

The STEP Integration Information Architecture

**William C. Burkett
Yuhwei Yang
Product Data Integration Technologies, Inc.
3780 Kilroy Airport Way, Suite 430
Long Beach CA 90806**

**310-427-3711, 310-427-4840 FAX
burkett@usc.edu
yyang@mcimail.com**

A B S T R A C T

Traditionally, when the term "integration" is used to refer to the interoperability of disparate, heterogeneous computer system, it means the ability to exchange digital data between the systems. For the more sophisticated systems designers, "integration" may mean shared, distributed databases or a federated database system. Within the development of the Standard for the Exchange of Product model data (STEP - ISO 10303), "integration" refers to an information architecture composed of conceptual constructs that is independent of implementation considerations.

The Integration Information Architecture of STEP is presented and explained. Instead of a flat representation of abstract (i.e., conceptual) data structures, integration within STEP takes place at four different levels:

- 1) Intra-Resource integration;
- 2) Structural integration of Application Protocols through Integrated Resources;
- 3) Semantic integration of Application Protocols through Application Interpreted Constructs (AICs);
- 4) Operational integration through Application Protocols.

Each level of integration and the relationships between the levels is explained.

1 Introduction

Over the past 40 years, industrial automation has seen dramatic advances in terms of the capabilities and precision of the available technology. From numerical control (NC) in the fifties, through the first design graphics applications and computer controlled production operations in the sixties, Computer Numerical Control (CNC) and Distributed Numerical Control (DNC) in the seventies, and Flexible Manufacturing Systems (FMS) and solid model-based design workstations in the eighties, automation technology has continued to advance and become more sophisticated in order to meet the individual needs of industry. However, as industry moves into the nineties, a new industrial need is becoming *the* critical problem to solve: the integration of these diverse automation systems (e.g., CAD, CAM, CIM, CAE, etc).

STEP addresses this need. The **Standard for the Exchange of Product model data** (STEP) is an evolving international standard for the representation and exchange of product data. The technical objective of STEP is to enable communication of product data (e.g., a product design) between heterogeneous CAD/CAM systems. While integration as a whole is a much larger problem, PDES/STEP provide an initial piece to the integration puzzle by standardizing a data exchange format that is developed and approved through international consensus.

In 1984, the International Organization for Standardization (ISO) Technical Committee 184 (Industrial Automation), Subcommittee 4 (TC184/SC4) embarked on the development of a new standard that was to be known (unofficially) as STEP. ISO 10303 is the official ISO designation for the standard commonly referred to as STEP. The full title of the standard is: ISO 10303 - Industrial Automation Systems - Product Data Representation and Exchange. It is under development by Subcommittee 4 (SC4) of TC184 and is defined as:

"... a neutral mechanism capable of completely representing product data throughout the life cycle of a product...The completeness of this representation makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing databases and archiving." [1]

ISO 10303 is a series of individual Parts; these Parts can be grouped into classes according to the role that the Part plays within ISO 10303. Some Parts are more intimately related than others, but overall the Parts of ISO 10303 are an integrated unit that specifies the requirements for exchanging product data.

Within SC4, the technical organization and structure of the standard was designed jointly by Working Group 4 - Qualification and Integration and Working Group 5 - STEP Development Methods and evolved through an iterative consensus process within the SC4. The Editing Committee and the Integration Project (WG4/P2) are jointly responsible for consistency of products of SC4 with respect to this technical architecture. The Editing Committee has focused on quality and consistency of the documentation across all Parts, while Integration has focused almost exclusively on the quality and consistency of the "conceptual schema" of STEP (i.e., the Integrated Resources and the Application Interpreted Models of Application Protocols).

The purpose of this paper is to introduce and explain the kinds of integration within STEP and SC4. Unlike traditional database integration efforts, STEP makes use of abstraction as a principal integration mechanism. This results in a hierarchical conceptual schema that permits data constructs to be

"re-used" in different contexts. The architecture of the standard will first be presented; this will be followed by an explanation of Integration in STEP.

2 Architecture of STEP

The objective of this section is to present the architecture of STEP by:

- presenting the EXPRESS language and the role that it plays in STEP;
- specifying the roles of the Integrated Resource and Application Protocols;

The presentation of the architecture will focus on the relationship between the Application Protocols, Integrated Resources, and Implementation Forms.

1 presents the document architecture for STEP. It shows the components of the standard and the general relationships between them.

The STEP architecture is a unique solution to a very complex problem: the meaningful communication of product information between unspecified industrial automation systems. It is also an innovative solution with respect to the design and use of conceptual models.

2.1 EXPRESS

The EXPRESS language [2] is a machine processable language used to specify the information content of STEP through the use of schemas. One or more schemas are combined into models which are documented in the Integrated Resource and Application Protocol Parts of STEP. The roles of the model/schemas in each of these classes of Parts is different, however.

