

This section contains shorter technical papers. These shorter papers will be subjected to the same review process as that for full papers.

The STEP Modular Architecture

Allison Barnard Feeney

National Institute of Standards and Technology, 100
Bureau Drive, Stop 8264, Gaithersburg, MD 20899-8264
e-mail: abf@nist.gov

The first Technical Note in this series [1] introduced the international standard ISO 10303, informally known as STEP (STandard for the Exchange of Product model data). Subsequent Technical Notes discussed various issues faced by users of STEP and how the ISO TC184/SC4 committee is addressing these issues. This paper presents the current move to modularize the STEP application protocol architecture. This paper describes the initial STEP architecture, requirements for improvements to the architecture, features of the new modular architecture, status and issues.

[DOI: 10.1115/1.1511520]

Background

STEP (STandard for the Exchange of Product model data, formally known as ISO 10303) is an international standard designed to enable the exchange of product data between heterogeneous computer systems used throughout the product life cycle. STEP is being developed under the leadership of ISO TC 184/SC4, the ISO sub-committee responsible for data exchange standards in the area of industrial and manufacturing applications [2,3]. A new modular STEP architecture addresses some deficiencies of the original architecture and is intended to speed development, facilitate implementation, and increase interoperability of applications of STEP.

This paper introduces the new modular architecture in the context of the initial architecture. First, the initial STEP architecture is described. Requirements governing the development of the modular architecture are summarized. Components of the new modular architecture are presented with an explanation of how the initial architecture is changed. The current status of the modularization effort is presented, followed by issues that remain to be addressed.

Initial STEP Architecture

The architecture of STEP is designed to support the development of standards for product data exchange and product data sharing [4]. The architecture is governed by the following concepts: (1) the scope of what is standardized and what is conformance tested is set at the level of "an application," (2) application requirements are based on a model of a business activity, (3) application requirements are standardized using natural language,

Contributed by the Engineering Informatics (Elx) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING Manuscript received August 2002; Revised August 2002. Associate Editor: R. Rangan.

and (4) a mapping is specified defining how the application requirements are satisfied using a STEP data model [5].

The key information models comprising the STEP standards are known as application protocols (APs). As illustrated in Fig. 1, an AP consists of the following major elements:

- An application activity model (AAM) describing the business process the information model supports.
- An application reference model (ARM) specifying the information requirements.
- A schema for data structures based on the SC4 common resources (CRs), called an application interpreted model (AIM), that is the basis for implementations of the standard.

STEP developers have made a conscious decision to focus on the information required by industrial processes rather than on the processes themselves as the processes may change over time while the underlying information requirements are longer lasting. This focus on information allows STEP to support data exchange, some forms of data sharing, as well as long-term data retention. To that end, the STEP community has standardized a data specification language, called EXPRESS [6], as well as implementation methods for file exchange and data access based on EXPRESS. The AIM of an AP is specified using EXPRESS. A conforming implementation of an AP must use the AIM schema in combination with one of the implementation methods. STEP standardizes conformance testing methods and abstract test suites to facilitate certification of AP implementations. An AP may specify conformance classes (CC) allowing certification of conformance to a specified subset of the AIM.

Application interpreted constructs (AICs) were introduced to the STEP architecture to enable cooperative use of multiple APs in a business enterprise. An AIC specifies a piece of an AIM that may be used to exchange product data common to two or more APs. However, an AIC does not document the common requirements or the mapping of those requirements into the AIM.

Requirements for the Modular Architecture

Industry adoption of STEP initially focused on implementing the first AP, AP203 *Configuration controlled design* [7]. As other APs with different capabilities have been standardized, it has become apparent that the "islands of APs" paradigm does not meet industry requirements for STEP. The capability of sharing the common information defined by two or more APs is an important requirement for companies to realize the full benefits of STEP. This capability is referred to as "AP interoperability." Industry is also asking for a "plug-and-play" environment where companies are able to choose from a set of comprehensive STEP capabilities to satisfy their data-exchange needs. APs are extremely large documents that take many years to standardize and are therefore not responsive to changes in industrial requirements.

