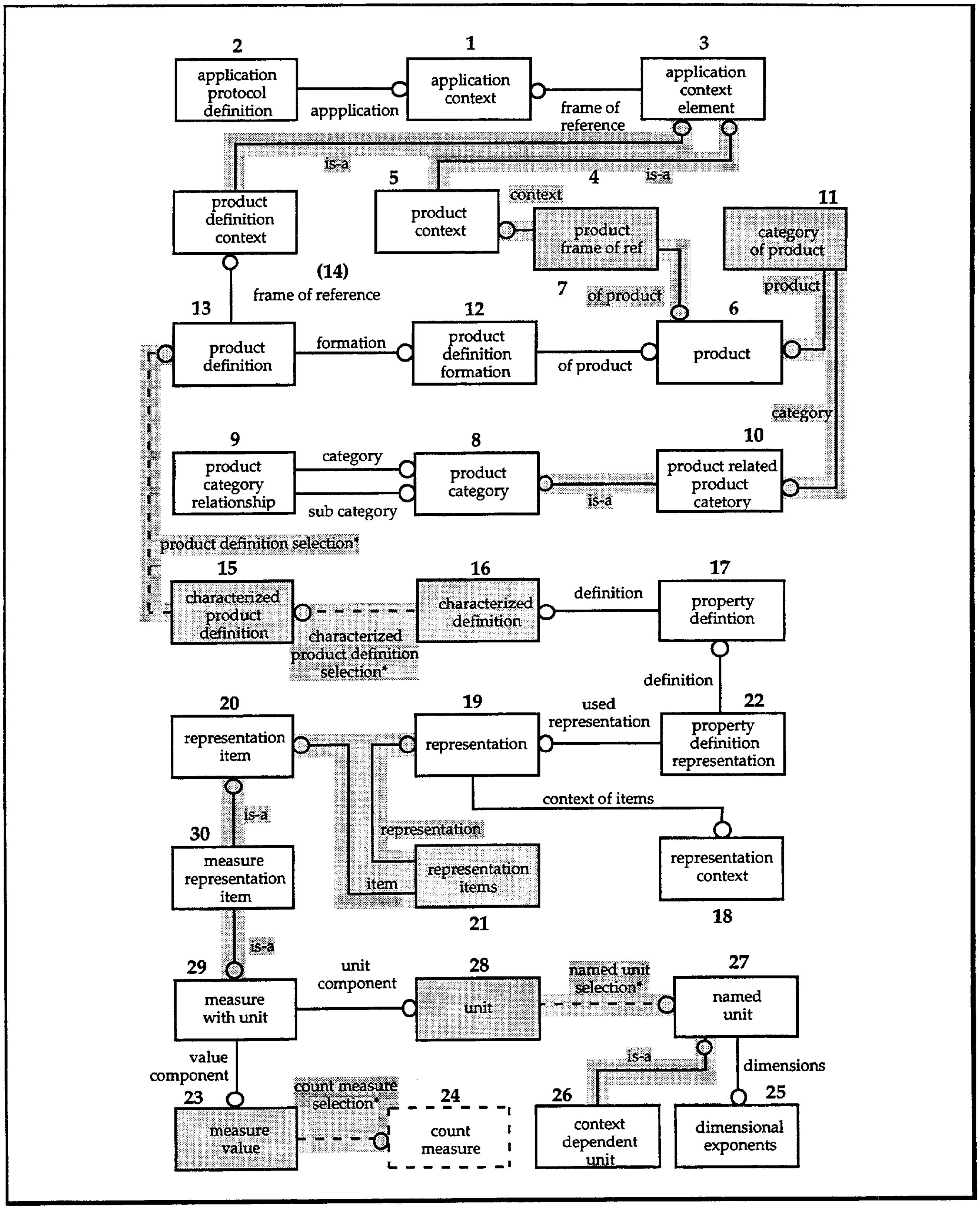


Figure 28. Logical data structure for example AP 1 AIM.

Each logical element for example AP 1 is assumed to have an OID. An instance of a logical element (i.e., a row in a table) is uniquely identified by the value of its OID.[13]

Unique instance identifiers are constructed for the sample population of example AP 1. Two approaches have been used for illustration. The first uses the logical element number times 1000. This approach allows the same number to be used for the identification of an instance in both the example database and when using the clear text encoding standard of STEP (i.e., for a physical file exchange example). The second approach uses the convention 'project_identifier-AIM_element_number-instance_number'. Therefore, the first instance of the application context table for a project identified as CIC1 (i.e., a project entitled Computer Integrated Construction 1) has an OID of CIC1-1-1.

Other more rigorous approaches have been suggested including the use of ISO registered identifiers [15]. The purpose of this document is not to address a solution to this issue, but rather to employ a simple approach for the purpose of illustration. Whatever the solution, identifying instances unambiguously is essential if users are to maintain information about a product over its entire life cycle.

Application context

The application context information is stored using four tables. They are the application context, application protocol definition, product context, and product definition context.