EXPRESS is a PASCAL-like, declarative, information modelling language. It is composed of constructs such as entities, types, rules and functions; these are used to specify the information content of some application domain (i.e., domain semantics). An example of an EXPRESS entity declaration is:

```
ENTITY cartesian_point
  SUBTYPE OF (point);
  local_coordinate_system : OPTIONAL coordinate_system;
  x_coordinate             : length_measure;
  y_coordinate             : length_measure;
  z_coordinate             : OPTIONAL length_measure;
  DERIVE
    dim : INTEGER := count_dimensions([x_coordinate, y_coordinate,
                                      z_coordinate]);
END_ENTITY;
```

EXPRESS is a very flexible information modelling language that is capable of representing many types of data found in information/automated systems.

2.2 Integrated Resources

Integrated Resources is the class of STEP Parts which specify the EXPRESS schemas that are independent of (specific) implementation and usage context, i.e., they may be used in a wide variety of implementations and applications. Each Part within the Integrated Resource class contains one or more schemas; the schemas are very small and narrowly defined and the resulting modularity promotes design extensibility. The schemas specify conceptually generic constructs that can be "reused" within different application contexts. The entire collection of Integrated Resource schemas is an interconnected, non-redundant network of conceptual constructs; it is "flat" collection of constructs. The entities (and the rules, types, and functions which make up a schema) are the data structures that constitute the "vocabulary" of the standard.

The application usage context of the Integrated Resources is specified within Application Protocols and may vary by product type, product life cycle phases, or any number of other application contexts. IR's are generic and "context-free" with respect to APs. The overall, umbrella context for the IRs (since nothing is really "context-free") is roughly described by the intersection of terms like "product data", "industrial automation systems", and "data structures".

The schemas are derived from many different application domains (as specified in APs) and are segregated into Parts based on similarity or applicability of the schemas. The schemas within the Integrated Resources are strongly dependent on one another and are tightly woven into a single consistent and uniform model. A single concept is represented only once within the Integrated Resources.

2.3 Application Protocols and AICs

Application Protocols are the class of STEP Parts which define the data exchange requirements for a relatively narrowly scoped application domain. It defines the context, scope and information requirements for the designated application and specifies elements of the Integrated Resources that are used to meet these requirements. In addition, Application Protocols include conformance requirements for assessing whether an implementation conforms to the standard (i.e., the AP).

There are three primary components to an Application Protocol:

- Information Requirements;
- Application Interpreted Model; and
- Conformance Requirements and Test Purposes.

The Information Requirements specify the context-specific information that is to be communicated via the Application Protocol. These requirements are derived from an Application Reference Model (ARM) - an information model that was developed based an Application Activity Model (AAM) of the application domain. These models are developed without considering the contents of the Integrated Resources. They represent the requirements of a specific user for exchanging product data.

The Application Interpreted Model (AIM) consists of the mapping between the ARM and the subset of Integrated Resources needed by the AP. The generic resource constructs are "interpreted" within the context defined by the AP; the mapping is called the "interpretation" of the generic construct. [3] In conjunction with the physical file structure, the AIM represents the "implementable" component of STEP,

i.e., the AIM specifies the contents of the physical file. While knowledge of the Integrated Resources is necessary, the AP is intended to be a standalone document. It is this approach to data exchange which is fundamentally different from IGES and other data exchange standards that specify a single, "flat" set of data structures.

The AIM is a single EXPRESS schema. It is "built" by importing entity declarations from the Integrated Resource schemas. Unlike the Integrated Resource schemas, an AIM is very large and multiple AIMS will overlap with respect to resource constructs used (hence the need for AP integration and AICs).

While the Integrated Resources provide generic constructs that can be used by many applications, the scope and interpretation of the constructs as specified in the AP provide the basis for conformance testing that will significantly increase the chances for a successful communication event (i.e., exchange of data/information). The Conformance Requirements and Test Purposes document these criteria.

Application Interpreted Constructs (AICs) were introduced to formalize interpretations of the integrated resources that are shared by multiple APs. An AIC is like a mini-AP: it specifies the structures and interpretation for a narrow and specific context. AICs provide semantic integration between APs.

The relationship between Integrated Resource constructs and Application Protocol requirements is illustrated in 2.

2.4 Implementation Forms

The class of STEP Parts called Implementation Forms contains the specifications of the physical format and grammar of the data structures contained in or used by implementations of the standard. The derivation of this specific format is based on EXPRESS schemas; constructs from the modelling language are syntactically mapped to the physical representation.

For the initial release of STEP, the only Implementation Form that is specified as part of the standard is the format for a discrete file exchange - Part 21. [4] Other implementation forms may be specified for databases, archives, or telecommunications. The relationship between the STEP conceptual schema and implementation forms is illustrated in 3.

