53/CW/IN/61

1999 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

STEP: WHAT IS IT AND SHOULD IT BE USED FOR KSC'S ISE/CEE PROJECT IN THE NEAR FUTURE?

Dr. Catherine C. Bareiss
Associate Professor
Computer Science
Olivet Nazarene University

Mike Conroy
NASA/KSC

ABSTRACT

The ability to exchange information between different engineering software (i.e. CAD, CAE, CAM) is necessary to aid in collaborative engineering. There are a number of different ways to accomplish this goal. One poplar method is to transfer data via different file formats. However this method can lose data and becomes complex as more file formats are added. Another method is to use a standard protocol. STEP is one such standard. This paper gives an overview of STEP, provides a list of where to access more information, and develops guidelines to aid the reader in deciding if STEP is appropriate for his/her use.

STEP: WHAT IS IT AND SHOULD IT BE USED FOR KSC'S ISE/CEE PROJECT IN THE NEAR FUTURE?

Dr. Catherine C. Bareiss

1. Introduction

A. Different methods for information exchange

This paper looks into one specific way to exchange information between different engineering software packages. This problem is similar to the problem of information exchange between different office productivity software (i.e. word processors, spreadsheets, presentation software, databases). Even though the data for engineering models is much more complex than that of office productivity software, much can be learned by looking at the different methods of office productivity software information exchange.

The easiest way to exchange office documents is by using the same version of the same software. This is often done within a company that has chosen to standardize on one package. When using one vendor (i.e. Microsoft, Lotus, Corel, etc.) another method for exchange of information is found between different types of software (i.e. spreadsheets, word processing, etc.). Platforms developers have designed methods to make it very easy to access information in one document (for example, a spreadsheet) in another (for example, a paper created in a word processor). A third way to exchange information is used when working with different platforms. The developers often include routines that allow their package to read data created by other packages. A fourth way involves total transparent information interchange and can be seen in database packages. Platform independent standards (i.e. SQL and ODBC) have been developed, and most database packages have implemented these standards.

Options for information interchange in engineering packages are similar. The easiest option is for an organization to standardize on one package. However, as in office software, it is common to need different packages to work on different aspects of engineering modeling (i.e. word processor and spreadsheets, mechanical and electrical engineering). The next easiest option would be to standardize within an organization on software developed by the same developer that allows for information interchange. However, such software is rare, and the politics are such that it may be difficult to require all engineers associated with an organization to use software from the same developer. The next option would be to use packages that include routines to read other data formats. But as in the case of office software, it is usual that detailed information can be lost. Also, these can get to be a very large package because as each new file format is added to the conversion routines, all previous file formats must be allowed to convert to and from the new one. Very quickly, this becomes too complex and large. In addition, upgrading versions can often cause problems with transferring the more complex data that the newer versions allow.

The fourth option (and the one presented in this paper) is to use a platform independent standard for information interchange. The three most common are ORB, CORBA, and STEP. The first two are generic standards for any type of communication protocol. STEP is an international standard designed to facilitate the exchange of information specifically for engineering and product design, and is the one covered in this paper. While this option seems the best, it is important to keep in mind that STEP is still not finished and not completely implemented commercially. This may require additional on-site programming to support the exchange between different software packages depending on the needs of an organization.

B. Overview of STEP

To better understand if STEP is worth any additional required effort, STEP must be better understood. STEP (STandard for Exchange of Product model data) is an ISO standard (ISO 10303 industrial automation systems - product data representation and exchange). STEP is a very large and complex standard. While it is very difficult for any one person to completely understand all of STEP, the components that are necessary to determine if it can meet KSC's ISE/CEE immediate needs can be understood by a single individual.

