STEP for Data Management, Exchange and Sharing

Julian Fowler

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN 1 871802 36 9

First published in Great Britain 1995

Technology Appraisals

Technology Appraisals is a leading independent provider of high quality information, education and training for users and vendors of computer communications.

For further information please contact:

Technology Appraisals Ltd., 82 Hampton Road,
Twickenham, TW2 5QS, UK
Tel: +44 181 893 3986 Fax: +44 181 744 1149
e-mail: techapp@cix.compulink.co.uk

Printed and bound in Great Britain

to Peter Fowler 1925-1992

Contents

Acknowledgements

It came as both a surprise and a pleasure to discover, when I started working on STEP in 1989, that the standard was being developed by a group of committed, intelligent and above all friendly people, who attacked the problems of developing a massive international standard with verve, enthusiasm, and good humour. I have formed many friendships with others involved in the development of STEP over the years, and it is in "after hours" discussions with many of those friends that the roots of this book lie.

At the risk of inevitable omissions, I would like to acknowledge a number of people within the STEP community for their contributions to the development of the standard and to the subject matter of this book: Yuhwei Yang, Bill Danner, Neal Appel, Mitch Gilbert, Howard Mason, Wolfgang Haas, Mary Mitchell, Brad Smith, Steve Clark, Sharon Kemmerer, Matthew West, Stuart Lord, Brian King, Kevin Freund, Allison Barnard Feeney, Diane Craig, Linas Polikaitis, Jens Erb, Peter Kruse, Bernd Ingenbleek, Hiroyuki Hiraoka, Martin Bringuel, Bob Parks, Tom Johnson, Andreas Burkert, Max Ungerer, Martin van Koetsveld, Mark Palmer, Dave Sanford, Adam Polly, Jim Fulton, Dave Price, Bernd Wenzel, Wim Gielingh, Augusto Nieva, and Barbara Warthen.

Also, present and past colleagues at PDT Solutions, CADDETC, and the Department of Mechanical Engineering at the University of Leeds, with whom I have not only worked on the development, implementation and deployment of the standard, but also shared many hours in aeroplanes and airports travelling to and from STEP meetings: Sheila Lewis, Alan Boyle, Phil Spiby, Jon Owen, Alison McKay, Nigel Shaw, Alison Woodall, and Steve Brett.

Several people have had a direct and vital involvement in the writing and production of this book: David Hitchcock and Sue Day at Technology Appraisals, whose encouragement not only of the development of the book, but also of the "APLS" seminars through which much of the book's technical content has evolved; Sheila Lewis, whose comments on early drafts of several chapters helped to clarify both the structure and content of much of the text; and my mother, Ann Fowler, whose trusty teacher's red pen was used to good effect in spotting and correcting many errors in my usage of the English language.

Last, but definitely not least, my wife Pauline, who not only showed much unrewarded patience during the many evenings and weekends devoted to the writing of this book, but also undertook much of the final proof-reading.

Some sections of the book are based on papers given at various conferences and seminars. In particular:

- Chapters 1, 2 and 3 contain material taken from papers given at seminars organized by the UK Process Industries STEP Consortium (1993), and the Loughborough University of Technology (1994).

- Appendix A is an updated version of a paper given at the "STEP Australia" conference in March 1995.

I have drawn together material and information from many sources: from the STEP standard itself, from conference papers and presentations, and from the results and deliverables of collaborative projects. Whilst I have attempted to acknowledge all direct quotations or borrowings, I am sure that there is much material that I have absorbed by some osmotic process, and reproduced here without reference to its source. To the original authors of such material I offer not only my thanks, but also my apologies.

The manuscript of the book was prepared using Microsoft Word version 5.1 on an Apple Macintosh LC475. The diagrams were prepared using Claris MacDraw Pro version 2.1 and Microsoft PowerPoint version 3.0. Final editing was undertaken using Microsoft Word version 6.0 and Microsoft PowerPoint version 4.0 on a Dell Dimension P90 PC. The camera-ready manuscript was printed on a Hewlett Packard LaserJet 4 Plus laser printer. All products named are trademarks or registered trademarks of their respective companies.

Introduction

The use of information technology within the processes of design, engineering and manufacturing has become widespread in the past 30 years. Consumers are now accustomed to the idea that the domestic appliances they use in their homes, the cars they drive to work or school, and the aeroplanes in which they fly for business or vacation are all designed and built using a variety of complex, advanced computer technologies. This is reinforced in advertising through images of "high-tech" factories, robots, eager young engineers clustered round a glowing computer screen …

Compare this, however, with a contrasting image: many manufacturers emphasize the "human touch", suggesting that while the computer-equipped technocrats may be able to deliver the functional aspects of a product, it requires the intervention of people to inject the "quality" aspects. In *The Machine That Changed The World*, the definitive comparison of the car industries of the USA, Europe and Japan, the need for post-production re-work is identified as a negative aspect of European manufacturing.[1] In the advertisers' world, however, this is seen as a positive aspect in the production of high quality cars, and a link to the "craft" industries of the past.

Although it is dangerous to draw conclusions from an analysis of advertising imagery, this variation in views of the use of information technology in manufacturing may be seen as part of a more fundamental issue. Many companies now freely admit that the benefits of information technology (IT) have not been as great as those promised or envisioned. There are a number of reasons for this dissatisfaction, including:

- unrealistic expectations of the benefits of using information technology within existing, manual processes;

- lack of capability of information technology to respond to changing requirements;

- mismatches between the needs of manufacturing industry and the products offered by the information technology industry.

These high-level problems are often compounded by a fourth, technical factor:

- information technology systems that are incompatible and prevent or hinder sharing of information between those systems.

The subject of this book is the collaborative development of a new information technology standard that responds to dissatisfaction with the level of support provided by the computer industry to manufacturing. This standard – "STEP", the Standard for the Exchange of Product Model Data – focuses on the problems of communicating between diverse computer systems. The development of STEP is being undertaken with the understanding that much of the current generation of computer systems is now acting as a *barrier* to the improvement of manufacturing.

The need for standards such as STEP results from the use of proprietary software systems with "closed" data; ie. the data created using a system from one supplier is not directly accessible or usable by systems from other suppliers. Industry requirements for the exchange of data between dissimilar systems may arise in many different circumstances, but these may, in general, be categorized as follows:

- the use of different information technology systems in different engineering disciplines within a company, eg. in engineering design and finite element analysis;

- the use of different information technology systems by companies or organizations co-operating on a specific project, eg. by an architect and a building services engineer working on the design for a civil engineering project;

- the use of different information technology systems within a supplier chain, eg. an automotive manufacturer wishing to supply data to, or receive data from, its component suppliers;

- the need to deliver data describing a complete product or project to a customer, eg. from a process plant design contractor to the plant owner/operator;

- the need to manage data independently of specific information technology systems, eg. to maintain configuration control of data created or modified using a number of heterogeneous systems;

- the need to manage data throughout the life of the products that the data relates to;

- the need to archive data beyond the active life of specific information technology systems, eg. the operator of a nuclear facility required by government regulation to maintain design information for many years after the decommissioning of the facility.

These scenarios share one characteristic: there is a business need to be able to access data created by one computer system from another, where the two systems may be separated organizationally, geographically, or in time. A further underlying fact is that the data created by computer systems is of significantly higher value to an enterprise than the particular software used to create or access that data, or the hardware by which it is stored or communicated.

The obvious source of information on STEP is the standard itself. However, the standard does not describe the context in which it has been developed, the methods by which it may be used in industry, or the benefits that will arise from its intended use. This book therefore complements the standard by giving a management overview of STEP. It is aimed at:

- **senior executives** wishing to assess the potential benefits of STEP within their businesses, and its incorporation into strategic planning for the more effective use of information technology in the context of business process re-engineering, concurrent engineering, and total quality management;

- **technical managers** wishing to understand the changes that STEP will effect in the acquisition and use of engineering information technology applications;

- **IT managers**, **systems developers** and **systems integrators**, with a need to assess the potential use of STEP in integrating current information technology systems and designing open architectures for future systems.

This book provides an introduction to STEP in terms relevant to the needs of decision makers who need to understand the impact of product data technologies on today's business environment. The intention is also to help readers decide how STEP is relevant to their business needs, and to determine those aspects of STEP that require further detailed investigation.

The book is *not* a detailed technical tutorial on STEP; rather, it provides sufficient information about the standard to enable readers to make informed decisions, and to refer them to sources of more detailed information where necessary.

The book is structured around four major themes:

- **Why is STEP being developed?** Chapters 1 and 2 present the business and technology drivers that have resulted in the development of STEP. Chapter 3 places STEP in the context of more general considerations of data management and data quality.

- **What is STEP?** Chapter 4 describes the history of the development of STEP, from the initiation of the standards effort in 1984 to the publication of the "initial release" of the standard in 1994. Chapters 5 and 6 explain the basic concepts and structures underlying STEP; further information in this area is given in Appendix A. Chapter 7 discusses the relationship between STEP and other standards.