Since the information content of STEP is to be independent of the syntax of the implementation, the implementation specification in Part 21 depends only on the structure and semantics of the EXPRESS language. The information content of the standard is specified by EXPRESS schemas in the Integrated Resource and Application Protocol Parts and *has no bearing whatsoever* on the resulting form of the physical file. The schemas are algorithmically mapped from the EXPRESS language to the physical data structure format.

3 The Role of Integration

Integration is generally perceived as an adjunct to the development of products that need to interface or relate to one another. It has been widely accepted that STEP Parts form a single standard; as such, integration is necessary among STEP Parts. Similar to the popular perception, the role of integration in STEP has been perceived as a "helper" that provides input to the development teams to make the Parts

"work together". However, this kind of surface integration would not have resulted in a consistent and cohesive standard. The Integration Project, therefore, undertook the task of enforcing a consistent and cohesive structure for the conceptual schema throughout the resource and AP Parts. This role of the Integration Project is much larger and more central than the perceived adjunct role.

4 Types of Integration

The "integration" of the kind that is taking place within SC4 is a revolutionary expansion of the concept of "integration". It is revolutionary because integration of this kind is not taking place (and isn't really recognized) in the field of computer science. It is far removed from hardware integration and really isn't even related to software integration, both of which tend to be the focus of computer science "integration" efforts. SC4 Integration, a necessary prerequisite to software/hardware integration for any true systems integration, focuses on the semantic integration of disparate information domains through a single conceptual schema. This conceptual schema is a much more complex structure than the term "conceptual schema" implies in the database field and the 3-schema architecture. "Conceptual Schema" in STEP refers to the collection of Integrated Resources and APs, at different levels of abstraction [5].

There are four distinct types of Integration that exist within STEP, as illustrated in 4.

4.1 Intra-resource Integration

The first type of Integration resembles the schema integration found in the database field and is indicated by (1) in 4. Individual STEP draft resource models are combined to form a single integrated model. This single integrated model is the STEP Integrated Resources, the standardized resource constructs for all Application Protocols to interpret. The STEP Integrated Resources are the most generic level of conceptual schema for the entire scope of SC4 standards.

4.1.1 Integration Strategy

Integration processing strategies can be divided into binary and n-ary strategies. [6] Binary strategies integrate any two modules/schemas at a time while n-ary strategies integrates more than two at a time. Binary strategies simplify the integration task at each phase of the integration but miss issues that only emerge upon consideration of the larger scope of an n-ary strategy. Binary processing is either balanced or ladder. Balanced processing places greater emphasis on the final product achieved through parallel integration. Parallel integration processing may speed the overall process, however, intermediate results are typically less useful and the integration of any the intermediate results may introduce significant changes to the previously integrated modules. Ladder processing attaches significance to the intermediate results at each phase of the integration. Each intermediate result is the current state of integration that serves as the basis for further integration.

STEP has a large and complex scope and STEP has a phased development environment that accommodates future needs as extensions of previous releases. STEP integrated resources are used as standard constructs for AP interpretation at any stage of the integration process. All intermediate results of the integration process are significant. An integration process that can satisfy STEP development objective and environment must (1) combine the desired characteristics of both the binary ladder and n-ary processing approaches; and (2) use an established explicit data structure as the integration framework.

The approach currently used by STEP Resource Integration is a n-ary ladder integration approach: one or more draft resource models is integrated into an intermediate resultant integrated resources (see 5). These intermediate resultant integrated resources are standardized for Application Protocol interpretation. This process was made possible by the establishment of a root structure for product data. This root structure is the baseline data architecture, i.e., the Integration Architecture of the SC4 technical architecture, used as the basis for all future resource extensions.

4.1.2 Form Integration and Structural Integration

Using this integration strategy, STEP integration differs from that in the database field in that it is more than just combining schemas and resolving conflicts. Schema integration for databases tends to be a "flat" integration; all schemas are integrated at "face value". The focus is on consistent terminology and proper interfaces, what we shall call "surface form integration"). In this type of integration there is little consideration of the design style of the schema. For example, the same concept may be represented differently in two different schemas; this issue must be addressed.

Rather than simply integrate the surface form, STEP Integration focuses on the underlying semantic structure of the draft resource models as they become a part of the integrated resources structure (what we shall call the "deep structure integration"). The goal is to integrate these individually developed draft resource models not only from a terminology/ semantic alignment standpoint (which could be accomplished with flat integration), but from an underlying semantic structural alignment standpoint as well. In order to form a truly integrated set of resource constructs, the conceptual organization (the design of the information representation) and the modelling style (the use of modeling language constructs), must be consistent across all the resource models.

Surface form integration is like sewing together different pieces of fabric which may not be of the same fiber or weave - the seam and fabric differences remain visible; Deep structure integration is like making sure that the different pieces of fabric are of the same fiber and weave and then interweaving them - there is no seam that remains; the result of the "integration" shows no boundary.