Based on experiences gained through early implementation and the various attempts to address AP interoperability within SC4, a comprehensive set of requirements for the modular architecture was gathered [8]. Some of the high-level industry requirements are to:

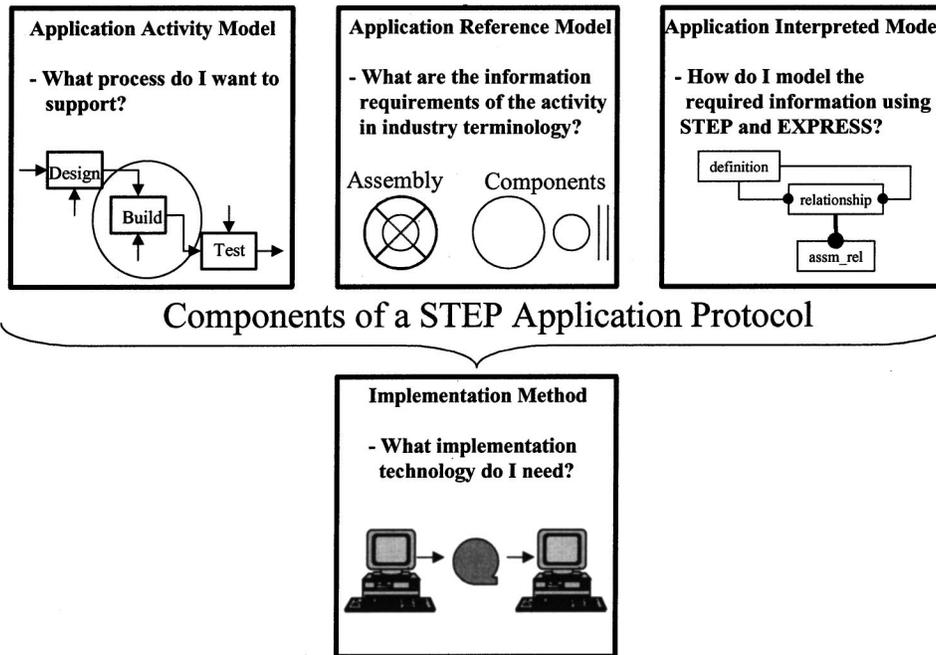


Fig. 1 Initial AP Structure.

- Reduce the high cost and lengthy time to develop an AP.
- Allow implementation of a combination of multiple APs or extension of AP implementations with additional capabilities.
- Enable application software reuse.
- Eliminate duplication and repeated documentation of the same requirements in different APs.
- Reuse data generated by an implementation of one or more APs, by an implementation of one or more different APs (AP interoperability).

Modular Architecture

The stated objective of the STEP modularization effort is “to enable the more efficient technical development, standardization, implementation and deployment of STEP standards without changing the fundamentals of the current technical architecture” [8]. The STEP architecture will continue to be based on standardizing industry requirements and mapping those requirements to a data model based on the SC4 common resources, a process known as “interpretation.”

The primary change introduced by the new modular architecture is the explicit harmonization of common information requirements. Rather than relying on harmonization occurring as a by-product of consistent interpretation across APs, module requirements are first harmonized across domains and the resulting mappings are standardized in smaller packages, known as application modules (AMs). AMs are reused by other AMs and ultimately in APs.

The implementable portion of the STEP modular architecture has two core components:

- Application module (AM) - A small, reusable data specification documented with an ARM (application reference model); mapping; interpreted schema; and usage guide.
- Application protocol (AP) - The use of a data specification to meet the requirements of some business process.

The objectives and function of the architectural components are described below.

Application Module. An application module is designed to maximize reusability of the harmonized requirements and the as-

sociated interpretation into the SC4 common resources, the data specification, and thus, software implementations. Modules can be referenced by other modules rather than be copied. Modules support reusability by the standards developer, implementer, and user.

Application modules replace AICs in the STEP architecture. Though the modular architecture does not prohibit the use of AICs as common resources for application modules, in practice, AMs have been created to replace AICs. The objectives of AICs and AMs are similar. They both standardize interpretation results for reuse in multiple APs. However, AICs and AMs are created differently, have different audiences, and have different content. An objective of modularization is to document a concept one time and then to directly reuse that concept in other modules. An AIC is only created when a concept has already been documented in two or more APs. AICs and CRs were both designed to be used by the standards developer. AMs and APs are also designed for industrial customers who evaluate the capabilities of standards for use in their organizations and for software developers who implement the standards. AMs, unlike AICs, contain harmonized information requirements and specifications of the mappings of those requirements to the CRs.