1 Application context

appl_con_oid	application
1001 / CIC1-1-1	economic viability analysis

2 Application protocol definition

ap_def_oid	status	aim_schema_name	ap_year	appl_con_oid
2001 / CIC1-2-1	wd	ap1_aim_schema	1996	1001 / CIC1-1-1

3 Application context element

appl_con_elem _oid	name	frame_of_reference
3001 / CIC1-3-1	building industry	1001 / CIC1-1-1
3002 / CIC1-3-2	as-required	1001 / CIC1-1-1

[13] A more traditional approach to relational database implementations would use keys (i.e., a combination of one or more rows in a relational table that are governed by a uniqueness constraint). The use of OIDs is for illustration and consideration as an alternative to the use of keys. Unique attributes identified in STEP AIMs would contribute to the more traditional approach that uses keys. Different demands would be made upon data management functions in the two approaches.

4 Product context

prod_con _oid	name	discipline_type	appl_con_elem_oid
4001 / CIC1-4-1	building industry	economics	3001 / CIC1-3-1

5 Product definition context

prod_def_con _oid	name	life cycle stage	appl_con_elem_oid
5001 / CIC1-5-1	as-required	specify economic design requirements	3002 / CIC1-3-2

When retrieving information, two queries (illustrated using SQL), one for product context and one for product definition context, identify information that may be referenced by the semantic elements of the product definition schema. Each includes an application context that is referenced by an AP definition for AP 1.

Product context:

```
SELECT DISTINCTROW Product_context.Product_context, Product_context.Discipline_type,
Applicationn_context_element.Name, Application_context.Application
FROM (Application_context INNER JOIN Application_context_element ON
Application_context.Application_context = Application_context_element.Applicationn_context)
INNER JOIN Product_context ON Application_context_element.Application_context_element =
Product_context.Application_context_element
WHERE ((Product_context.Discipline_type="economics") AND
(Application_context_element.Name="building industry") AND
(Application_context.Application="economic viability analysis"));
```

product_context.oid	CIC1-4-1
product_context.discipline_type	economics
application_context_element.name	building industry
application_context.application	economic viability analysis

Product definition context:

```
SELECT DISTINCTROW Product_definition_context.Product_definition_context,
Product_definition_context.Life_cycle_stage, Application_context_element.Name,
Application_context.Application
FROM (Application_context INNER JOIN Application_context_element ON
Application_context.Application_context = Application_context_element.Application_context)
INNER JOIN Product_definition_context ON
Application_context_element.Application_context_element =
Product_def_context.Application_context_element
WHERE ((Product_definition_context.Life_cycle_stage="specify economic design requirements") AND
(Application_context_element.Name="as-required") AND
(Application_context.Application="economic viability analysis"));
```

product_definition_context.oid	CIC1-5-1
product_definition_context.life_cycle_stage	specify economic design requirements
application_context_element.name	as-required
application_context.application	economic viability analysis

The reference path constraints serve two purposes with respect to the example database population. First, the constraints specify the values that are acceptable in storing the information when using the application context appropriate to a particular AP conformance class. Second, the reference path constraints provide criteria for querries that retrieve information that is both meaningful and useful in the application context. Reference path constraints are used to specify how the data structure is to be used both to store and retrieve relevant information.

Product definition

The product definition information is stored using seven tables. They are product, product context relationship, prodcut category, product related product category, category product relationship, product definition formation, and product definition.

6 Product

prod _oid	name	id	description
6001 / CIC1-6-1	NIST Blue Auditorium	Room 1 Building 2002	an example product

7 Product frame of reference

prod_frame_of_ref_oid	prod_con_oid	prod_oid
7001 / CIC1-7-1	CIC1-4-1	6001 / CIC1-6-1

8 Product category

prod_ cat _oid	name	description
8001 / CIC1-8-1	performance hall	a performance space

10 Product related product category

prod_rel_prod_cat _oid	prod_cat_oid
10001 / CIC1-10-1	8001 / CIC1-8-1

11 Category products

cat _prods_oid	product_oid	prod_rel_prod_cat_oid
11001 / CIC1-11-1	6001 / CIC1-6-1	10001 / CIC1-10-1

12 Product definition formation

prod _def_form_oid	id	description	product_oid
12001 / CIC1-12-1	v 1.00.00	initial version	6001 / CIC1-6-1

13 Product definition

prod _def _oid	id	description	prod_def_form_oid	prod_def_con_oid
13001 / CIC1-13-1	economic requirement definition 1	by BFRL	12001 / CIC1-12-1	5001 / CIC1-5-1

A query on product definition identifies data as being a BFRL economic requirement definition for the initial version (v1.00.00) of the NIST Blue Auditorium (an example product). This product definition was created during the specify economic design requirments stage of the products life cycle as part of an economic viability analysis for as-required performance halls.

product_definition.oid	CIC1-13-1
product_definition.name	economic requirement definition 1
product_definition.description	by BFRL
product_definition_formation.id	v1.00.00
product_definition_formation.description	initial version
product.name	NIST Blue Auditorium
product. description	an example product
product_definition_context.life_cycle_stage	specify economic design requirements
application_context_element.name	as-required
application_context. application	economic viability analysis

Product property definition

The product property definition information is stored using three tables. They are characterized product definition, characterized definition, and property definition.