Thus, many of the changes to the resource schemas recommended by Integration not only have to do with resolving conflicts between the to-be-integrated schemas and the integration baseline (i.e., the intermediate integrated schemas), but with restructuring of the schemas to ensure a consistent overall conceptual structure and modelling style to fit with the Integration Architecture. Integration makes the "fiber and weave" of the schemas the same before putting them together. The structural integration of resource models does not mean that the semantics are unimportant, but rather that the semantics are conveyed by a consistent structure which may be different than that used in the pre-integration resource model.

The Integration Architecture [7] is a framework of context, definition, representation, association and presentation. The Integration Architecture is a "data model" which is used to specify and integrate the resource schemas. The Integration Architecture is a combination of EXPRESS and the rules for what constitutes a "good" (e.g., generic, consistent, structurally compatible) integrated resource schema.

4.1.3 Integration Rules

Many of the integration rules [8] have been established over the course of the integration practices in STEP. These rules are set against the Integration Architecture. These rules are classified to be semantic or syntactic. The semantic rules of integration include: modular organization of conceptual structures, modelling structures are generic and conceptual in nature, uniqueness of the concepts, functionally adequate structures, and proper placement in the Integrated Architecture. The syntactic rules enforces the modelling style and consistent usage of the EXPRESS language. Based on these integration rules, similar draft resource model structures are restructured using the same practices.

An example of the restructuring practices is to resolve definitional conflicts; different mandatory attributes exist for the same concept or identical attributes exist for different concepts. The EXPRESS subtype/supertype construct is used to specify a structure that the general concept is represented in a SUPERTYPE entity while the different additional attributes appear in the specialized concepts are represented in the constituent SUBTYPE entities. This structure represents the permissible combinations of attributes (that result from inheritance and inter-subtype constraints) results in the desired set of entities. This approach can result in an extremely "subtle" structure in stating the semantics from an application perspective and thus hide the semantics that the subject area experts wish to make visible.

Another example is the restructuring practice that replaces an entity from the draft resource model with an EXPRESS SUBTYPE entity to inherit attributes from a more generic concept, an existing SUPERTYPE entity. With this replacement, indirect relationships among entities are established where a direct relationship had previously existed. The pre-integration resources models can be considered "external views" of the Integrated Resource. This analogy is even more relevant if the original, pre-integrated models are considered to be the same thing as the Application Reference Models to the Integrated Resources.

The overall goal of these practices is to produce a cohesive, consistent, and structurally and conceptually stable structure. These rules, combined with the Integration Architecture constitute the integration of the STEP Integrated Resources.

4.2 Structural Integration of APs through Integrated Resources

The second type of integration is via the STEP interpretation method, which maintains a structural consistency among STEP Application Protocols (AP). This is the integration that is provided by the re-use of the Integrated Resources by Application Protocols (indicated by (2) in 4). The STEP interpretation method develops an application interpreted model (AIM), the precise specification that satisfies the needs for product data communication among product data applications. The interpretation process uses the requirements for a product data application (formally documented in the application reference model (ARM) of an AP), selects the appropriate constructs from the integrated resources, adapts the selected constructs into the application specific context.

The interpretation process consists the following steps:

- (1) Semantic Analysis: the study and understanding of the underlying meaning of the ARM requirements.
- (2) Construct Selection: the selection and copying of the appropriate resource constructs to fulfill the ARM requirements.

- (3) Construct Adaptation Analysis: the modification and addition to the selected resource constructs, application specific semantics that is required from the ARM.
- (4) Construct Mapping: the documentation of relationships between the ARM and the AIM constructs.

Interpretation rules [3] have been established to guide the use of EXPRESS language for the interpretation. The objectives of these interpretation rules are (1) to create a stand-alone specification that is self-explanatory for readability and self-contained for implementation, i.e., complete with all definitions and references; (2) to use the language syntax unambiguously, so that a given EXPRESS construct is always used to represent the same semantics; and (3) to minimize any structural changes to the baseline constructs so that AIMs are structurally and semantically integrated.

This type of integration is structural in nature, however, it focuses on the overall structure of the standard, i.e., the consistent structure among APs and the structural relationships between APs and the Integrated Resources. This type of integration does not address the internal structure of a given AIM, since all constructs used in the AIM are consistently selected from the already structurally integrated Integrated Resources. AIMs are integrated from a structural standpoint through the re-use of the same resource "vocabulary". Any AP requirements that can not be satisfied by the Integrated Resources existing constructs, are identified as resource voids to be developed. AIM can not include any additional constructs that are not part of the Integrated Resources.

4.3 Semantic Integration of APs through AICs.

The third type of integration is a semantic integration that recognizes that although Application Protocols may vary in scope, usage, life-cycle stages, common information requirements may exist. These common requirements that can be met by the same interpretation of Integrated Resources, must share the same interpreted constructs. These shared common interpreted constructs are called Application Interpreted Constructs (AICs).