An application module may be as generic as a common resource or as application context-specific as an AP. No requirements have been set on the size or level of granularity of an AM. Each AM will define an information model for one or more concepts. Where a module requires concepts defined in other modules, it will reference the other modules. The links between modules form a directed, acyclic graph. Three categories of modules have been proposed. *Foundation modules* are characteristically generic for reuse, have no scope overlap, avoid use of global rules, and are considered part of the SC4 common resources (and often merely add a requirements model and a mapping to constructs defined in the SC4 common resources). *Application modules* are built from foundation modules and common resources, add EXPRESS constructs beyond those referenced from the CRs, contain local rules to reflect project requirements, and may overlap in scope with other AMs. An *AP module* is the top-level module that provides the data specification for a modular AP. These

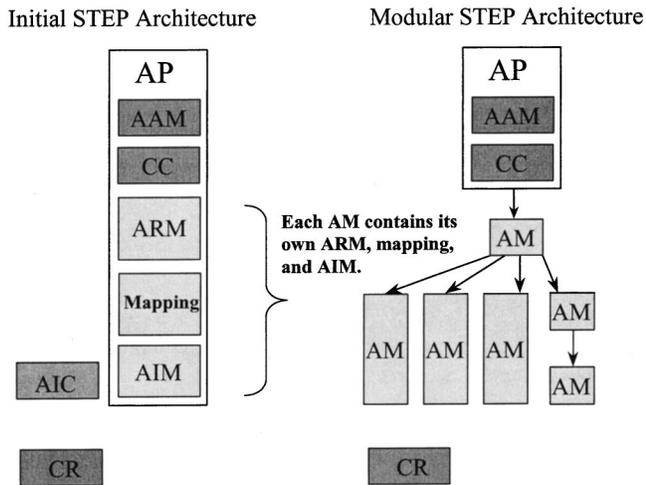


Fig. 2 Modular Architecture vs. Initial STEP Architecture.

categorizations are primarily to distinguish internal SC4 management practices. All modules have the same content and follow the same ballot process [9].

The modular approach calls for the use of EXPRESS rather than natural language for the documentation of requirements, i.e., the ARM, in an application module. This allows the use of tools to validate the dependencies between application module ARMs. It has the added benefit of allowing for the future use of an in-development EXPRESS mapping language called EXPRESS-X [10] that will make the requirements-to-resources mapping computer interpretable as well.

Application Protocol. A modular AP is a documented use of an application module for a specific business process. A single, normatively referenced AM is the data specification for the AP. This AM uses other normative AMs to provide the documentation of its requirements and interpreted model. The top-level AM may include specific business process rules or constraints. An AM that serves as the top-level AM for a modular AP may also be used in another modular AP, should the second AP completely encompass the scope of the first AP.

The proposed content for an AP document includes an activity model and conformance class definitions. Industry terminology mappings from the generic AM terminology may be defined in an AP to make it more understandable to reviewers from the application domain [11]. The content of a modular AP is a matter of current debate. Inclusion of a graphical EXPRESS ARM and a complete EXPRESS AIM is generally considered to be necessary to support implementation. In Fig. 2, the initial STEP architecture is contrasted with the modular STEP architecture.

Status Of STEP Modularization

The STEP modular architecture was approved as an ISO TC 184/SC4 work item in 1999. To meet standardization requirements set out for the architecture, SC4 decided that application modules will be published as ISO technical specifications, an ISO deliverable requiring only one ballot cycle prior to publication.

Guidelines. Today, three documents detailing aspects of the modular architecture, including rules for specifying the content of the architectural components [11–13] have been provisionally approved. A draft set of guidelines was balloted in SC4 but the final guidelines have not been published.

Development Environment. Modules are being documented in the Extensible Markup Language (XML) [14] and managed in a Concurrent Versions System (CVS) [15] repository on SourceForge [16]. Webs of related modules may be published in Hyper-

Text Markup Language (HTML) with hot links traversing the many interrelationships between the modules. This sophisticated publishing environment is needed to manage the many related documents and their many references to each other and to the SC4 common resources. The module repository currently supports AMs; the AP capability is being developed.