- **What are the benefits?** Chapters 8 and 9 examine the potential business benefits of the effective implementation and use of STEP, considered through case studies drawn from current industry pilot projects.

- **How is STEP used?** Chapters 10, 11 and 12 present the software tools available to developers, implementors, and users of the standard; strategies for adoption of STEP and its adaptation to specific enterprise needs; and a glimpse of the future information technology environment that STEP is helping to build.

Appendices provide additional, supporting information for readers who wish to take "the next STEP" and find out more about the standard and the many activities that support its development.

---

[1] J. P. Womack, D. T. Jones & D. Roos, *The Machine That Changed The World*, pp 90-91

1 Industry requirements

The background to the development of STEP covers many advances in management techniques, manufacturing approaches, and information technology. Although STEP has been developed to solve particular technical problems, as will be discussed in Chapter 2, the real importance of STEP lies in the business drivers that are creating a new environment for manufacturing industries in which the opportunities offered by STEP may be realized.

The business drivers for the development of STEP give the context for the standard. Without this context, any standard is no more than a collection of documents with little real worth. After all, unless businesses see an opportunity for improved financial performance resulting from the adoption of a new standard, it is likely that the standard will not be used at all.

Gaining competitive advantage

Manufacturing industry today faces many challenges. The need to gain competitive advantage is seen as ever more important, emphasising increasing market share, introduction of new products, shorter product life cycles and the need to respond to changing markets.

Companies are no longer able to compete just on the cost or functionality of their products. The 1980s and 1990s have seen an ever increasing emphasis on the importance of quality, and on the ability to respond to customer needs in a timely manner. As business performance in these areas has improved across the board, however, these factors cease to be a discriminator between companies: the use of ISO 9000 quality systems has, for example, moved from being an advantage to a requirement.

In their book *Competing for the Future* Professors Hamel and Prahalad identify the need to focus on competitive advantage on a five to ten year time scale.[1] This long-term approach creates a significant challenge to those who need to understand the relationships between business objectives and the use of information technology. If successful businesses are those that can anticipate their products and markets in ten years' time, can they also predict the kinds of technology that will be needed?

Strategic issues?

Many solutions have been offered to these challenges, and have become part of every management consultant's vocabulary.

- Business process re-engineering: the proactive evolution of company structures through continuous improvement, focus on core business activities and competencies.

- Total Quality Management: business improvement through the adoption of quality approaches to all functions.

- Concurrent Engineering: the reduction of product development time through parallel interaction between all the disciplines involved in the product.

- Design for Manufacturing: the recognition that product designers need to understand and incorporate the ways in which products are made.

Some solutions have even become institutionalized, as in the case of the US Department of Defense's "CALS" (Continuous Acquisition and Life cycle Support) programme.[2] CALS addresses the benefits of ever closer interaction between customers and suppliers through all stages of the product life cycle.

Drivers of change

These solutions all address changes in the way in which business operates. However, a second factor in this environment is the use of information technology and its adaptation to changing business needs. Since the 1950s, information technology has advanced at an ever increasing rate. Every time an apparent plateau is reached in the development of computer systems, another innovation appears to initiate another cycle of rapid improvement.

Many early introductions of information technology applied computers directly to tasks previously carried out manually. For example, initial attempts at office automation replaced the typing pool's typewriters one-for-one with dedicated word processing systems. Similarly, many drawing offices have sought to replace drawing boards with CAD systems, seeking a basic improvement of productivity of an established function.

However, the automation of existing procedures has not delivered the expected benefits, and many organizations have therefore addressed how business change and technology change need to interact. A common goal of companies is the evolution of new business structures, such as the "extended enterprise", with support from information technology to allow integration of processes, often across enterprise boundaries. Business process re-engineering studies often identify the existence of "functional silos": groups or departments linked by undertaking a common function, as opposed to the cross-functional groups that characterize a process-driven approach.[3]

The key lesson that has been learned is that the introduction of information technology should be an enabler for new business processes. Obviously, the coupling of this approach with the philosophy of continuous improvement implies that information technology systems need to be sufficiently flexible to be able to support continuing variation in the needs of their users.

To demonstrate this, it is sufficient to consider only two facts:

- successful US corporations typically undergo major reorganizations at least once every three years;
- information technology systems typically have a life of between five and ten years.

Therefore, a major computer system installed today is expected to support the needs of the business over a period in which that business will be reorganized two or more times.

Integration

"Integration" is one of the key words that arises in discussions of the relationships between business and technology. The word does, however, have a number of different meanings.

- **Enterprise integration** is the identification of new organizational structures and linkages through analysis of the activities carried out within a business, and of the flows of material, information and control between those activities.
- **Application integration** (or systems integration) is the process by which different computer systems are made to work together.

Effective integration in both cases is dependent on the ability to communicate: in the first case between people, in the second case between computers systems.

This gives the initial context for the development of STEP: STEP is a standard for communication between different computer systems that acts as an enabler for both applications integration and enterprise integration.

The role of product information

In considering how information technology needs to adapt to the changing needs of its users, it is important to understand how the technology is used. All manufacturing businesses carry out some process by which raw materials, components, or sub-assemblies are transformed through the processes of manufacturing to deliver a product to a customer. Information about the company's products and processes is clearly of vital importance to the business.

When this information is created and managed in computer systems, we refer to it as *product data*. Generally, this term applies to the data created or used by computer-aided design (CAD), computer-aided engineering (CAE) and computer-aided manufacturing (CAM) systems.[4]

One of the key characteristics of product information is that it is created, used, and added to throughout the life cycle of a product. For example, the basic shape of a building will be created by an architect (design phase), modified by structural engineers (analysis phase), and used in managing the use of the building (operations phase). It is possible in many companies to identify and distinguish between the "manufacturing business", that transforms materials into products, and an "information business" that uses and transforms the information that is needed to support the manufacturing activities.

Before the introduction of computers, this information was communicated using the written and spoken word, together with pictures and models. We may now regard the hand-drawn engineering drawing and the slide-rule as artefacts of a superseded technology, but they were part of a language system that allowed free and easy communication.

The role of information

The role of information is to support decision making. Further, one of the fundamental characteristics of information is that in order to be make such decisions meaningful, then the information must be shareable. Few, if any, organizations are able to undertake tasks, or make decisions, without interaction with other organizations. The information that underlies these tasks and decisions must therefore be shared. It may be thought that some information that is regarded as confidential or sensitive is *not* shared. In fact, confidential information is that where sharing is limited: the usefulness of the information is enhanced by this limited sharing.

The importance of product information (and therefore of product data) has been discussed above. This helps to identify two key linkages. Firstly, that a loss of quality in data can lead directly to a loss of quality in products. For example, if the data that describes the tolerances in a manufacturing process is of poor quality, the manufactured products themselves will suffer from quality problems. The decisions made in developing the manufacturing processes may have been compromised if the information used is inaccurate, incomplete, or ambiguous.

Secondly, the costs of data quality are a direct contributor to the costs of product quality. Using the same example, if problems with the quality of the manufacturing tolerance data incur additional recurring cost, this will increase the costs associated with maintaining the quality of the manufactured product.

Barriers to communication

In an information technology environment, we can identify three types of barrier to communication, as shown in Figure 1 below: barriers between people, barriers between computer systems, and barriers between data. The removal of the barriers to communication between people is a key tenet of concurrent engineering. Many benefits have been claimed as a result of removal of the communications barriers between people working in different disciplines or departments.

However, removal of such organizational or institutional barriers is not likely to be sufficient. Consider the following example. A company has implemented concurrent engineering in the form of multi-disciplinary teams. The product designers, process planners, and manufacturing logisticians all work together and intercommunicate fully. However, each discipline uses different computer systems that cannot talk to each other, and cannot readily make use of each others' data.

This situation is the technological legacy of the "islands of automation" or "islands of information" that were identified during the 1970s and 1980s. These islands result, in part at least, from the piecemeal, unplanned acquisition of computer systems without regard to the requirements of applications integration or data integration. This gives rise to the situation in which "lowest common denominator" solutions have to be identified to enable communications: often, it is only achieved through the use of paper documents.