This kind of integration, as indicated by (3) in 4, is an integral part of AP development. Product data applications that have overlapping information scope may use different application protocols to communicate their application specific information along with the common information. Integration of the AIMs using the AIC strategy, ensures that common AIM elements exist to facilitate the sharing of information among application systems that use different Application Protocols. AIMs are structurally integrated because they share the same interpretation baseline, the Integrated Resources. The Semantic Integration of APs using AICs ensures that the resultant adapted constructs may also be shareable between Application Protocols.

The STEP AP interpretation process focuses on maximizing the shareability of the interpretations among Application Protocols; i.e., the semantic integration of AIMs using AICs. The AIM interpretation process always uses an established AIC library as its first source for AIM development. Only upon determining that no AIC exists for a requirement, is interpretation of the Integrated Resources constructs undertaken.

4.4 Operational Integration of Implementation Schemas through APs

This type of "integration" isn't apparent or obvious, but it is the ultimate end-objective of STEP: the integration of disparate, heterogeneous systems through the use of a common language. See (4) in 4. The integration is of systems that "understand" the same AP or the same AICs; the actual implementation form is not important because the true integration is achieved through the common understanding of a collection of abstract data structures and the relationship of those structures to the implementation form and implementation schema.

In a broader context, the operational integration is achieved through a proper implementation strategy of the whole of STEP. STEP AIM specifications are the portions of STEP where implementation conformance criteria are based on.

AIMs are single EXPRESS schemas. They are mapped to a physical data representation format via a mapping algorithm specified in ISO 10303 Part 21. Part 21 specifies the format for the exchange of product data using a discrete physical file that is exchanged and processed by automated systems. But this is not the only usage of AIM, nor is it the only "implementation" of STEP.

An original objective in the development of STEP was the separation of the semantics of the specification from the syntax of the exchange/implementation format. The resulting semantic specification thus would be "implementation independent" and, theoretically, suitable for many different implementation formats. The semantic specification - the STEP Conceptual Schema - could be mapped to different implementation formats that are semantically equivalent. The result of this semantic equivalence is that not only could information be exchanged between systems using a neutral representation format, but that *different* representation forms would be semantically equivalent. Thus, integration is achieved not only by a shared data format, but by a shared understanding *across* data formats.

5 Summary

STEP has undertaken the development of a set of standards and integrated constructs that specify the precise semantics of information to be used for the effective communication among product data application systems. Two classes of integrated constructs are created within STEP: the Integrated Resources and the Application Interpreted Models. The Integrated Resources capture product data subject area requirements that serve as resources for the development of application protocols. Application interpreted models capture application requirements for information communication among product data application systems. The key methods by which these models are developed are the different types of integration processes: the Intra-resource Integration, the Structural integration of Application Protocols through re-use of the Integrated Resources, and the Semantic integration of Application Protocols through using AICs.

The development of STEP included the development of these integration methods and the overall technical architecture of STEP. In the system integration technology area, STEP has introduced a revolutionary concept to the method as well as the role of structural and semantic integration. The combination of resource integration and application interpretation provides the semantic specification of conceptual schemas necessary for an unambiguous information communication capability. STEP has in place a method to accommodate all product data applications through the integration of resource that are based on the requirements of product data related subjects. The expansion of scope for the integrated

Resources is determined by the needs of industry evidenced by the development of application protocols. STEP has in place a method that ensures utility of shared information in specific product data application contexts through the interpretation of the Integrated Resources. The re-use of existing functionality in developing application interpreted models provides information sharing within stated contexts and, where scope and requirements overlap, across application contexts.

The STEP Integrated Resources or APs are not ends in themselves, but all the components of the standard and each of the four types of integration together will meet the ultimate objective of STEP.

References

- [1] ISO DIS 10303-1 Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles. ISO TC184:Industrial Automation, Subcommittee 4:Industrial Data and Global Manufacturing Programming Languages. N193, 26 March 1993.
- [2] ISO DIS 10303-1 Product Data Representation and Exchange - Part 11: The EXPRESS Language Reference Manual. ISO TC184:Industrial Automation, Subcommittee 4:Industrial Data and Global Manufacturing Programming Languages. N151, 31 August 1992.
- [3] McKay, A. (1992) Application Interpretation Requirements Version 3. ISO TC184/SC4/WG4 N37. 10 January 1992.
- [4] ISO DIS 10303-1 Product Data Representation and Exchange - Part 21: Implementation Methods Clear Text Encoding of the Exchange Structure. ISO TC184:Industrial Automation, Subcommittee 4:Industrial Data and Global Manufacturing Programming Languages. N204, 28 May 1993.
- [5] Burkett, W.C., and Yang, Y. (1992) On the Implementation of STEP. ISO TC184/SC4/WG4 N68. October 1992.
- [6] Batini, C., Lenzerini, M., and Navathe, S.B. (1986) A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys, 18, 4, December 1986.
- [7] Danner, W.F., and Yang, Y. (1992) STEP Development Methods: Specification of Semantics for Information Sharing. ISO TC184/SC4/WG5 N30, 10 January 1992.
- [8] Danner, W.F., and Yang, Y. (1992) STEP Development Methods: Resource Integration and Application Interpretation. ISO TC184/SC4/WG5 N31, 10 January 1992.