15 Characterized product definition

char_prod_def_ oid	selection
15001 / CIC1-15-1	13001 / CIC1-13-1

16 Characterized definition

char_def _oid	selection
16001 / CIC1-16-1	15001 / CIC1-15-1

17 Property definition

prop _def _oid	name	description	char_def _oid
17001 / CIC1-17-1	audience capactiy	viability criterion	13001 / CIC1-13-1

A query on property definition provides all the information available from the previous query on the product definition and identifies a property named audience capacity which is described as a viability criterion that characterizes economic requirement definition 1.

property_definition.oid	CIC1-17-1
property_definition.name	audience capacity
property_definition.description	viability criterion
product_definition.id	economic requirement definition 1
product_definition.description	by BFRL
product_definition_formation.id	v1.00.00
product_definition_formation.description	initial version
product.name	NIST Blue Auditorium
product. description	an example product
product_definition_context.life_cycle_stage	specify economic design requirements
application_context_element.name	as-required
application_context. application	economic viability analysis

Representation

The representation information is stored using four tables. They are representation context, representation item, representation, and representation items.

18 Representation context

rep _con _oid	id	type
18001 / CIC1-18-1	audience capacity context 1	valued

19 Representation

rep _oid	name	rep_con_oid
19001 / CIC1-19-1	minimum seated and standing audience capacity	18001 / CIC1-18-1

20 Representation item

rep _item _oid	name
20001 / CIC1-20-1	seating audience capacity
20002 / CIC1-20-2	standing audience capacity

21 Representation items

rep_items _oid	rep_oid	rep_item_oid
21001 / CIC1-21-1	19001 / CIC1-19-1	20001 / CIC1-20-1
21002 / CIC1-21-2	19001 / CIC1-19-1	20002 / CIC1-20-2

A query on representation provides two tuples, each for a minimum seated and standing audience capacity representation in audience capacity context 1 of type valued. The first tuple contains a representation item named seated audience capacity and the other named standing audience capacity.

representation.oid	CIC1-19-1
representation.name	minimum seated and standing audience capacity
representation_context.id	audience capacity context 1
representation_context.type	valued
representation_item.name	seated audience capacity

representation.oid	CIC1-19-1
representation.name	minimum seated and standing audience capacity
representation_context.id	audience capacity context 1
representation_context.type	valued
representation_item.name	standing audience capacity

Product property representation

The product property representation information is stored using one table.

22 Property definition representation:

prop_def_rep _oid	prop_def_oid	rep_oid
22001 / CIC1-22-1	17001 / CIC1-17-1	19001 / CIC1-19-1

This table establishes seated/standing audience capacity as a representation of audience capacity.

property_definition_representation.oid	CIC1-22-1
representation.name	minimum seated and standing audience capacity
representation_context.id	audience capacity context 1
representation_context.type	valued
property_definition.name	audience capacity
property_definition.description	viability criterion
product_definition.id	economic requirement definition 1
product_definition.description	by BFRL
product_definition_formation.id	v1.00.00
product_definition_formation.description	initial version
product.name	NIST Blue Auditorium
product. description	an example product
product_definition_context.life_cycle_stage	specify economic design requirements
application_context_element.name	as-required
application_context. application	economic viability analysis

A query results in two tuples that repeat this information with the one specifying a representation item seating audience capacity and the other standing audience capacity.

Measure

The measure information is stored using four tables. They are measure value, dimensional exponents, named / context dependent unit, and measure with unit.

23 Count measure

count_meas _oid	number
23001 / CIC1-23-1	3000
23002 / CIC1-23-2	200

24 Measure value

meas_val _oid	selection
24001 / CIC1-24-1	23001 / CIC1-23-1
24002 / CIC1-24-2	23001 / CIC1-23-1

25 Dimensional exponents[14]

dim_exp _oid	l	m	t	ec	tt	as	li
25001 / CIC1-25-1	0	0	0	0	0	1	0

26 Named unit

named _unit_oid	dim_exp_oid
26001 / CIC1-26-1	26001 / CIC1-25-1

27 Context dependent unit

con_dep _unit_oid	name	named_unit_oid
27001 / CIC1-27-1	persons	26001 / CIC1-26-1

28 Unit

unit_oid	selection
28001 / CIC1-28-1	26001 / CIC1-26-1

29 Measure with unit

meas_with _unit_oid	meas_val_oid	unit_oid
29001 / CIC1-29-1	24001 / CIC1-24-1	28001 / CIC1-28-1
29002 / CIC1-29-2	24002 / CIC1-24-2	28001 / CIC1-28-1

---

[14] The abbreviations for dimensional exponents in this table include l (length), m (mass), t (time), ec (electric current), tt (thermal temperature), as (amount of substance), and li (luminous intensity).