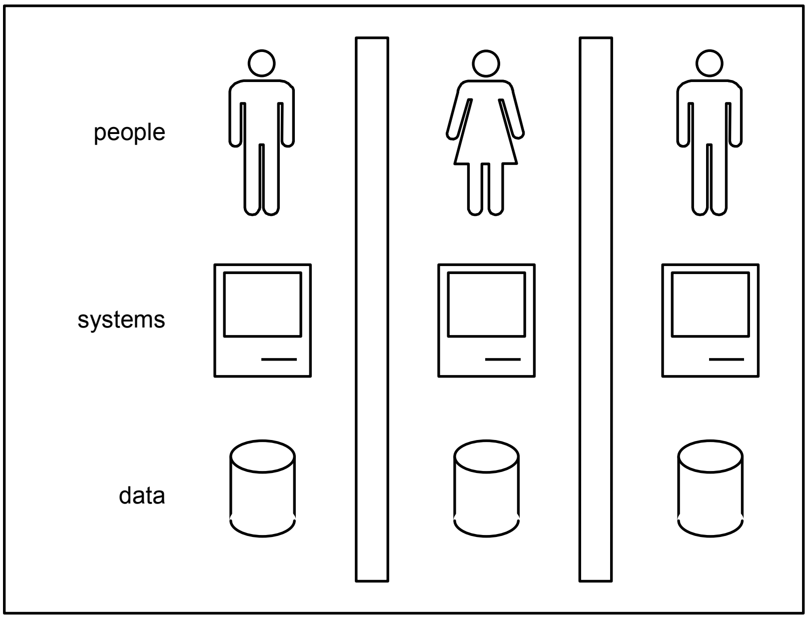

Figure 1: barriers to communication

Lifetimes in information technology

One of the key drivers for data integration is the recognition that data has, in relative terms, a very long life. In fact, data should have the same life as the facts that it represents. Even though the life cycles of manufactured products have been becoming significantly shorter over recent years, increases in customer expectations and of product quality have often led to an increase in the operational life. Today's cars, designed and brought to market in only two or three years, are expected to have a useful life of more than ten years.

Therefore, data related to a manufactured product may have a lifetime of 10-20 years. In some cases this will be much longer: for example, the nuclear and petrochemical industries are required, by law, to maintain design, manufacturing and operations data for plants for 50 or more years *after* the decommissioning of the plant.

In contrast, the applications used to process the data have a lifetime of three to five years and the systems software a lifetime of five to ten years. The hardware elements of computer systems may have lifetimes of less than three years.

Consider a manufacturing company in the year 2015. One of the company's products, nearing the end of its operational life, was designed in the mid 1990s. During this time, the company will probably have:

- replaced its major applications three or four times;
- replaced its systems software at least once;
- replaced its hardware six or seven times.

As previously discussed, the company may also have undergone major organizational changes – including acquisitions, mergers and divestments – as many as ten times in the same period.

It is not difficult to see that access to and maintenance of data will be a problem: even if solvable, the costs of such "legacy data" access are likely to be high unless measures to enable such access are included as part of the product development process.

"Information glut"

Another factor that cannot be neglected is that rate at which the quantity of data has increased with the availability of computers. The advent of low-cost personal computers and workstations has accelerated the creation of ever greater quantities of data.

This increasing volume of data has a number of consequences:

- more "meaningless" data: much data is created but never analysed in a way that makes it useful as information;
- more information: for example, financial decisions previously made on the basis of a simple balance sheet or management account are now taken in the context of voluminous forecasts and "what-if" scenarios;

- the volume of data required to convey a given piece of information is also increasing: compare the size of a typical word processor file with the corresponding ASCII version – is the information content different?

Information costs

It is important to recognize that there are real costs associated with the ownership, management and use of information. Some of these are direct costs:

- the costs of duplicated or redundant effort in the recreation of data: this will be a common factor where paper documents are used to bridge the gaps between "islands of information";

- the costs of maintaining "legacy systems": computer systems whose sole use is to access data created in the past;

- the costs of acquiring and maintaining software that allows data to be exchanged or shared between different applications (translators and interfaces);

- data storage: as data volumes increase, so does the cost of data archiving.

In additional to these direct costs, there are other indirect costs that relate to the less tangible aspects of information management. These costs are often those associated with lost opportunities, such as:

- access to data: time spent looking for data does not contribute to a company's "value-added" activities;

- transcription and translation errors: when data is only accessible through translators or interfaces between systems, there are potential costs associated with the detection and correction of errors;

- loss of quality: data quality problems give rise to product quality costs.

Such costs are often hidden (although accounting practices such as Activity Based Costing may help to identify them). In general, however, the fact that data or information is not seen as a corporate asset can lead to problems. Most companies recognize investments in hardware and software, but not necessarily investment in data.

In fact, the current accounting approach to computer-systems may give an incomplete or invalid measurement of their worth. The treatment of computer hardware as capital equipment, to be depreciated over two, three or five years, may accurately account for the value of a computer in terms of its re-sale. However, the real value of the system to the company is in its use, and particularly in the data created. If this were not the case, why do companies invest in procedures for creating and securing backup copies of data?

Summary

A number of key business drivers have contributed to an environment in which a product data integration standard has become a key requirement. Such a standard is required not only to solve specific technical problems (as will be described in the following chapter), but also as an enabler to fundamental changes in business processes. The key business drivers that have led to the development of STEP are:

- the recognition of the value of information, and the relationship between information and data;
- the need to link data to the products and processes that the data describes, not the computer systems that create or use the data;
- the need to create a flexible information technology infrastructure that supports the goals of business process re-engineering;
- the need to exchange and share information across departmental and enterprise boundaries;
- the linkage between data quality and product quality;
- the need to manage information to the benefit of the business.

Through deployment of a standard that allows key technical information to be freely exchanged and shared, independent of any specific computer system, companies will gain the ability to use information technology more effectively within the processes of product design, engineering, manufacturing and support.

---

[1] Gary Hamel and C. K. Prahalad, *Competing For The Future*, pp 73-77

---

[2] Formerly “Computer-aided Acquisition and Logistic Support”

[3] Michael Hammer and James Chapney, *Re-engineering The Corporation: A Manifesto for Business Revolution*, pp 28-29

[4] The acronym “CAx” is sometimes used as a generic term for the various “computer-aided” systems used in manufacturing industry.

2 Technology drivers

In Chapter 1, the background to the development of STEP has been described in terms of the changing business environments that make an information standard an essential requirement for future competitiveness. However, these were not the key issues that prompted the initiation of the STEP development effort in the mid 1980s. STEP was created at that time to solve a number of specific technology issues; it is only during the development process that the relevance of STEP to "top down" approaches to business improvement has become apparent.

This chapter identifies the technology problems that STEP is designed to solve, and examines some of the alternative solutions to STEP that are available to industry today.

The basic problem

The need for standards that support exchange or sharing of data arises from one basic problem: the fundamental incompatibilities between computer systems. For much of the period that we now regard as the "information age", computer systems have been designed for functionality (what they can do) and performance (how fast they can do it). Most people still think of computers in terms of the operations they carry out (programs, software).

When such computer systems operate in isolation, no problems arise. However, as soon as a requirement exists for two or more computer systems to be used together to solve a specific problem, the need to communicate between the systems – to exchange or share the data upon which they operate – becomes apparent. The problem will be familiar to any user of word processors or spreadsheets. Until recently, such software offered very limited capabilities to exchange or share data that, from the users' perspective, should *by definition* be shareable. For example, it *should* be the case that a table of figures contained in a document is the table created in a spreadsheet, not a copy of it extracted as a simple unstructured ASCII file.

While these issues are being addressed in the office automation market through the development of innovative operating systems and applications software designed for data sharing, these facilities are open to very few users in the domains of design, engineering, or manufacturing. With a few exceptions, most

CAD/CAM systems on the market today are direct descendants of packages developed in the 1960s and 1970s, a time when software design and development was concentrated on an algorithmic approach. This approach may lead to an under-emphasis on the importance of data and of data structures, and thereby to problems with the exchangeability of data.

Bridging the gaps

Given that the computer systems used in manufacturing industry today cannot read, write and share each others' data, what options are available to users to bridge the gaps between their computer systems? Many different solutions are available; these, however, fall into five categories:

- manual re-input of data;
- adoption or imposition of "standard" systems;
- direct translation;
- neutral format translation;
- shared product databases.

Each approach has pros and cons; however, the last two options show the greatest promise for future flexibility and adaptability, and are therefore the basis for the data exchange and sharing environment to be supported by STEP. The way in which STEP provides this support will be examined in later chapters. Here, however, some of the details of these five options are examined.

Manual re-input of data

When no obvious, cost effective means of exchanging data digitally is available, the only remaining solution is to take the output of one computer system and use it as the basis for re-creating the same thing in a second system. For example, a document produced on a word processor is printed and then mailed or faxed to a collaborator, who re-enters the document into a second word processor. In many circumstances this may be the most effective means of exchanging data: for example, there are no costs of translator software, and little or no needs for set-up procedures.

However, there are costs associated with the time taken to re-enter the data, and other, often hidden, costs associated with storing and maintaining redundant data and with finding and correcting errors and inaccuracies. Therefore manual re-input should only be seen as an alternative for “one-off” requirements. Most organizations will accept the additional costs of re-typing a short document; few, however, would countenance employing skilled CAD operators to recreate thousands of complex engineering drawings produced by another CAD system.

System standardization

If the problem of data exchange arises from the incompatibilities between different systems, then a possible solution is to avoid the problem in the first place by adopting a single, common source of computer systems. Elements of this approach are found in all organizations: few users will be given the choice of buying *any* appropriate system within budget.

System standardization within a company makes very good sense: it reduces diversity, cuts down costs of training and maintenance, and can often lead to significant cost savings in systems acquisition. For major industry users, the adoption of a single system can lead to a close and advantageous relationship with the developer or vendor of that system.