AMs. Nine application modules have been approved and published to date. These modules provide the capability to assign appearance to shapes and to assign geometric elements to layers. 61 modules that support product data management and geometry are undergoing a confirmatory ballot within SC4 and ISO publication is expected in early 2003. More than 300 module part numbers have been assigned to projects planning modular APs in the areas of engineering change, engineering analysis, geometric dimensioning and tolerancing, and product lifecycle support, among others.

APs. No modular APs have been published yet. A subset of the first 70 modules will be used in the first modular AP, the second edition of ISO 10303-203. AP203e2 is under development even while SC4 is working to determine the ideal content for a modular AP.

EXPRESS. The modular approach required modifications to the EXPRESS language to solve problems that arise from specifying a model across multiple subdocuments. These modifications have been agreed upon and are due to be published in the final draft of the second edition of ISO 10303-11, expected in late 2002. These changes to the EXPRESS language provide the ability to define structures that may be extended and constrained outside of the schema in which they are defined.

Harmonization Team. A harmonization team was officially created by SC4 resolution in June 2000, but no resources have been allocated and the team is not active.

Technical Issues

There are a number of issues and unanswered questions regarding modularization. Some primarily affect SC4 standards developers, yet others affect the users and implementers of SC4 standards.

Guidelines. The lack of stable, consensus guidelines documents governing modularization makes it difficult for the projects developing modules to do so in a consistent manner, and delays development of tools. The guidelines document governing content of an AM is fairly stable (though does not quite match the proposed content provisionally approved by SC4), but the guidelines document prescribing content of a modular AP is not. The stability of the guidelines reflects the amount of experience SC4 has had with the different types of documents.

Development Environment. Use of the sophisticated modules development environment requires technical savvy on the part of the standards developer. Modules published to date have been produced by one project, working closely with the repository developers. For modularization to be successful, all modules must reside in the repository. Further, the scalability of the module repository may raise issues in the future—when more projects begin using the repository to develop modules SC4 will need rules and procedures to govern their interaction.

Harmonization. The harmonization team identified in the development of the architecture was established, but resources to staff the team have not materialized. In the absence of a harmonization team, there is no impartial advisory group to facilitate requirements harmonization, or to advise SC4 on the level of consensus or harmonization of modules submitted for ballot. There is no group with a comprehensive knowledge of existing modules to advise projects on their appropriate application. There are no documented principles governing requirements harmonization. In

short, there are no checks and balances on the development of modules other than the ISO ballot process that, in the case of modules, consists of only one ballot cycle.

Granularity. No guidance has been given to AM developers regarding the appropriate level of granularity for AMs. A balance needs to be reached between reusability and usability. Many of the AMs are at a very fine level of detail to maximize reusability. It has not yet been demonstrated that the foundation modules provide value beyond that gained by simply using the SC4 common resources: 1) there is a not insignificant cost associated with creating, documenting, and maintaining a module; 2) the requirements models and mappings provided in foundation modules are often no more informative than the common resources themselves; and 3) the foundation modules have so many interdependencies that by using one foundation module you are committing to using several. Further, the fine-grained requirements of foundation modules are aggregated in the top-level module for an AP. In traditional APs, the ARM was modeled at a more coarse level of detail in general, only providing fine-grain detail in areas of particular interest to the AP. Perhaps a better model would be to create fewer modules with more context and less reusability—modules that are interesting at the requirements level.

APs. The content and presentation of modular APs are not yet finalized. SC4 has an opportunity to leverage implementation experience to design and deliver APs that best meet the needs of STEP users and implementers. The currently proposed content of the modular AP is very similar to the traditional AP, but is there other information or other means of presenting the same information that would be of higher utility? XML technology provides enormous flexibility to deliver not only what ISO requires, but what may be more important to any given audience (e.g., a sorted entity.attribute listing would be a useful view). Modular APs may be delivered as an HTML web of interrelated modules that can be easily navigated by the reader. However, navigation through many layers of modules is a painful way to visualize the “big picture” and some implementers are asking for a paper document to print and put on their bookshelves. It remains to be seen whether SC4 will produce guidelines for modular APs that will result in presentation of an ARM that serves the purpose intended for ARMs: be a useful scope and requirements development tool for AP developers; adequately present the AP scope to industry reviewers; and provide explanation of the AP AIM, through the mapping specification, for implementers.