A query provides two tuples, one identifying a measure with unit of 3000 persons and the other 200 persons.

measure_with_unit.oid	CIC1-29-1
count_measure.number	3000
context_dependent_unit.name	persons

measure_with_unit.oid	CIC1-29-2
count_measure.number	200
context_dependent_unit.name	persons

Each of these tuples would include the dimensional exponents 0, 0, 0, 0, 0, 1, 0 that indicates that amount of substance is used for dimensional checks with respect to the these values and units (i.e., 200 persons has a dimensionality of amount of substance).

Measure qualification

The qualification information is stored using one table. It is the measure representation item.

30 Measure representation item:

meas_rep _item_oid	meas_with_unit_oid	rep_item_oid
30001 / CIC1-30-1	29001 / CIC1-29-1	19001 / CIC1-19-1
30002 / CIC1-30-2	29002 / CIC1-29-2	19002 / CIC1-19-2

The measure representation item table connects all of the information available in the previous reports.

property_definition.oid	CIC1-17-1
property_definition.name	audience capacity
property_definition.description	viability criterion
product_definition.id	economic requirement definition 1
product_definition.description	by BFRL
product_definition_formation.id	v1.00.00
product_definition_formation.description	initial version
product.name	NIST Blue Auditorium
product. description	an example product
product_definition_context.life_cycle_stage	specify economic design requirements
application_context_element.name	as-required
application_context. application	economic viability analysis

property_definition_representation.oid	CIC1-22-1
property_definition.oid	CIC1-17-1
representation.oid	CIC1-19-1

The property definition representation identifies the representation minimum seated and standing audience capacity (CIC1-19-1) as being a representation of the audience capacity property definition.

representation.oid	CIC1-19-1
representation.name	minimum seated and standing audience capacity
representation_context.id	audience capacity context 1
representation_context.type	valued
representation_item.name	seating audience capacity
count_measure.number	3000
context_dependent_unit.name	persons

representation.oid	CIC1-19-1
representation.name	minimum seated and standing audience capacity
representation_context.id	audience capacity context 1
representation_context.type	valued
representation_item.name	standing audience capacity
count_measure.number	200
context_dependent_unit.name	persons

A property definition with name audience capacity and description viability criterion characterizes a product definition with an id of economic requirement definition 1 and description of by BFRL. This product definition is an element of a product definition formation with an id of v1.00.00 and description of initial version of a product with the name NIST Blue Auditorium with a description of an example product. This product definition has a frame of reference of the specify economic design requirements life cycle stage. This stage is of an application context element with the name as-required that in turn has a frame of reference of an application context where the application is economic viability analysis.

The audience capacity is represented as a minimum seated and standing audience capacity that has seating audience capacity and standing audience capacity items within the context of items with an id of audience capacity context 1 of type valued. The seating audience capacity representation item is a measure representation item with a value of 3000 and a unit of persons. The standing audience capacity representation item is a measure representation item with a value of 200 and a unit of persons.

Providing for multiple representations of property definitions is a principal feature of the STEP product data architecture. If in the future, this AP or another AP has a requirement for a different representation of the same audience capacity, the two or more representations may exist compatibly. This is similar to the other principal feature of STEP that an indefinite number of compatible product definitions may be developed for the same product. These two features are essential elements that provide the basis for the development and use of interrelated APs.

4. Interrelated domain ontologies in an integrated application context environment

An AP is one perspective used to describe and represent one or more products. Interrelated APs (IAPs) are different perspectives on one or more products that together provides a more comprehensive perspective. A collection of IAPs (IAPC) has a scope that combines that of constituent IAPs (Fig. 31). The collective scope may contain different granularities of abstraction for different purposes. As STEP evolves, more inclusive IAPCs are possible and thereby the comprehensiveness of the picture that is achievable increases.

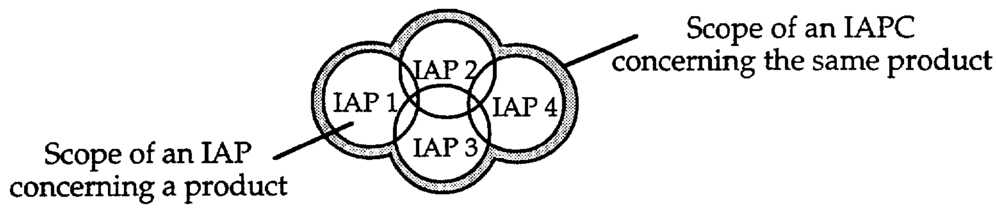


Figure 31. The scope of IAPs and of an IAPC concerning the same product.

The use of conformance classes from a single AP and of conformance classes of an IAPC containing multiple IAPs address different industry requirements. An AP conformance class is used to satisfy the requirement for unambiguous communication for specific purposes within a particular application context and scope. Users of information can communicate unambiguously to the degree that the conformance class of the AP coincides with the application contexts and scopes of the users (Fig. 32).