There are four major components of STEP that are of interest: integrated resources (IRs), application protocols (APs), application modules (AMs) and EXPRESS (a description method). An AP defines the protocol for describing all the information associated with a given area (i.e. electrical) of engineering. APs are referred to as parts 201-299. Each AP is made of one or more IRs. The IRs are used to describe "components" that might be of interested in one or more AP's (for example, the shape of an object). IRs are further divided into 2 categories. Parts 41-99 are integrated generic resources whereas parts 101-199 are integrated application resources. Generic resources are independent of the applications and can reference each other. Application resources can reference generic resources and are used to add additional constructs that might be used by a group of similar applications. AMs are used to help the sharing of information between APs and the speed up the development of APs. They are a new concept are still under development. EXPRESS is the data definition language for STEP and is used to define that structures of APs, AMs, and IRs. It is the first of the description methods (referred as parts 11-19). The other parts of STEP include Part 1 (which is the overview), parts 21-29 (implementation methods), parts 31-39 (conformance testing methodology and framework), parts 301-399 (abstract test suites), and parts 501-599 (application interpreted constructs) which are components of application protocols.

C. Guide to the rest of the document

The next section will introduce STEP by covering its history. The second section explains the contents of STEP that might be of importance to KSC's ISE/CEE project. It includes a list of acronyms and terms are given to help understand the paper and information to help understand integrated resources, application protocols, application modules, and EXPRESS and implementation methods. The third section talks about what is involved in PDM (product data management) software and how STEP fits in that picture. The fourth section (which is the conclusion) is designed to help the reader decide if STEP would help meet the immediate and short term needs of the KSC's ISE/CEE project.

2. History of STEP

Work on STEP was officially started in 1984. It was proceeded in the 1970s by three specifications: IGRES (Initial Graphics Exchange Specification) in the USA, SET (Standard D'Exchange et de Transfert) in France, and VDA-FS (Verband der Automobilindustrie-Flachen-Schnittstelle) in Germany.

STEP started with the following objectives.

1. The creation of a single international standard, covering all aspects of CAD/CAM data exchange.
2. The implementation and acceptance of this standard in industry, superseding various national and de facto standards and specifications.
3. The standardization of a mechanism for describing product data, throughout the life-cycle of a product, and independent of any particular system.
4. The separation of the description of product data from its implementation, such that the standard would not only be suitable for neutral file exchange, but also provide the basis for shared product databases, and for long-term archiving.

The initial release of STEP occurred in 1995 with the following twelve parts.

Part 1	Overview and Fundamental Principles
Part 11	The EXPRESS Language Reference Manual
Part 21	Clear Text Encoding of the Physical File Exchange Structure
Part 31	Testing Methodology and Framework: General Concepts
Part 41	Fundamentals of Product Description and Support
Part 42	Geometric and Topological Representation
Part 43	Representation Structure
Part 44	Product Structure Configuration
Part 46	Visual Presentation
Part 101	Draughting
Part 201	Explicit Draughting
Part 203	Configuration Controlled Design

3. Overview of STEP

A. TERMS

The following are a list of acronyms and terms that will be of help for understanding the rest of the paper.

AAM	Application Activity Model
AIC	Application Interpreted Construct
AIM	Application Interpreted Model
AM	Application Module - an extension of an AIC
AP	Application Protocol
API	Application Program Interface
ARM	Application Reference Model
CEE	Collaborative Engineering Environment
ER	Entity-Relationship
EXPRESS	A data definition language defined in STEP
EXPRESS-G	A graphical standard (defined in STEP) to represent a portion of EXPRESS
IR	Integrated resource
IAR	Integrated Application Resource
IGR	Integrated Generic Resource
ISE	Intelligent Synthesis Environment
ISO	International Standards Organization
PDM	Product Data Management
SDAI	STEP Data Access Interface
STEP	STandard data Exchange Protocol