Such standardization of systems can also be attempted across company barriers. It is common practice in Japan that component suppliers will use the same CAD/CAM systems as their customer, and may in many cases be supplied with that system as part the business relationship. Similarly, when a number of organizations come together to work on a specific long-term project they may agree at the start of the project to avoid data exchange problems by adopting a single, common system.

Many attempts have been made in the past to adopt standard systems across supplier chains, either by consensus, coercion, or contract. In most cases, pressure will be exerted by the major customers at the top of the chain (sometimes in partnership with their preferred information technology supplier) on to their suppliers and sub-contractors lower down the chain. Such imposition can be effective when the relationships between customers and suppliers are stable and long term.

However, few supply chains are simple. Consider the plight of a medium sized component supplier whose business depends on the ability to service the needs of several major customers. Customer ‘A’ decides to standardize on CAD system ‘X’, and to apply this standard to its customer chain. Customer ‘B’ makes a similar decision, but chooses system ‘Y’. Our component supplier is now being told by two of its major customers that the use of system ‘X’ *and* the use of system ‘Y’ is essential to maintaining the business relationships. The supplier is faced with three alternatives:

- choose one of the systems as its own standard, and risk losing business with the customer requiring the second system;

- acquire both systems, train operators in their use, and accept the additional overhead costs of doing business with ‘A’ and ‘B’ using their preferred systems;

- negotiate with ‘A’ and ‘B’ to develop a more rational technical solution to the problem.

A variation on standardization of systems is to standardize the “native” data format of a specific, proprietary information technology system. This has the effect of moving the requirement to transform or translate data away from the interface between companies or organizations. In this case, a supplier is not required to *use* the preferred CAD system of its customer; it does, however, have to maintain the capability to deliver and to receive data *in the format* of the system in use within the customer. The net effect of this approach is to make the supplier solely responsible for data translation.

Direct translation

If manual re-input of data is not cost effective, and system standardization generally impractical, solutions must be developed that allow one computer system to read and write data in the format required by others. Since the data formats used by the computer systems are different, a translation process is involved. This process is analogous to that between human languages. For example the phrases “the cat” and “le chat” mean the same thing in the English and French languages; school children in England and in France are taught to translate “the cat” into “le chat” and vice versa.

A similar process is required to translate data between computer systems. For example, one CAD system represents a straight line on a drawing by storing the co-ordinates of the start of point of the line and the co-ordinates of the end point of the line. A second system represents the same line by the co-ordinates of its start point, its direction, and its length. To exchange data about straight lines we have to translate from the “two points” representation of a line to the “point plus direction plus length” form.

Software packages that accomplish this translation for a given pair of systems are known as “direct translators”. Typically, two translators are required to achieve bi-directional exchange of data between two systems, as shown in Figure 2.

Such direct translators can achieve very high quality results, but suffer the disadvantages that:

- they are expensive to acquire and maintain, as they have to be updated each time *either* CAD system is upgraded;
- they are limited to “point solutions”, ie. to a specific pairing of systems;
- the number of such translators increases exponentially with the number of systems involved: for **n** systems, the number of direct translators required is $\mathbf{n} \times (\mathbf{n} - 1)$.

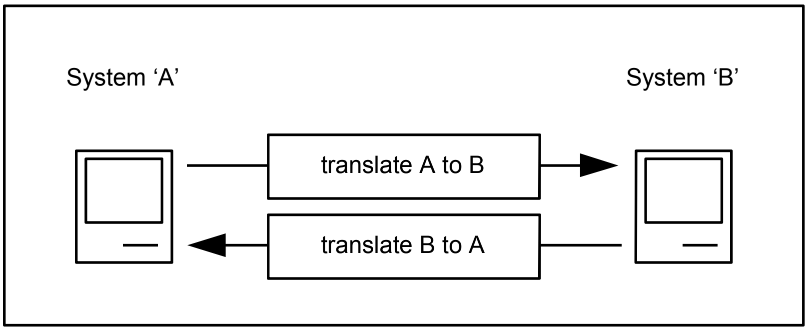

Figure 2: data exchange using direct translation

Nonetheless, direct translators today offer very high quality solutions to many of the common data exchange scenarios encountered in industry. Indeed, for pairings of market-leading CAD systems, users may be able to choose between direct translator products from several technology providers.

Neutral format data exchange standards

An alternative to the development and maintenance of direct translator software for each pair of systems is to develop and agree specifications for data exchange that can be supported by all systems. This approach, often referred to as the use of “neutral formats”, can be thought of an “Esperanto” for CAD systems, ie. a universally accepted and understood language for data.

The need for data exchange standards was originally recognized in the late 1970s and led to the development of specifications such as IGES (the Initial Graphics Exchange Specification) in the USA, SET (Standard D'Echange et de Transfert) in France, and VDA-FS (Verband der Automobilindustrie-Flächen-Schnittstelle) in Germany. These standards all prescribe the use of standard file formats for the exchange of data, and therefore require the use of two translators: one to translate from the internal data format of the application to the standard ("pre-processor"), and second for the reverse translation ("post-processor"). Pre- and post-processor software packages may also be referred to as "half-links".

The elements of data communication based on the use of such standards is shown in Figure 3 below.

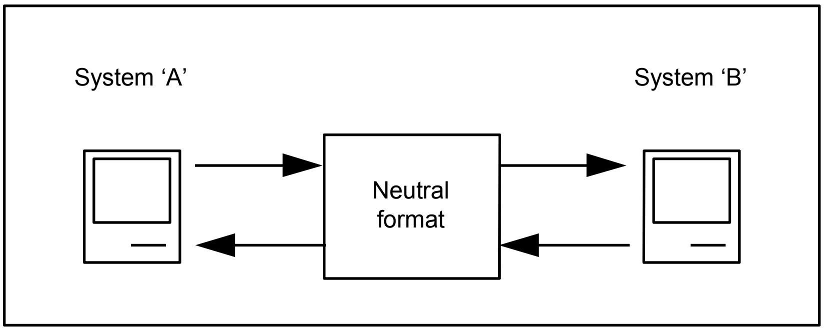

Figure 3: neutral format data exchange

While the use of such standards represent "best current practice" in many industries, all require considerable effort to achieve effective results: none are "black-box" technologies. The following factors contribute to the perceived deficiencies of standards such as IGES:

- the specification is open to ambiguous interpretation, and there are therefore variations in the quality of translator software;
- every CAD system vendor supports a different subset of the standard applicable to its own products;
- the standards are limited to the exchange of geometric information (the two- or three-dimensional shapes of objects), engineering drawings, and some non-graphical data such as connectivity.

Further details of the various neutral format specifications available are given later in this chapter.

Linking applications through OLE

As more CAD/CAM applications are developed for the Microsoft Windows operating system, then the opportunities offered by Microsoft's OLE method for linking and embedding data will become significant as a mechanism for exchanging and sharing data between systems. OLE extensions for modelling and design allow 3D geometric and graphical objects to be shared between appropriately designed applications.

These OLE extensions do not, however, support the complete functionality of CAD/CAM systems, and offer little opportunity for exchange or sharing with applications that use other operating systems. The primary benefit of OLE is to integrate applications packages from different vendors, allowing a user to create a set of interoperable tools as a single system. OLE does not, however, support exchange of data across organization and system boundaries.

Direct translators and neutral formats compared

Table 1 below summarises the pros and cons of direct translators and neutral formats.

Direct translators	Neutral formats
Software designed for specific translation need	Combine two "half-links" from potentially different suppliers to achieve translation
Includes necessary conversions of data as well as translation	May require "flavouring" of data to achieve best results
Expensive to maintain: have to be updated every time one system changes	Published formats are stable
Require **n** × (**n** - 1) translators to communicate between **n** systems	Require 2 × **n** half-links to communicate between **n** systems

Table 1: direct translators vs. neutral formats

Some of the limitations identified above have been addressed through developing direct-translation systems based on a "hub-and-spoke" architecture, as shown in Figure 4. These systems represent a combination of the advantages of both neutral format standards (reduced number of interfaces) and direct translators (high quality interfaces).