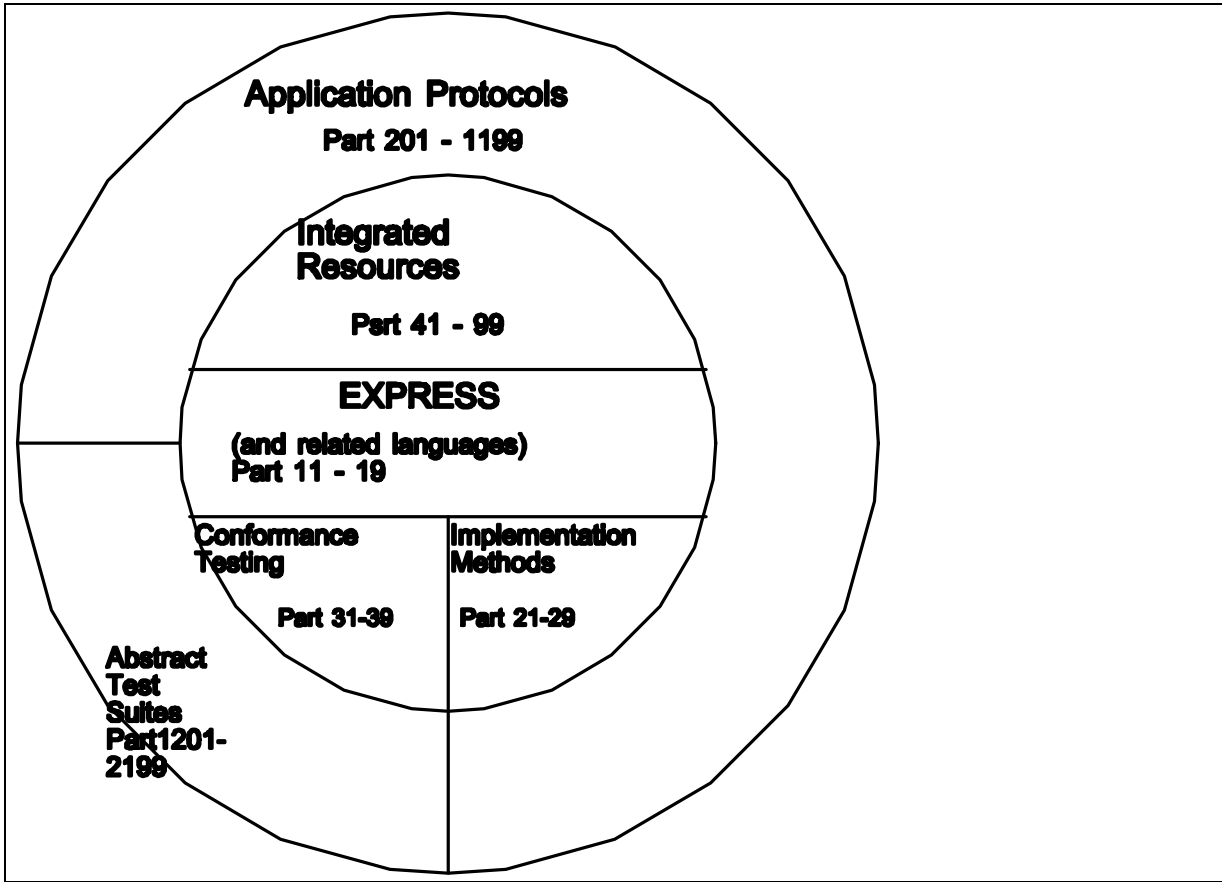


Figure 1: STEP Documentation Architecture

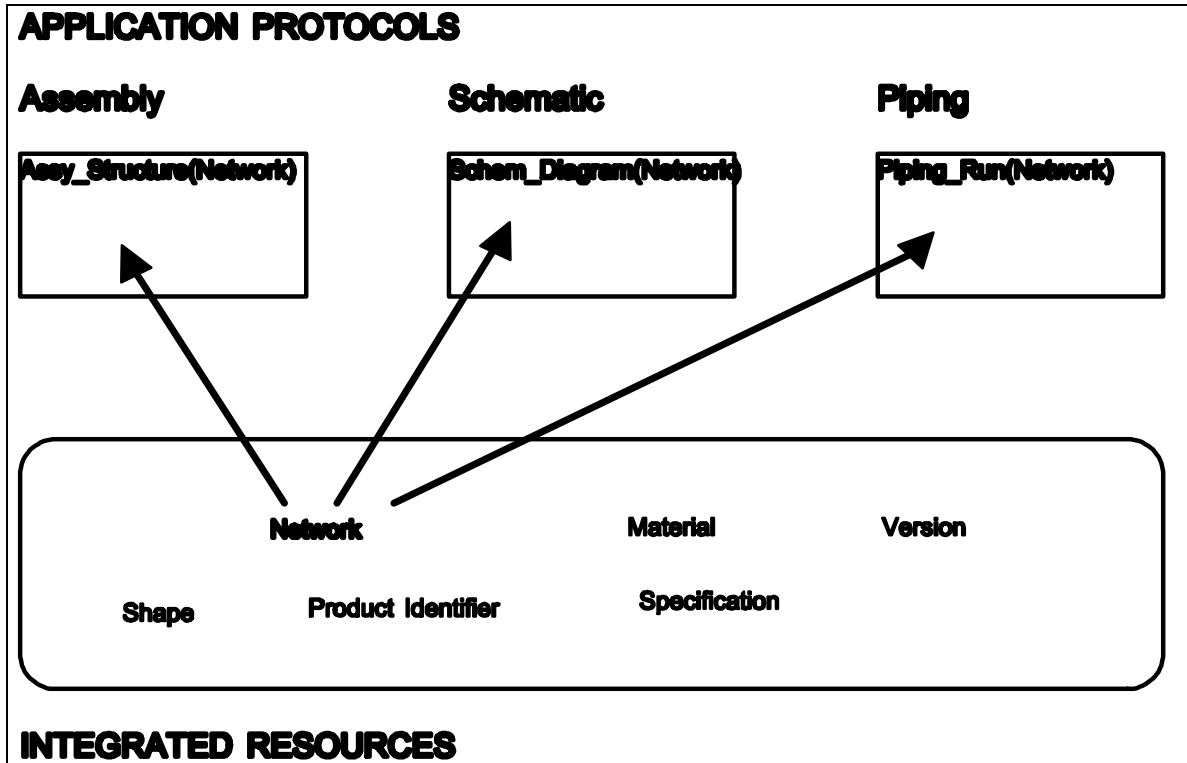


Figure 2: Relationship between Application