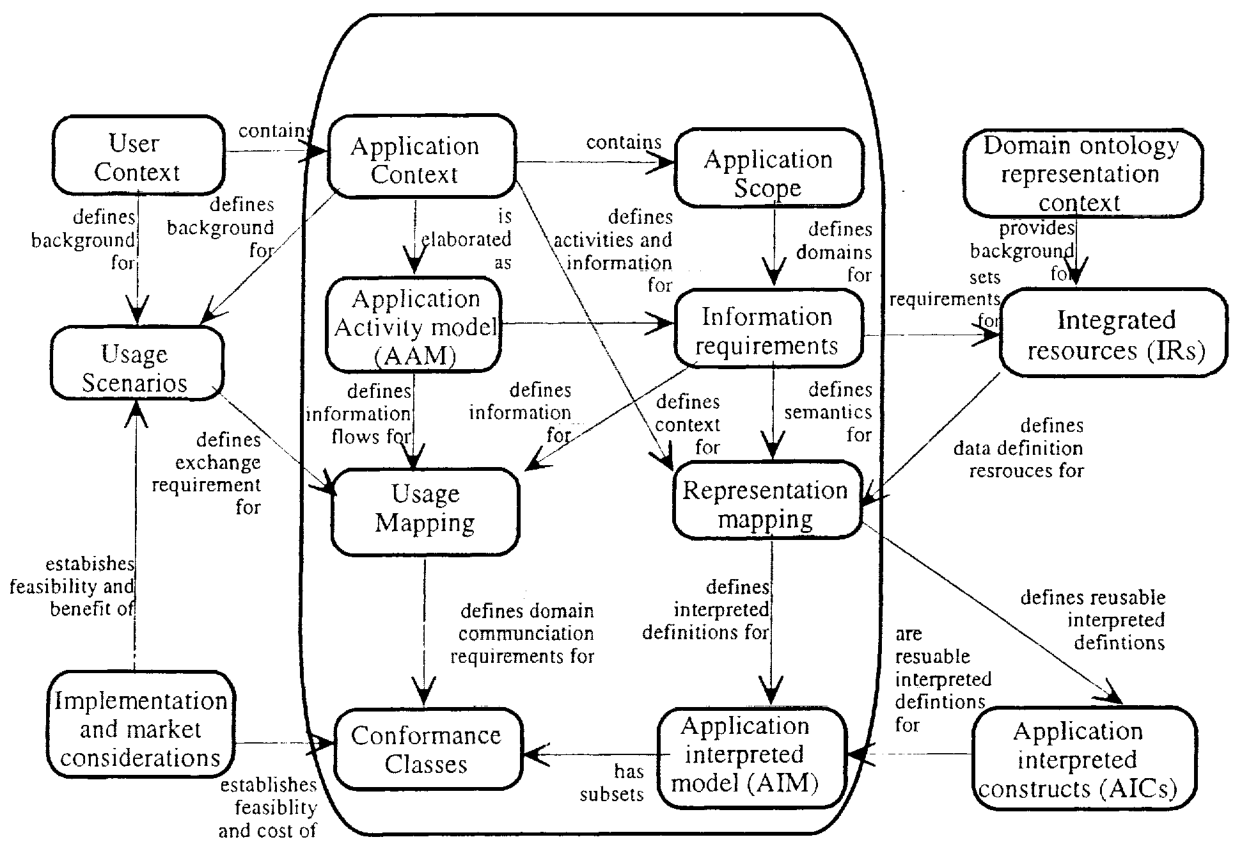
B. Integrated Resources

Integrated resources are used to describe components that might be of interest to one or more APs. There are two categories of integrated resources. Generic resources are application independent, and application resources support the needs of a group of applications. The integrated generic resources support common themes such as geometric and topological representation, materials, visual presentation, and process structure and properties. Integrated application resources include draughting, finite element analysis, and kinematics. While IRs are used to develop APs, they are not sufficient by themselves to meet the needs of an application and are therefore not used by themselves but are only used as part of an application protocol.