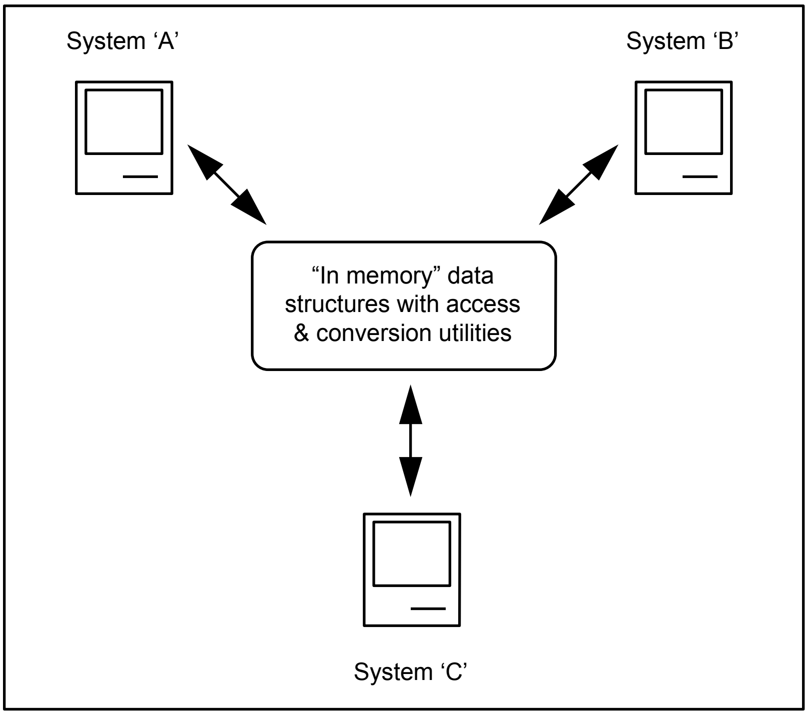

Figure 4: direct translation using "hub-and-spoke" architecture

Standards for CAD/CAM data exchange

The advantages of neutral format exchange, as described above, can only be realized if the specification of the neutral format is widely available. Such availability is commonly achieved through the agreement and publication of standards. This section identifies and discusses the various standards and other widely-available specifications for CAx data exchange that are available today.

The following statements, paraphrasing guidelines developed by the British Standards Institution, summarize the benefits of using standards.[1]

Good standards …

- promote consistent quality and economic production;
- simplify manufacture and encourage interchangeability;
- rationalize processes and methods of operation;
- make the exchange of goods and services easier;
- give confidence to users.

Since the earliest development and use of standards in industry during the 19th century, these benefits have been achieved. It should be noted, however, that it is only *good* standards that fulfil these criteria: if a standard does not deliver these benefits, then it is nothing more than a collection of paper.

Standards are achieved through several routes, although in every case the criterion of effective use in industry has to be employed. The most "visible" form of standardization is that undertaken by national and international standards bodies such as ISO. Here, standards are developed according to well defined processes for achieving consensus; national or international standards also have the potential benefit of being accepted contractual documents. However, the process of standardization can be lengthy, and is best adapted to the codification of established, existing industry practices. In a rapidly changing environment, such as information technology, this can lead to the situation of standards being developed that reflect outdated or superseded technology or practices.

In response to this, many industry groupings have been formed to define and set standards in a more timely manner. Again, standards are driven through the processes of achieving consensus, but greater responsiveness can be achieved by focusing the development within industry and bypassing the formal procedures of international standards approval and publication. Many such groups exist, including the Object Management Group (OMG), which defines standards for "object-oriented" programming and database systems, and the Petrotechnical Open Software Consortium (POSC), which defines standards for technical computing in the oil and gas industries.[2] Such standards may lack the authority of national or international standards, and there is a greater likelihood of overlap or even conflict between different standards development activities.

The final, and probably most common, route for the establishment of standards is through market acceptance of products and specifications. For example, Microsoft Windows is *the* standard for operating systems and graphical user interfaces for personal computers. Such "de facto" standards by definition represent best current practice in industry. They do, however, not only share the disadvantages of industry-developed standards, but also may not fulfil criteria for universal availability.

Current data exchange standards

The following sections summarize the development, use, and current status of the leading standards for CAx data exchange in use today.

IGES

IGES, the Initial Graphics Exchange Specification, was developed in the USA in the late 1970s. It is now established as the most widely used format for CAx data exchange. IGES is developed by the IGES/PDES Organization (IPO), a voluntary standards development body, and is published as a US national standard (ANSI Y14.26M). IGES has been developed and extended continuously since its initial release; several different versions have been published.[3] Most major CAD/CAM systems support IGES, although translators may be based on different versions of the specification and may (as discussed in greater detail below) implement different subsets of the standard.

IGES has been adopted in many major companies and projects, as well as being used within public sector procurement programmes. Subsets of IGES have been published as US military standards (as part of the CALS programme), and a Federal Information Processing Standard (FIPS) has been developed. Conformance testing services for IGES translators claiming support for both the CALS subsets and the FIPS are available from the National Institute for Standards and Technology in the USA, and from CADDETC in the UK.

Early versions of IGES, and translators based upon them, suffered from many teething problems and as a result acquired a poor reputation. This was worsened by the problems associated with the large sizes of IGES files. In response to these problems with IGES, a number of alternative specifications were developed in the mid 1980s.

SET

SET (the Standard D'Echange et de Transfert) was developed in France, and first published as a national standard in 1985. The development of SET was driven by major manufacturing companies in the automotive and aerospace industries, and was designed to address the issues arising from difficulties in using IGES. SET has proved successful within French industry, and projects in which French companies participate (such as the Airbus Industrie consortium). Association GOSET, an organization established by industry and government in France to support the development and maintenance of SET, is now an active contributor to the development of STEP, and works to ensure that the benefits of SET over IGES are maintained and enhanced in the new standard.

VDA-FS

At the same time as industry in France was initiating the development of SET, the car industry in Germany identified a requirement for a standard to support the effective exchange of surface models, such as those used in the styling and design of car bodies. VDA-FS (Verband der Automobilindustrie-Flächen-Schnittstelle) was developed by the automotive industry trade association (VDA) in Germany in the 1980s. VDA-FS is specifically designed for exchange of surface models, and has achieved considerable success in the automotive industry.

The VDA has been responsible for the development of other standards, such as the VDA-IS subsets of IGES, and VDA-PS, a standard for component libraries that has formed one of the bases for the development of the ISO Parts Libraries standard, ISO 13584.[4] As with SET, the developers of VDA-FS are now actively involved in STEP, and are defining the requirements for migration from VDA-FS to STEP.

EDIF

Although IGES has provided some capabilities for the exchange of data describing electrical or electronic products, this industry sector has developed a number of standards specific to its own needs. The most prominent of these is the Electronic Design Interchange Format (EDIF), developed by the Electronic Industries Association (EIA), for the exchange of integrated circuit and printed circuit board designs. Other standards in use in the electrical and electronic industries include VHDL (an IEEE standard for exchange of designs between simulation tools), IPC (for packaging), and VNS (for cabling and wiring assemblies). Work on these standards is being integrated with that on STEP through liaison between IEC and ISO.

DXF

DXF (Data Exchange File or Format) is a specification developed by Autodesk, Inc. to support links with their “AutoCAD” software. Unlike IGES or SET, DXF is not a published or approved standard. The DXF specification is, however, published by Autodesk and is therefore widely available, and has been implemented in many CAD systems, particularly those that operate on personal computers. DXF is a simple and relatively limited format: possibly for this reason, it has become well established as a means for exchanging engineering drawings. In 1991 DXF was recommended by a working party of the UK National Economic Development Council (NEDC), as the preferred format for drawing exchange in the building and construction industries. Its use in this industry sector remains wide-spread.

Who uses which standard?

Table 2 below summarises the use of the various standards outlined above by different industry sectors. A double-tick (✓✓) indicates the standard that has the widest use in each sector.

	IGES	**SET**	**VDA-FS**	**EDIF**	**POSC**	**DXF**
Aerospace	✓✓	✓				✓
Automotive	✓✓	✓	✓			✓
Building and Construction						✓✓
Process plant	✓	✓				✓
Oil and gas					✓✓	✓
Shipbuilding	✓					✓
Electrical/ electronic	✓	✓		✓		
Consumer goods	✓					✓

Table 2: comparison of the use of exchange standards

The near universal use of DXF that is apparent from this table needs to be understood in the context of the capabilities of the specification: it is common to find that DXF is used to exchange drawings where a specification with higher functionality, such as IGES or SET, can not be used.

Data exchange in practice

The requirements for data sharing, the final option identified above for communications between systems, are discussed in the next chapter. The remainder of this chapter deals with the details of data exchange technologies and examines some of the issues that arise from their use. Later chapters identify the way in which the STEP standard, and its associated technologies, provide solutions to such problems.

The data exchange process

Solutions based on the use of direct translation or neutral formats each require a sequence of operations to be carried out to effect the exchange of data from one computer system to another. In order to understand the issues that arise in data exchange, it is useful to examine this process in some detail.

Figure 5 illustrates this process, which may involve as many as seven separate operations. These operations may be combined or hidden within translator software. A direct translation process will usually involve fewer operations, as is discussed below.

The seven possible operations within the data exchange process are as follows.

1. **Translation**: conversion from the internal data structures of the sending system into a format suitable for exchange. In the case of a neutral format exchange, this will be the neutral format. For a direct translator, this will either be the internal format of the receiving system, or some suitable intermediate form.

2. **Flavouring**: it is often useful or necessary to manipulate or “flavour” the data to be exchanged, so that it is more readily useful to the receiving system. Flavouring may be specific to the “target” system (eg. specifying the accuracy and tolerances of numerical data), or may be specific to the *use* of the target system (eg. mapping the sender’s convention for the use of layers in a drawing to that of the receiver).