EXPRESS. At this point, EXPRESS edition 2 implementations are in experimental stages, but a mapping between EXPRESS edition 2 constructs and EXPRESS edition 1 exists as a workaround. Some SC4 implementation method standards may require subsequent editions to take advantage of the new capabilities in EXPRESS edition 2.

EXPRESS-X. The modular approach is designed to take advantage of the coming EXPRESS-X capabilities. EXPRESS-X is

a STEP description method for mapping between two related EXPRESS schemata. The modular approach would like to leverage this new capability for mapping between the requirements and interpreted model schemata. How that will happen has yet to be completely resolved.

Conformance Requirements. Changes may be made to the STEP conformance testing methods to support the new modular architecture. The way conformance classes are specified may also need to be changed for modular application protocols.

Summary

The STEP modular architecture was created to enable more efficient development, standardization, implementation and deployment of STEP. The architecture is an improvement over the original. Projects now have the flexibility to produce monolithic APs, as before, or divide their scope in any way they determine will maximize reusability. It remains to be seen whether SC4 will succeed in applying the architecture to its advantage. We will not know that until we see how efficient, reusable, and interoperable the first set of modular APs is proven to be.

References

- [1] Pratt, M.J., 2001, “Introduction to ISO 10303—the STEP Standard for Product Data Exchange,” ASME J. Comput. Inf. Sci. Eng. 1, (1), pp.102-103.
- [2] SC4 On-Line Web Page: <http://www.tc184-sc4.org>.
- [3] ISO TC 184/SC4 Web Page: <http://www.iso.ch/>.
- [4] ISO 10303-1:1994, Product data representation and exchange—Part 1: Overview and Fundamental Principles.
- [5] Kemmerer, S. (ed.), 1999; *STEP the Grand Experience*, NIST Special Publication 939, US Government Printing Office, Washington, DC 20402.
- [6] ISO 10303-11:1994, Product Data Representation and Exchange—Part 11: Description Methods: The EXPRESS Language Reference Manual.
- [7] ISO 10303-203:1994, Product Data Representation and Exchange—Part 203: Application Protocol: Configuration Controlled Design.
- [8] Price, D. M., “WG10 STEP Modularization Requirements Specification,” ISO TC 184/SC4 WG10 N227, 1998-12-14. Available on-line at http://www.nist.gov/sc4/wg_qc/wg10/current/n227/wg10n227.htm.
- [9] Dunford, J. “Categorisation of Application Modules,” ISO TC 184/SC4 WG12 N1031, 2002-03-25. Available on-line at http://www.tc184-sc4.org/SC4_Open/SC4_and_Working_Groups/WG12/N-DOCS.
- [10] ISO/DIS 10303-14:2002, Product Data Representation and Exchange—Part 14: Description methods: The EXPRESS-X Language Reference Manual.
- [11] “Guidelines for the Content of Application Protocols Using Application Modules,” ISO TC 184/SC4 N1113, 2001-05-02. Available on-line at http://www.nist.gov/sc4/howto/methods/modules/ap_content/.
- [12] “Guidelines for the Content of Application Modules,” ISO TC 184/SC4 N1161, 2001-05-02. Available on-line at http://www.nist.gov/sc4/howto/methods/modules/am_content/.
- [13] Barnard Feeney, A. “Application Module Development Points in the Application Protocol Development Process,” ISO TC184/SC4 WG12 N1312, 2002-07-02. Available on-line at http://www.tc184-sc4.org/SC4_Open/SC4_and_Working_Groups/WG12/N-DOCS.
- [14] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation. 2000-10-06. Available on-line at <http://www.w3.org/TR/REC-xml>.
- [15] Cedergvist, P. et al. “Version Management with CVS.” Available on-line at <http://www.cvshome.org/docs/manual/>.
- [16] STEP Module Repository Web Page: <http://stepmod.sourceforge.net>.