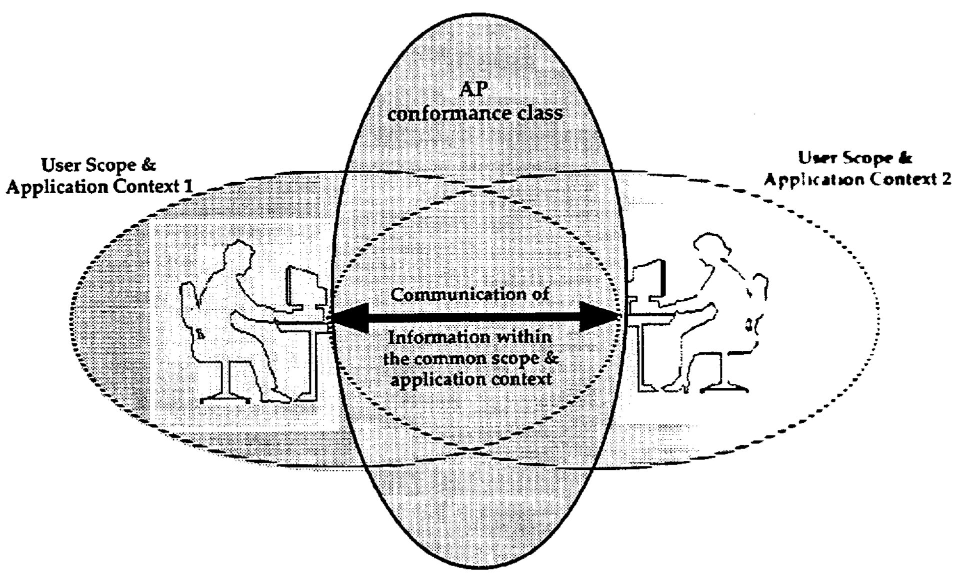


Figure 32. Communication using a single AP conformance class.

The use of a conformance class from a single AP for communication ensures that the information conveyed is that which was intended. It also ensures that the information is adequate for the intended purposes described in usage scenarios.

The use of an IAPC has a different goal. It is to facilitate the development of an environment in which information is used cooperatively among multiple application contexts (i.e., an integrated application context environment). An integrated application context environment not only provides the ability to create and use information about a product when it is needed in a form appropriate to the task, it also provides coordinated management of shared information among its application contexts. IAPCs are useful to industry in its effort to increase accuracy and efficiency in its use of information by applications.

The cooperative use of information by applications within and among divisions, departments, disciplines, and enterprises has met with only limited success in the absence of standardization. An international standard that enables the cooperative use of information about products is urgently needed. STEP is expected to fulfill this requirement. A principal question for STEP is whether the product data architecture and methods that guide the development and use of APs for communication in a specific application context are suitable for the development and use of IAPCs for the cooperative use of information in an integrated application context environment.

This section describes a general strategy for the development and use of IAPCs supported by the current architecture and methods of STEP. It involves the development of IAPs using the IRs in such a way that they standardize different but compatible perspectives. These compatible perspectives can be used both for communication involving more than one IAP and for sharing information among applications in an integrated application context environment (Fig. 33). The scope of such an environment is the collective scope of the IAPC.

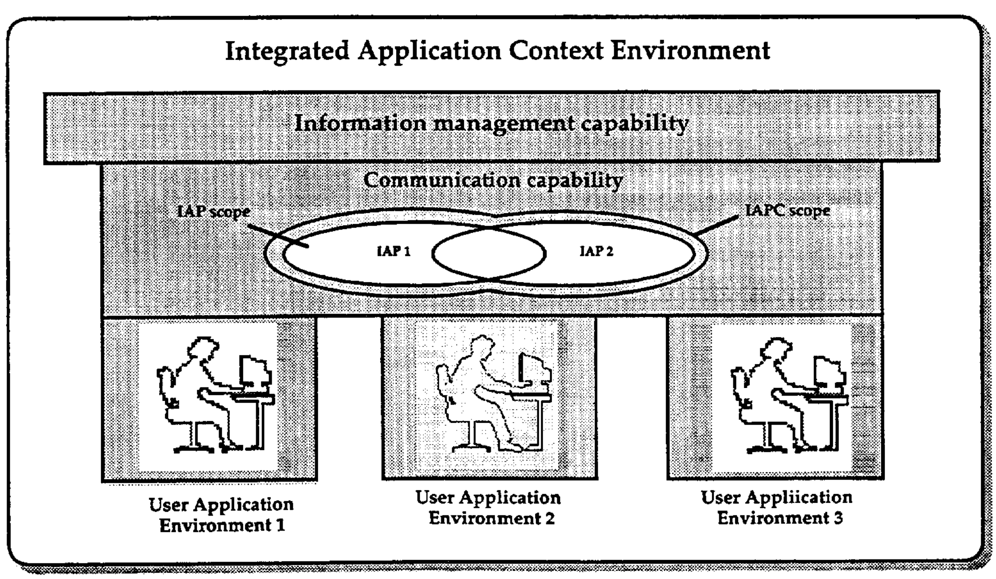


Figure 33. An integrated application context environment using IAPs.