3. **Encoding**: the data is encoded in a form suitable for exchange. Typically, data is encoded as ASCII files in order to overcome differences between computing environments; encoding may, however, also include file compression and data encryption.

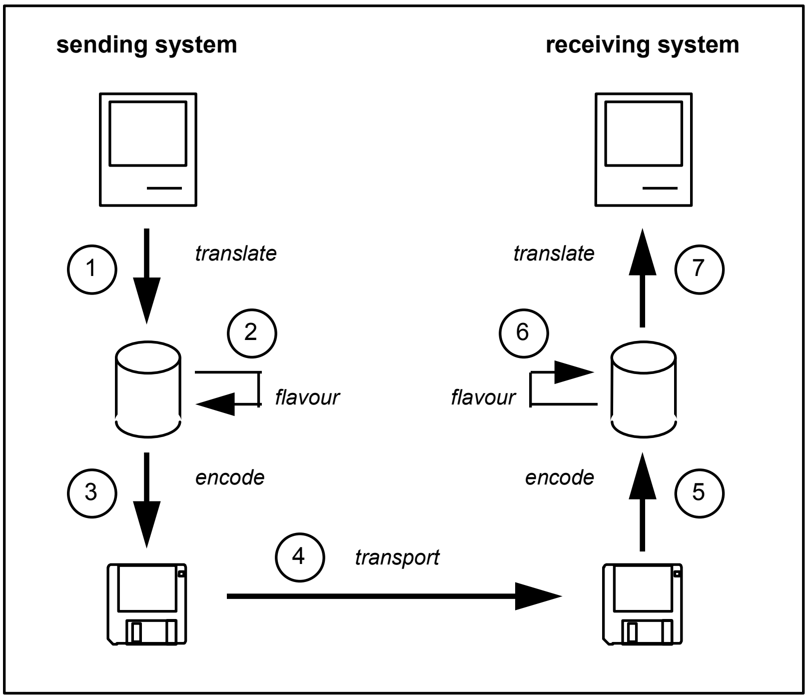

Figure 5: the data exchange process

4. **Transport**: in most cases, the sending and receiving systems will be physically separate, and so the encoded data has to be transported between locations. This may be accomplished by physical transfer (sending a disk or tape from one site to another), or by electronic transfer.

5. **Decoding**: the received file is extracted from the exchange medium and made available in a form suitable for further processing.

6. **Flavouring**: as in stage (2), the exchanged data may be manipulated in order to improve the quality of the translation.

7. **Translation**: the exchanged data is translated into the internal format of the receiving system.

At each stage in this process there may be secondary tasks, including data validation and authorization. Not every case of data exchange will make use of all of these stages. In a direct translation process, there is likely to be only one translation process, which will normally include the "flavouring" function. The translation will happen before or after the transport operation: ie. the data that is "physically" exchanged between the systems will be in the native form of one or other system. In the most simple direct translation, where the sending and receiving systems are networked together, the only process visible to the user may be the first translation stage: a single command in the sending system results in the creation of a file in the format of the receiving system. In complex neutral format exchanges all seven stages may be identifiable.

Data exchange issues

The break-down given above allows analysis of the issues and problems that are commonly found in data exchange. Although some of these issues relate to the technology used for data exchange, many result from the management (or lack of it) applied to the use of the sending and receiving systems, and of the data exchange process itself. The identification of these issues supports the creation and maintenance of a checklist for users of data exchange.[5]

Data organization

The first issue is that of data organization. This includes naming conventions for files, elements of CAD models and drawings, as well as the use of colours, layers, and symbols. It is important that the parties to an exchange identify and agree in advance the conventions that are to be used. For example, if company 'A' has a convention that red lines indicate hot water pipes, while company 'B' uses red lines for pipes carrying hazardous waste, potential problems may arise if such a difference in conventions is not identified before any data is exchanged.[6]

There are typically two approaches to dealing with this issue. If the companies involved are embarking on a project that will involve many exchanges of data, it is recommended that they agree on a single set of conventions for use in that project, ie. the intention is that engineers in companies 'A' and 'B' will use red lines for hot water supplies, not for hazardous waste. The alternative is to identify and compensate for different conventions as part of the exchange process. Here, the exchanged data is manipulated so that the red lines created by an engineer in company 'A' to represent hot water pipes are automatically converted to the convention used for hot water pipes in company 'B'.

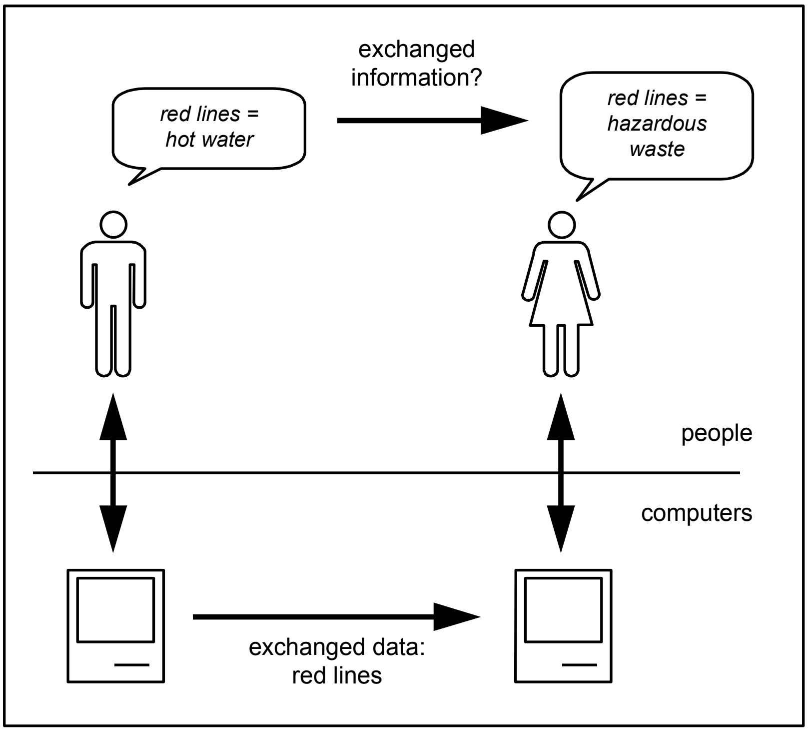

Figure 6: exchange of data or information?

This is an important point that will be revisited several times in later chapters. The issue arises from the fact that the engineer has in his or her mind the idea of "hot water pipes". In this simple example, the exchange between the CAD systems is of coloured lines: the meaning "hot water pipe" is added in the engineer's mind. There is clearly considerable potential information loss in this exchange.

Choice of exchange method

The second issue to be addressed is the choice of method for the exchange. As already discussed, there is a broad choice between the use of direct translators or of neutral formats. The choice between the two paradigms will be driven by a combination of three factors:

- contractual requirements;
- availability of software;

- business relationships.

For many organizations, the first of these will be the overwhelming factor: if a customer requires delivery of data in a certain format, then the appropriate technical solution will have to be found. It is noteworthy, however, that increased requirements for data exchange have led to an improvement in both the availability and the quality of data exchange software tools. Up to the early 1990s, a user would be fortunate to find *one* appropriate solution to a data exchange or delivery requirement; today, however, a user can often choose one of several solutions.

Nonetheless, the issue of software availability may be a significant constraint. Although the CAD/CAM market is now dominated by a relatively small number of software vendors, there are still large numbers of specialist, “vertical market” suppliers. If the requirement is to exchange drawings or 3D geometric models between two market leading CAD systems, there may be a choice between several successful, proven solutions. If, however, a user’s need is to extract such data from a CAD system for use in a specialist analysis package, the number of choices will rapidly diminish.

The third factor, that of business partnerships, will come into play when the relationships between organizations are established, or are expected to be long-term. Even with the future advantages offered by STEP, data exchange is not a “black box” technology and may require considerable costs in testing and benchmarking translators, and developing procedures for data organization and exchange. Clearly, if the need for exchange is short-term, then these set-up costs may be a significant (if not overwhelming) proportion of the total costs of a single exchange.

If, however, two or more organizations anticipate working together and exchanging data on a frequent basis over a period of months or even years, then the set-up costs can be apportioned over a large number of individual data exchange transactions. Such partnerships are becoming more common, and are no longer just the province of major aerospace or automotive groupings. Indeed, the realization of the cost savings and benefits in sharing the “up-front” costs of data exchange technology may itself be seen as a driver in the development of STEP.

Translator software

Even if one or more translators are available to solve a specific need, there are other issues to be addressed before “painless” data exchange can be predicted or achieved. The first of these are those of the scope, functionality, and quality of the translator software. Even with high quality direct translators or neutral format interfaces, there is a strong probability that the translator will support less than 100% of the data that may exist within the internal data of the CAD/CAM system. When two translators are combined (as will be the case for neutral formats), then there is likely to be a further mismatch. Exchange between two translators that each support 90% of the data within the respective CAD/CAM systems can result in up to one fifth of the available information being lost in an exchange.