C. Application Protocols

Application protocols are the primary way to interface with STEP for most users. APs define the requirements for a specific application of product data related to a particular industry. They do this by combining both integrated generic resources and integrated application resources. The numbering systems allows for ninety nine different APs with thirty two being worked on, finalized, or retired. As industry needs expand, more APs will most likely be developed with each AP possibly requiring over one year to be developed.

APs are more complex than just identifying which integrated resources to use. They start with one or more application contexts which contain the descriptions of the functionalities, technologies, types of product, disciplines, industry sectors, and life cycle stages that comprise the background knowledge of users. The AAM (application activity model) elaborates the activities and flow of information between activities for given application context. These activities and flow of information are identified by the application scope. Both the application scope and AAM are used to define



the information requirements of the users of a particular AP. These requirements include the natural language definitions of the application objects and application assertions using the terminology of the users. These requirements can be displayed graphically via an application reference model (ARM).

The AAM, information requirements, and usage scenarios are combined to define usage mappings. These mappings identify which activities, information flow, and requirements are needed by each usage scenario. The usage mappings are then used to define conformance classes which group specific information requirements based upon usage scenarios that describe how the information is expected to be used. These classes are also influenced by implementation and market considerations.

A representation mapping combines the application contexts, information requirements, and IRs to create an application interpreted model (AIM) which is the formal specification for communication when satisfying a conformance class of an AP. Groups of AIM constructs may represent a common semantic in more than one AIM. These are combined into reusable application interpreted constructs (AIC). An AIC satisfies information requirements that are common to more than one AP.

D. Application Modules

Application modules (AMs) extend the AIC (application interpreted construct) concept by including the relevant portions of the AP application reference model. This extension was done for five reasons.

1. To reduce the high cost of developing an application protocol. (It currently takes a team of four or more people about a year and a half to produce an initial AP specification for a typical engineering function.)
2. To ensure the ability to implement a combination of subsets of multiple APs or to extend existing APs to meet a business need.
3. To ensure the ability to reuse application software developed to support one AP in the development of an implementation of another AP with the same or similar requirements.
4. To avoid duplication and repeated documentation of the same requirements in different application protocols leading to potentially different solutions for the same requirements.
5. To ensure the ability to reuse data generated by an implementation of one or more APs by an implementation of one or more different APs.

There are three major components to an application module.

1. the scope and functional requirements;
2. the application reference model as a representation of the application domain information requirements; and
3. the module interpreted model that specifies the required use of the common resource.

The use of AMs is relatively new to STEP and is just starting. No standards have yet to be produced using AMs but they are being worked on (esp. in terms of PDM applications). Any future development in APs should also develop and use AMs.

E. EXPRESS and Implementation Methods

i. EXPRESS

Express is a data definition language that is describes the structure of the IR's, AP's, and AM's (including the data structure and constraints) and is used in a number of standardization projects including STEP. It does not contain the actual data, nor is it executable.

EXPRESS supports the following modeling capabilities:

- definition of data entities, attributes, and relationships,
- the specification of local and global constraints on these, and

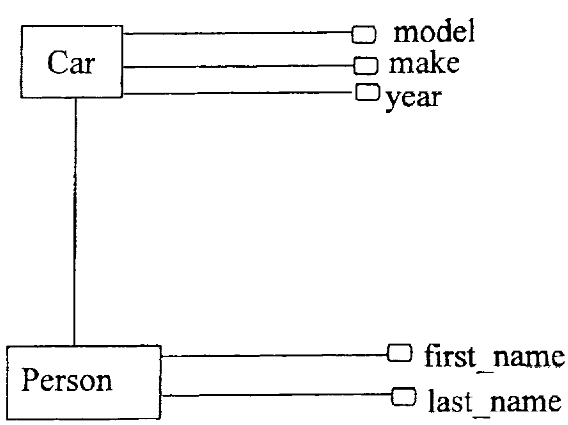
- the collection of data definitions and constraints in separate schemata, supporting modular development of data models.