Protocols and Integrated Resources

STEP Implementation Forms

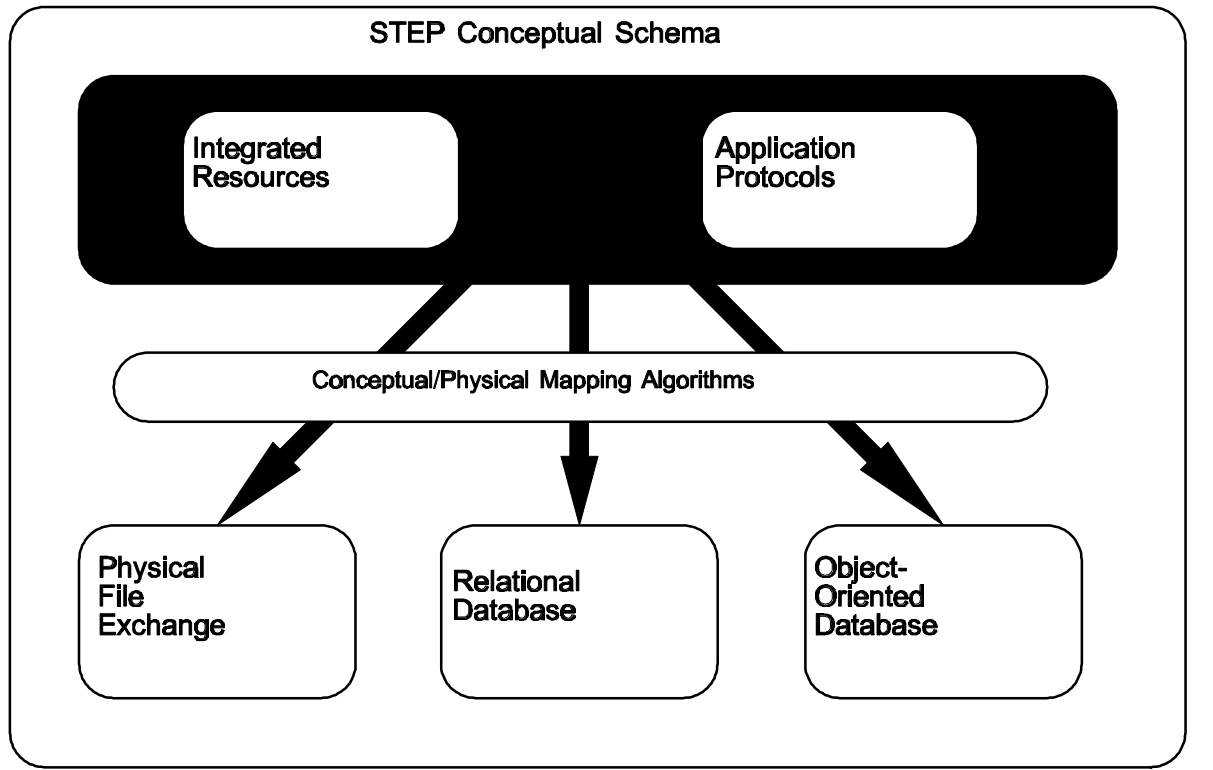


Figure 3: Relationship between conceptual schema and implementation forms