A second, related issue, that applies specifically to neutral formats, is that of the subset of the total specification chosen to be implemented in a translator. The data exchange standards and specifications established in the 1970s and 1980s contain a large collection of data structures (entities and attributes), often with options and variations, based on the diverse and increasing requirements of contemporary CAD/CAM systems. A programmer, faced with the task of developing an interface based on one of these standards, will then pick and choose from the specification the subset that matches the CAD/CAM system for which the interface is written.

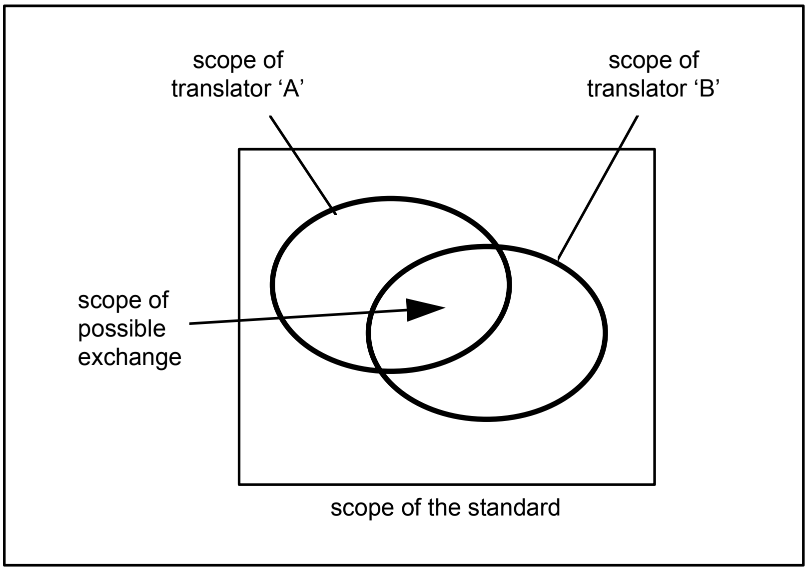

Figure 7: mis-matches between subsets of a standard

Interface developers for every other system will do the same thing, each basing a translator on a different subset of the standard. This clearly leads to problems with interoperability between systems, resulting from mis-matches between these subsets. This issue is illustrated in Figure 7 above; this shows the case for two systems, from which it will be appreciated that significant loss can occur if data is exchanged between successive pairs of systems, as may occur in a supplier chain. The loss of information that can occur here closely resembles the classroom game of "Chinese Whispers".

The final issue that may apply to translator software is that of quality. While there have undoubtedly been problems in the past with poor quality translator software, the increased perceived importance of data exchange has brought a consequent increase in the quality of the software.

However, there are issues that relate more directly to the quality of the *specifications* on which the translator software is based, ie. the standards themselves. When such a standard is specified in a "natural" language such as English, the specification is likely to be ambiguous, or confusing, or both. The chances of two programmers reading part of the specification, taking the same meaning from the English words, and implementing the same functionality, is certainly less than 100%!

From subsets to application protocols

As discussed above, one of the persistent issues with the use of current standards such as IGES is the mis-match between the subsets of the standard that each system vendor selects as the basis for implementing a translator.

One possible solution to this is to specify subsets of the standard, and to use these subsets as the basis for implementation. This approach has been used with IGES: both the German automotive industry and the US Department of Defense have defined subsets of IGES to meet their particular needs.[7] Although this has proved of some use in improving the quality of data exchange, problems are still encountered if two system developers use the same entity in the standard to mean different things, or use different entities for equivalent meaning. In order to address this issue, the concept of the *Application Protocol* was introduced into IGES in the late 1980s.

The Application Protocol concept is a refinement of the development of subsets, which focuses on the clear definition of the end-user or application requirements to be satisfied. A subset of the standard is chosen that meets these requirements. The "added value" of an Application Protocol is that it not only specifies a subset as a basis for implementation and testing, but that it also relates the subset explicitly to a set of industry requirements and thereby specifies the precise meaning of each entity.

Exchange media

The final issue to be taken into account in maintaining effective data exchange practices is that of the choice and management of exchange media. Again, there is a series of choices to be made. The first is between physical media (disk or tape) or direct communications. Physical media are by far the more common method for routine data exchange, being reliable, cheap, and readily available. There are nonetheless a number of possible pitfalls in using such media for data exchange; they may, however, be easily avoided through simple management procedures.

The most common problems associated with the management and use of exchange media are those of labelling: no amount of external study of an unlabelled magnetic tape or floppy disk can tell its recipient what the contents are! Given that there may be several different formats for the same "physical" medium, accurate identification can help to avoid potentially disastrous errors.

Configuration management

One of the key issues associated with exchange of data is that of configuration management. In Figure 5 above, the result of the exchange process will be the creation of additional data. Before the exchange, data exists in the sending system; as part of the exchange process, at least one exchange file will be created; after the exchange, data exists in the receiving system. This clearly creates a potential problem: which of these is now the "master" form of the data?

This problem becomes more significant when data exchange forms part of two-way communication between organizations, and is illustrated by the following scenario. A car manufacturer is developing the design of a new model. The styling department, responsible for the external shape of the car, is concentrating on the details of the front of the car body, including the shape of the headlights. As part of a partnership approach to product design and development, the car manufacturer works closely with a preferred supplier of headlights, to the extent that the headlight supplier is involved in the design process.

Now, suppose that the design department of the car manufacturer sends a three-dimensional CAD model data to the headlight supplier, requesting a feasibility and costing study for delivery of headlight assemblies. A neutral format exchange is used, so after the exchange the CAD model exists as three separate sets of data:

- the data in the car manufacturer's CAD system;

- the exchange file sent from the car manufacturer to the headlight supplier;
- the data in the headlight supplier's CAD system.

The headlight supplier analyses the proposed design, and, in order to increase the use of existing parts, suggests a number of changes in the shape of the headlight assembly and mounting. These changes are communicated to the car manufacturer, accompanied by a CAD model that represents the changed shape. As before, this data is communicated by neutral format file; now, the number of sets of data has increased further, as follows:

- the data in the car manufacturer's CAD system;
- the exchange file sent from the car manufacturer to the headlight supplier;
- the data in the headlight supplier's CAD system;
- the modified data in the headlight supplier's CAD system;
- the exchange file sent from the headlight supplier to the car manufacturer;
- the modified data in the car manufacturer's CAD system.

We cannot assume, however, that the car manufacturer's designers have been idle during the analysis and redesign work by the headlight supplier. Therefore, a number of changes will have been incorporated into the design of the front section of the car, and these will have been incorporated into the models stored within the manufacturer's CAD system (ie. a seventh set of data). How does the design engineer within the car manufacturer respond to the proposals from the headlight supplier?

In such a scenario, of course, all manufacturing companies will have well established procedures for change management or approval and release of designs. The purpose of exploring this scenario is to demonstrate that such procedures are further stretched by the use of CAD systems and of data exchange.

One response to the problems associated with this situation is the adoption of product data management disciplines, and the implementation of systems that support these disciplines. A second, more radical response is to concentrate on the benefits of data sharing, rather than data exchange, and thereby to reduce the impact of configuration management issues inherent in the process of exchange. These two responses are examined in further detail in the next chapter.

Conclusions

The existence of incompatibilities between information technology systems used in the processes of design, engineering, and manufacturing has resulted in the development of several different approaches to achieving the exchange of data. Past attempts to develop neutral format standards for data exchange have had limited success, and have led to the development of STEP as a replacement data exchange standard that builds on the “lessons learned” of the current standards.

Nonetheless, concentration on the quality of both the standards and the software based upon them has delivered many effective data exchange capabilities; however, the inability of current standards to support industry needs for effective data sharing as well as exchange has been a further technical driver on the development of STEP.

---

[1] British Standard BS0: Part 1: 1981 “General principles of standardization”

[2] The POSC standards, and their relationship to STEP, are discussed in Chapter 7

[3] The most recent version is 5.2, published in 1994

[4] ISO 13584 is described in Chapter 7

[5] Guidelines for the creation and use of such a checklist are given in *The Exchange Agreement: Guidelines for the Successful Exchange of CAD-CAM Data*

[6] Standards for the use and meaning of colours, linestyles, layers, etc. in CAD drawings are being developed by ISO TC10; these extend the existing requirements of national and international standards for engineering drawings.

[7] The German automotive industry subsets of IGES are published as VDA-IS; the US Department of Defense subsets are published as MIL-D-28000A (one of the “CALS” standards for electronic data interchange).

3 Data management

The discussion at the end of Chapter 2 regarding the configuration management issues raised by the use of CAx data exchange is just one illustration of a wider set of problems. The widespread adoption of information technology within all sectors and disciplines of industry over the past 20-30 years has created vast quantities of data. This data frequently represents vital information to the successful operation of an enterprise, and yet may not be readily available to those who need it.