Conceptually, an integrated application context environment has three principal elements. They are user application environments, a communication capability, and an information management capability. User application environments are characterized by their scope, application context, and granularity of abstraction. The communication capability supports all of the IAPs within the collective scope of the IAPC (e.g., IAP 1 and 2 in Fig. 31). User application environments need have access only to those IAPs relevant for sending and receiving information to support their application activities. The information management capability provides unambiguous identification of information across all applications and maintains the integrity of shared information within the collective scope of the IAPC.

The integrated application context environment can be used for communication and information sharing in a number of different ways depending upon the needs of the user application environments. In one scenario, user application environments 1 and 2 (in Fig. 33) could be two acoustical consulting firms. They might be working on the same task, that of providing a recommendation for the target volume of the audience space in a new concert hall. In this scenario, environments 1 and 2 have a common activity (Activity A in Fig. 34) with common information. The IAP 1 conformance class supports a usage scenario event where the output of activity A at one firm is an input to the same activity at another firm. The identity and integrity of the information is maintained by the information management capability of the integrated application context environment. IAP 1 can be used for communication between the firms which share both the activity and the information with both user environments.

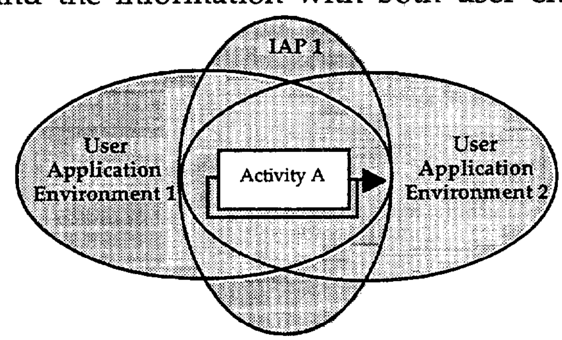


Figure 34. The same role for information in the scope of a single IAP.

Another scenario involves two activities, B and C (Fig. 35). Activity B is within the scope of environment 1 and activity C is within the scope of environment 2. When the user application environments support different activities within the scope of a single IAP communication is still possible. The information will play the role of an output of an activity at one firm and an input (or control) to an activity at the other.

One consulting firm might be responsible making preliminary recommendations about the materials used on surfaces within the hall based on calculations to achieve desired reverberation times. The information on the materials may then be communicated to the second firm that conducts computer simulations to give detailed reverberation times in specific audience areas. In this scenario, the information about the materials plays the role of an output of the activity performed by the first firm, activity B in user environment 1, and an input to the activity performed by the second firm, activity C of environment 1, and an input to the activity performed by the second firm, activity C of user environment 2. Each activity is within the scope and application context of only one environment. Only the information that flows between these activities is common to both environments as well as the conformance class of IAP 1.

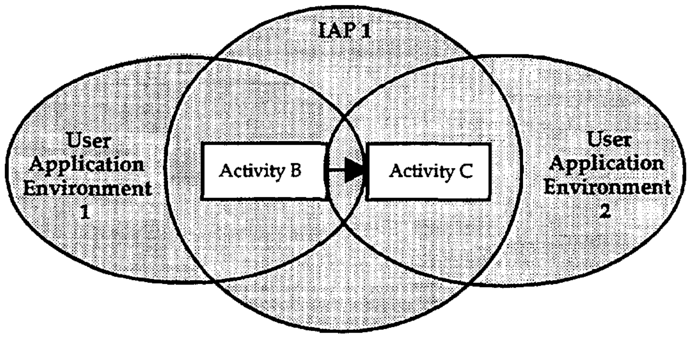


Figure 35. Different roles for information in the scope of a single IAP.

IAP 1 contains within the scope of its conformance class both activities B and C as well as the information flow between them. Both application environments can use IAP 1 to communicate the information about the materials used in the concert hall. The information will be an output from the perspective of user environment 1 and an input from the perspective of user environment 2. The information management capability of the integrated application context environment must manage the identity and integrity of the information.

Information to be communicated may also be within the scope of more than one IAP. The user application environments may support a single application activity that is common to two IAPs or each application environment may support a different application activity one of which is within the scope of one IAP and the other within the scope of another IAP. The role of the information is the same if the information is for a common application activity within the scopes of the two IAPs (Fig. 36). In this example, user application environment 2 uses IAP 1 while user application environment 3 uses IAP 2. Activity A is within the scope of both user application environments and both IAPs. This may occur when two coordinated applications have different IAPs available for communication each of which is generally a better match for the information with which it deals.

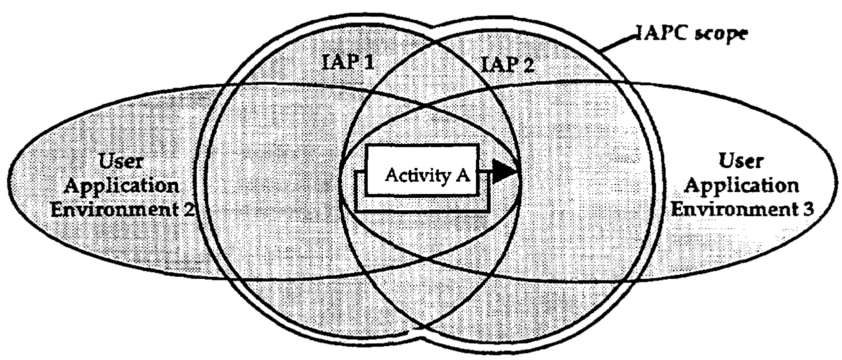


Figure 36. The same roles for information in the scope of different IAPs.