An example of a data model in EXPRESS might include the following:

```
ENTITY car;
        make    : STRING;
        model   : STRING;
        year    : INTEGER;
        owner   : person;
END_ENTITY;
ENTITY person;
        first_name      : STRING;
        last_name       : STRING;
END_ENTITY;
```

EXPRESS has seven constructs: schema, type, entity, constant, function, procedure, and rule. A complex EXPRESS schema (which is very typical) can be very hard to follow, so EXPRESS-G was developed and displays a subset of EXPRESS graphically.

EXPRESS-G is similar to ER diagrams in that it has definitions (similar to entities), relationships, and compositions (which allow a diagram to span more than on page). It does not support functions, procedures, or rules. It supports simple data types, named data types, relationships, cardinality, and one or more schemas. See following diagram for an example.



ii. Implementation Methods

Currently there are six implementation methods defined. The first is a physical file exchange structure. This method uses EXPRESS-I (instantiation language) and records this is a plain ASCII file. Below is the data portion for an instance of a car.

#1 = CAR ("Saturn", "SL", 1996, #2);
#2 = PERSON ("John", "Doe");

While this method is sufficient (can support all file exchange), it does have a number of disadvantages.

- It can be very slow for accessing specific information.
- It does not support concurrent access.
- It does not support distributed storage.
- It can be difficult to share information between different AP's (since the data may be structured differently and software would be needed to converted to the structure of the other AP).

Part 22 defines SDAI (STEP Data Access Interface) which defines how to interface with advanced data storage methods (i.e relational databases, object-oriented database, knowledge bases, etc.). It defines the methods used by an application to access the data. It is then up to the data storage method to support these methods. The last four parts (23, 24, 25, 26) of the implementation methods define bindings for languages. Part 23 defines the C++ language binding to the SDAI, while part 24 defines the C language binding and part 25 defines the late FORTRAN binding. Part 26 defines the interface definition language binding to SDAI. These are used to support language access to the STEP data. They could be used when developing necessary code to meet any needs that cannot be met via commercially available products.

4. PDM

The purpose of a Product Data Management system(PDM) is to help an organization track the information associated with a product throughout its entire life-cycle. This can include multiple versions, views of parts and/or products, managing the hierarchy of a product composition, tracking the status of parts and/or products, and/or managing the information associated with different variations of a part and/or product.

The field of PDM software is still be refined and researched. Some of the still open questions include:

1) What is the theoretical required information to be managed?
2) Since different organizations have different life-cycles for products, what is the best way to design a system to allow for these differences and still ensure that it will support all needs?
3) How should PDM systems be integrated with design software (i.e. CAD, etc.)?

Since the problem of PDM systems have not been completely solved, if one wants to use such a system, there are a number of questions to ask.

1) How does my organization manage the life-cycle of a product?
2) What information is important to be tracked?
3) What software can best meet answer one and two?
4) What software can best interface with other software (i.e. CAD) being used to developed the product?

The fourth question is where STEP plays a role. Within the STEP community there is the PDM Implementor Forum. This form is made up of software developers and testers who are working on PDM systems and translators based on STEP. This form is called PDES, Inc. consortium and was started in February of 1995.

The STEP PDM Schema (as of May 19999) contains the intersection of AP 203, AP 212, AP 214, and AP 232. It includes the following units: part identification, part classification, part properties, part structure and relationships, document identification, document classification, document and file properties, document structure and relationships, external files, document and file association to product data, alias identification, authorization, configuration and effectivity information, and work management data.

5. Conclusions

A. What options exist to meet KSC's ISE/CEE's immediate and short-term needs?

When trying to decide if STEP should be used for the immediate needs of the ISE/CEE environment at KSC, there are a number of issues to look at. For each alternative one needs to consider the costs and benefits. The options are to:

1. Standardize on one software package
2. Use existing protocols to convert to standard file formats
3. Develop own information exchange protocols for limited needs
4. Use STEP and develop own software to fill in the gaps.