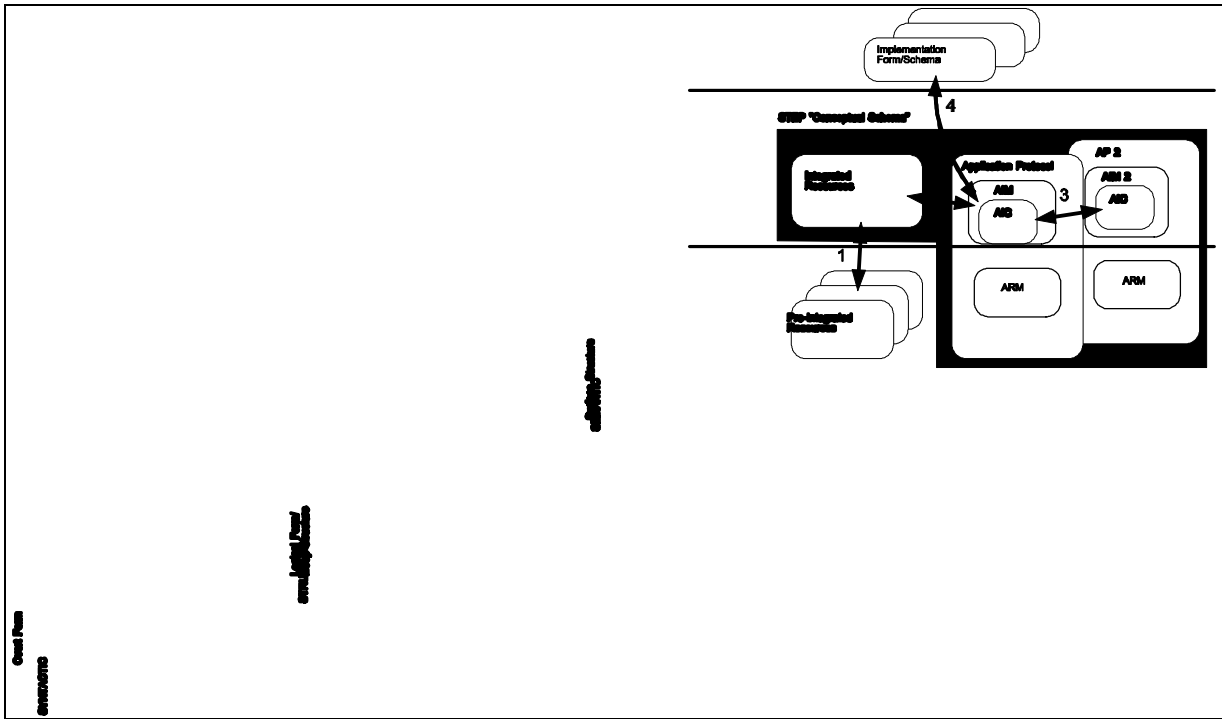


Figure 4: Kinds of Integration within STEP

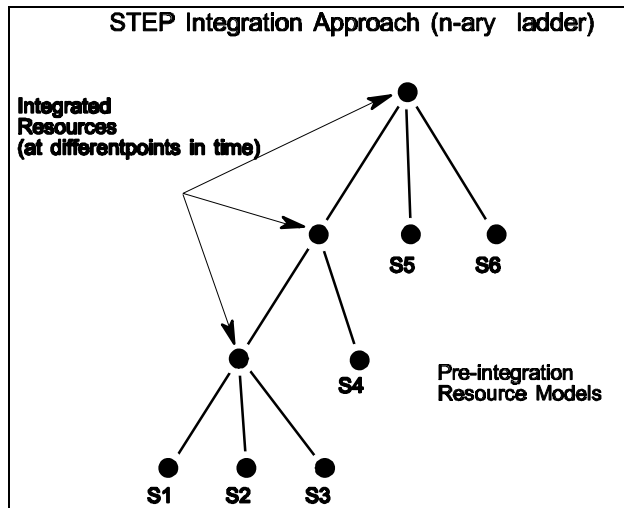


Figure 5: STEP Integration Approach