When user application environments support different activities each of which is within the scope of a different IAP, the information plays different roles from the perspectives of the users (Fig. 37). In this example, the information plays the role of an output of activity B from the perspective of both user environment 2 and IAP 1. The information is an input to activity C from the perspective of both user environment 3 and IAP 2. The IAP scopes in this case have only the flow of information between these activities in common.

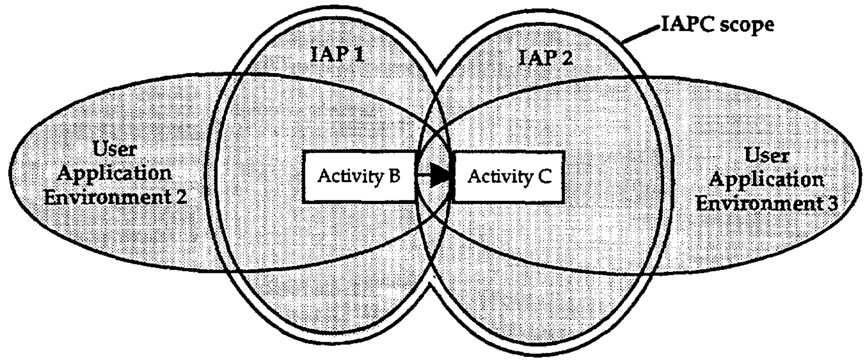


Figure 37. Different roles for information in the scope of different IAPs.

Whether one or more IAPs are used for communication, and whether that communication is for the purpose of information exchange or information sharing, the essential element with respect to the use of STEP is that the information is within the scope and application context of the user application environments and the scope of the conformance classes of the applicable IAPs. Communication is possible using a single AP and using multiple IAPs. This communication can be for the purposes of exchange or for information sharing. These are separate but related issues. The fundamental requirements regarding each of these choices is presented below.

Communication of information

Communication among user application environments **using one AP** is possible when:

- a coincidence of scope and application context exists between the user application environments and one or more conformance classes of the AP.

Communication among user application environments **using IAPs** is possible with the IAPC approach when:

- a coincidence of scope and application context exists between the user application environments and one or more conformance classes the IAPs, and
- the IAPs contain the same or related application activities for which there is one or more information flows that contain common information (both in terms of its semantics and context).

Exchange of information using an IAPC

Communication involving information exchange among user application environments using IAPs is possible with the IAPC approach when:

- a coincidence of scope and application context exists between the user application environments and one or more conformance classes of the IAPs,
- the IAPs contain the same or related application activities for which there is one or more information flows that contain common information, and
- a communication capability is established that uses knowledge of coincident scopes, application context representations, and common information requirement representations among the user application environments and the conformance classes of the IAPs available to those environments.

Sharing of information using an IAPC

Communication involving information sharing among user application environments using one or more IAPs is possible with the IAPC approach when:

- a coincidence of scope and application context exists between the user application environments and one or more conformance classes the IAPs, and
- the IAPs contain the same or related application activities for which there is one or more information flows that contain common information, and
- a communication capability is established that uses knowledge of coincident scopes, application context representations, and common information requirement representations among the user application environments and the conformance classes of the IAPs available to those environments, and
- an information management capability is established that maintains the identity and semantic integrity of the information within the collective scope and application context of the IAPC.

IAPCs can be developed and used to enable the cooperative use of information by applications. Such use of information can involve one or more products and cover as much of each product's life cycle as is needed. For STEP to contribute to the development of integrated application context environments within industry, the architecture and methods of STEP must be employed rigorously. Consensus domain ontology descriptions and representations, using the common constructs of the IRs, must be developed with an explicit mapping between them. The domain ontology representations must be compatible through concise and consistent interpretation that captures the diversity of perspectives about products such that the information can be used cooperatively in an integrated application context environment.

References

[1] Yourcenar, M.; Reflections on the composition of Memoirs of Hadrian, in Memoirs of Hadrian, The Noonday Press, New York, 1963.

[2] Casti, J.L.; Complexification: Explaining a paradoxical world through the science of surprise, Harper Collins Publishers, New York, 1994.

[3] Mayer, R.J., Menzel, C.P., Painter, M., and Benjamin, P.C.; The role of ontology in enterprise integration, in IDEF-Users Group Conference Proceedings, University of Maryland, College Park, MD, May 10-13, 1993.

[4] ISO 10303-41:1994(E) Industrial automation systems and integration - Product data representation and exchange - Part 41: Integrated generic resources: Fundamentals of product description and support.

[5] Feeney, A.B., Gilbert, M.; Guidelines for AIM development, ISO TC184/SC4 N435, April 1996.