It is assumed that options 1 and 2 will not meet the immediate requirements of ISE/CEE. If either one does, then there is no cost and that option should be used. This leaves two options: total internal development or the use of STEP.

B. Why one might not use STEP.

There are three primary disadvantages with using STEP: training, development costs, and complexity with the complexity being the primary cause of the other two. Because of this complexity, the system developers will need to invest a significant amount of time in becoming proficient in the use and development of APs along with the remaining parts of STEP. Also, if the existing software and/or APs do not meet all the current requirements, there will be a lot of time and effort necessary to develop these. Because one would be developing generalized software and APs instead of ones designed to meet the exact needs of KSC's ISE/CEE, more effort would be required.

C. Why one might use STEP.

There are four major advantages of using STEP: already developed application protocols, investment in the future, a large amount of international commitment, and the existence of commercial products.

With STEP, the application protocols have either already been developed or are in the process of being developed. The process of developing APs (either for STEP or the equivalent of own software) is a large, time-consuming task. It involves correctly understanding the application domain, identifying all requirements in that area, creation of application protocols, and testing the final results. There are many places in this process where errors can develop. By using STEP, either the application protocol is finished (and "error-free") or is being developed by a "large" group of people investing in this process. This would tend to make the development of the application protocols more error-free in STEP then doing it by own self given the complexity of the task. In addition, the cost of develop of APs should be decreasing as the development and use of AMs increase. A presentation by Julian Fowler from PDTSolutions in 1997 also indicated that "AP interoperability" is still perceived as a problem. This is where a good deal of current effort in STEP is being focused.

In May of 1999, the NASA CIO officially approved NASA-STD-2817, which includes the requirement for CAE/CAD/CAM systems used by NASA to have interchange tools that support STEP. This includes making sure that tools to enable data interchange (which are compliant with STEP, currently AP 203, AP209, AP210, AP 225, and AP 227)) be available to all CAE/CAD/CAM. users at each NASA center. In addition NASA is a member of PDES, Inc., which is a join industry/government consortium specifically formed to accelerate the development and implementation of STEP. It appears as if NASA's future includes a strong commitment to STEP. Therefore any investment in STEP now (both learning and development of systems) is an investment in the future.

There is a large international commitment to STEP (over 25 nations), in addition to a significant commitment from vendors and industry (a total of more than 200 companies involved). This also indicates that the initial investment in time and energy by KSC would not be wasted.

D. How to decide.

So with these advantages and disadvantages, should KSC use STEP to meet the immediate and short term needs of the ISE/CEE? How should KSC go about answering that question?

Step 1: Identify the needs. This includes identifying the CAx software used and the related APs in STEP.

Step 2: Determine the status of the APs. Are they still under development?

Step 3: Identify if there is existing software that implement the necessary APs and what software exists that can help with data interchange that is not based on STEP. These steps will identify how much can be achieved via "off-the-shelf" vs. internal development, and how much work is involved in using STEP vs internal development vs other commercially available methods.

Step 4: Make the decision. After determining how much would need to be developed on-site, the decision on using STEP can be made based upon the one wishes to invest in the future vs. the need for immediate solutions.

References

Danner, William F., "Developing Application Protocols (APs) Using the architecture and methods of STEP (STandard for the Exchange of Product Data). Fundamentals of the STEP Methodology", NISTIR 5972, Jan. 1997.

"Guidelines for application module development, Revision 0.4: Proposed ISO TC 184/SC4 Standing Document", 1997, ISO TC 184/SC4 Nxxx

Kindrick, Jim and Markus Hauses, "Usage Guide for the STEP PDM Schema", May 27, 1999, http://www.pdm-if.org.

"Product Data Exchange Using STEP (PDES) Part 1 – Overview and Fundamental Principles", ANS US PRO/IPO-200-001-1994

"Product Data Exchange Using STEP (PDES) Part 11 – The EXPRESS Language Reference Manual", ANS US PRO/IPO-200-011-1994