[6] Feeney, A.B., Craig, D.; Guidelines for the development of mapping tables, ISO TC184/SC4 N436, April 1996.

[7] Craig, D., Feeney, A.B.; Guidelines for AIC development, ISO TC184/SC4 N437, April 1996.

[8] Integration definition for function modeling (IDEF0) , FIPS PUB 183, Federal Information Processing Standards Publication, U.S. Department of Commerce, 1993.

[9] Nijssen, G.M. and Halpin, T.A.; Conceptual schema and relational database design: A fact oriented approach, Prentice Hall, New York, 1989.

[10] Integration definition for information modeling (IDEF1X) , FIPS PUB 184, Federal Information Processing Standards Publication, U.S. Department of Commerce, 1993.

[11] ISO 10303-11:1994(E) Industrial automation systems and integration - Product data representation and exchange - Part 11: Descriptive methods: The EXPRESS language reference manual.

[12] ISO 10303-43:1994(E) Industrial automation systems and integration - Product data representation and exchange - Part 43: Integrated generic resources: Representation structure.

[13] ISO/DIS 10303-45:1995(E) Industrial automation systems and integration - Product data representation and exchange - Part 45: Materials.

[14] ISO 10303-1:1994(E) Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.

[15] ISO 10303-21:1994(E) Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure.

[16] Christensen, N.C.; Globally Unambiguous Identifier, electronic mail, 1995.

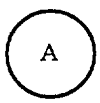
[17] Hawryszkiewycz, I.T.; Database analysis and design, Science Research Associates, Chicago, 1984.

[18] Date, C.J.; An introduction to database systems, Addison-Wesley Publishing Company, Reading, 1986.

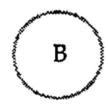
Annex A: Elements of NIAM

The graphical symbols of NIAM used in this document (after [9]):

Non-lexical object — an entity

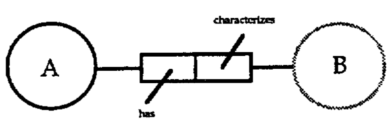


Lexical object — a valued element

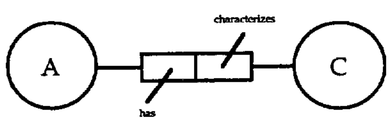


Role — the relation between one object and another (expressed as a verb or verb phrase)

Fact — an assertion (i..e., a well formed sentence)

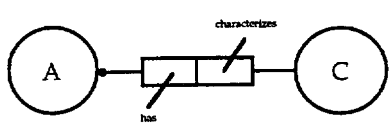


A has B / B characterizes A



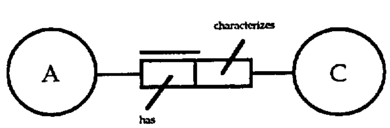
A has C / C characterizes A

Mandatory constraint — identifies a fact that must be true for every instance of an entity



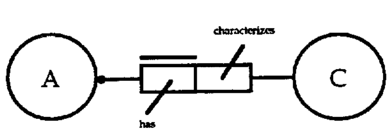
A has at least one B

Uniqueness constraint — identifies a fact that may be true at most once for any given instance of an entity



A has at most one C

Mandatory and uniqueness constraint — identifies a fact that is true exactly once for any given instance of an entity

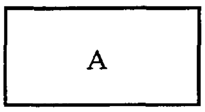


A has exactly one C

Annex B: Elements of EXPRESS-G

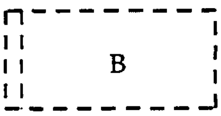
The graphical symbols of EXPRESS-G used in this document[15] (after [11]):

An entity data type



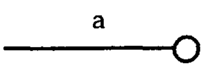
A representation of an entity.

A select data type



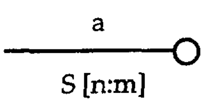
A generalization of the data types (entity or select) in a select list.

An attribute



A relationship between an entity and a referenced data type.

An attribute set

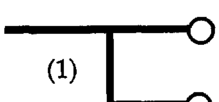


A relationship between an entity and an aggregation (set) of from n to m instances of a referenced data type (entity).

A supertype/subtype relationship

An inheritance relationship between a supertype and a subtype.

A ONEOF constraint



A supertype instance may be an instance of at most one subtype.

[15] The EXPRESS-G used in this document is for the presentation of resource assertions only. The descriptions given here, though adequate for the intended purpose, are not nearly as precise as those contained within the EXPRESS Language Reference Manual (ISO 10303-11).

Annex C: Elements of STEP reference path syntax

The graphical symbols of reference paths used in this document (after [6]):

[] multiple AIM elements or sections of the reference path are required to satisfy an information requirement

( ) multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement

{ } enclosed section constrains the reference path to satisfy an information requirement

-> attribute references the entity or select type given in the following row

<- entity or select type is referenced by the attribute in the following row

[i] attribute is an aggregation of which a single member is given in the following row

[n] attribute is an aggregation of which member n is given in the following row

=> entity is a supertype of the entity given in the following row

<= entity is a subtype of the entity given in the following row

= the string, select type, or enumeration type is constrained to a choice or value