This has led to a realization that the *management* of data is as critical to a business as the management of people, resources, or money. Data management encompasses the activities associated with administering and controlling how data is used within an organization, together with planning and implementing the processes and systems that are used to undertake these tasks.[1]

This chapter examines three key aspects of data management:

- the requirements for data management;
- the processes and systems used to manage product data;
- the roles of data models and data modelling, and their relevance to STEP.

The need for data management

Data management occurs at many levels. Maintaining control of data requires that it is organized in a systematic way. All computer systems now provide basic capabilities for such organization, allowing users to store data in files, to arrange the files in directories and sub-directories. People will tend to use these facilities differently: for example, it would not be obvious to another person where to locate the word processor file for this chapter on my computer.[2] For a genuinely single-user system, this is not a problem – as long as the user remembers what the directory structure is! However, as soon as computer systems are shared between people, it becomes imperative to make use of data management approaches that are consistent.

Any data management scheme has to focus on the *retrieval* of information. Storage of information is easy; finding what you need within a collection of stored information can be much more difficult.

The symptoms of data management problems are not always easy to detect. In some cases, the lack of adequate management will be readily apparent:

- engineers spend considerable time searching for data related to a product or process;
- people create and maintain "local" databases that duplicate shared facilities;
- managers have to reconcile data pertaining to the same project coming from several different sources.

However, since data management issues are frequently associated with factors such as incompleteness, ambiguity, or version control, the resulting problems can be much more difficult to detect and therefore to correct. The consequences of *undetected* problems are often manifested at a later point in time: the problem may have been magnified through series of decisions made on the basis of incomplete, inaccurate, or ambiguous data.

The following factors relate to data management requirements:

- availability of data to all who need to be able to use it;
- the independence of data from the applications that create or modify it;
- the flows of information within an organization, and between an organization and its trading partners;
- the need to be able to control and manage legacy systems and legacy data.

Availability of data

There are several factors that influence the availability of data.

- **Controlled access**: access to data is critical to those making decisions; however, such access cannot be universal without introducing issues of security, and confidentiality.
- **Timely access**: data is useful only if it is available when it is needed.

- **Version management**: data, particularly the highly complex and structured data associated with engineering processes, frequently exists in multiple versions. Different versions (and combinations of versions) must be identified and distinguished.

- **Data redundancy**: many sets of data relating to the same product or process may be created. When this occurs, the inter-relationships between data must be identified. This identification must support the concept of “create once, use many times”, whilst taking into account the performance benefits that may arise from controlled redundancy.

- **Location of data**: as computer systems have become more complex and powerful, data becomes more widely distributed. It is therefore a function of data management to identify where data is stored, whether this is within the directory structure of a single system or across multiple, distributed networks.

- **Format of data**: even if all the criteria described above are satisfied the usefulness of data will be severely restricted if it is not available in a format that enables its further processing. The data management function therefore encompasses the translation of data between different formats.

Independence from applications

As discussed in detail in Chapter 2, many current computer applications have proprietary formats for data. These formats are often “closed” in that the data is *only* accessible through the application that created it. Given that typical industry processes require many such applications, data translation is required to exchange data between them. Figure 8 illustrates the exchange of data between two such applications.

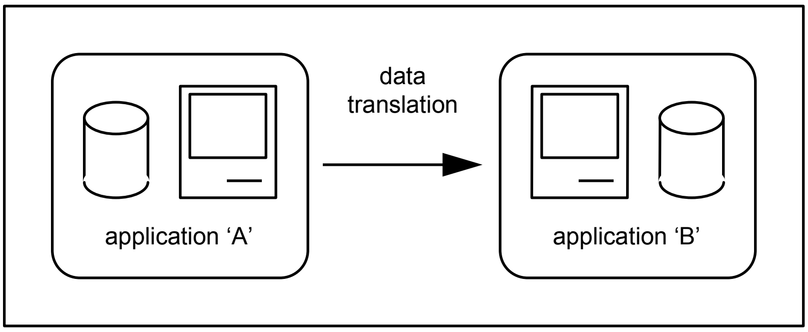

Figure 8: exchange of data between systems with proprietary data formats

Note that after the exchange:

- the quantity of data has increased; in addition to the data stored in the database of system "A", there is now data in the database of system "B", and there may also be a file (eg. an IGES or DXF file) that has been used to move the data from one system to another. This creates additional data management requirements (eg. is the data in system "A" or system "B" the master?).
- the data in system "B" may not be the direct equivalent of that in system "A", ie. some information may have been lost in the exchange.

The recognition of these issues has led to the identification of a long-term requirement for the separation of data from applications. This is illustrated in Figure 9. Here, the applications "C" and "D" no longer have persistent data storage (ie. proprietary databases). Rather, data is stored independently of the applications. Now, when the requirement for system "C" and "D" to be used as part of the same process, "C" will create data within the shared database; "D" will then access *the same data* for further processing, and will use the shared database to store its results.

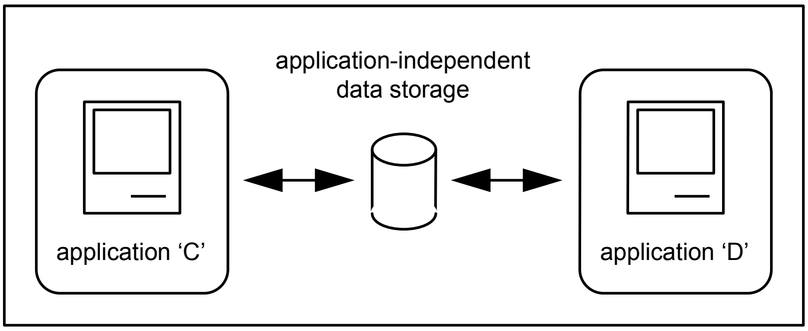

Figure 9: sharing of data between data-independent applications

It should be noted that some data translation will still take place, since the requirements for shared storage will frequently conflict with those for efficient processing. However, if:

- the applications "C" and "D" are *designed* to work with independent data storage, and
- data created to facilitate efficient processing within the applications is not stored persistently

then many of the potential problems discussed above will not arise. Since the data that is shared between the applications is stored only once, it is much easier to manage.

Data communication and data integration

The discussion above illustrates an important distinguishing feature between different possible solutions to data management requirements. Figure 8 illustrates solutions based on data *communication*, ie. the need to share information between people or organizations is fulfilled by communicating that information. Figure 9 illustrates solutions based on data *integration*, ie. the need to share information is fulfilled by having a single source of the information to which all users have access.

It is important to recognize that requirements for sharing and management of data require solutions of both types. In general, data communication is appropriate to sharing across enterprise boundaries, particularly where this sharing is in the context of a contractual interface. Any requirements associated with delivery of data are likely to be addressed using a data communications solution. Data integration is appropriate to the needs for data sharing within an enterprise, particularly where different individuals, departments or companies are working together in a shared environment on a common set of tasks.

Legacy systems

For many users, the requirement to maintain lifetime access to data often results in the need for maintenance of "legacy systems", ie. obsolete information technology systems retained within the organization *only* to provide access to data created using those systems. For example, many companies still maintain CAD systems acquired 10-15 years ago, long since replaced, but still needed to access drawings or other data created during the system's active life.

A true ability to make data independent of applications may provide a more effective solution to the problem of legacy data. If data can be accessed by *any appropriate application*, then the retention of legacy systems will no longer be necessary. The alternative is the creation and maintenance of system-independent product data archives. To be useful, such archives must support the long-term retention, in a usable form, of all the data relating to design and manufacturing.

Product data management

Product data management covers both the processes that are used to manage product data, and the computer systems that are used to implement and enforce these processes.[3] Product data management has become a very high-growth area of the information technology marketplace, reflecting the potential impact of data management problems as discussed above.

Engineering data management (EDM) is also a widely used term. This covers a number of related approaches and systems. "EDM" may be:

- **Engineering *Document* Management**: the use of database systems to catalogue and control documents in any form, including paper, microfilm, and raster (scanned) images.

- **Engineering *Drawing* Management**: a refinement of the above applied to "traditional" drawing office procedures and practices.

- **Engineering *Data* Management**: the use of independent database systems to manage data across multiple, heterogeneous design and engineering applications.

Within such systems, five different areas of functionality may be identified:

- **access management**: providing access to data for authorized users (and denying it to others);

- **product structure definition**: describing the structure of a product in terms of components and assemblies;

- **configuration management**: identification of versions of products and of data, and their valid combinations;

- **design review and approval**: maintaining records of reviews and the assignment of approvals to products and product data;

- **action management**: including effectivity and work-flow information.

All approaches to PDM offer some or all of these functions.

CAD systems, in general, are used to create and store representations of the shape (geometry) and other properties of products. Such systems do not, however, include the concept of the abstract "product" of which the shape is a property.

This product-data oriented functionality is today provided by a combination of CAD systems plus PDM systems. As is shown in Figure 10, a number of CAD systems may be used to create representations of different properties of a product; the PDM system is then used to define and manage the relationships between these in terms of the product or version of a product to which they relate.

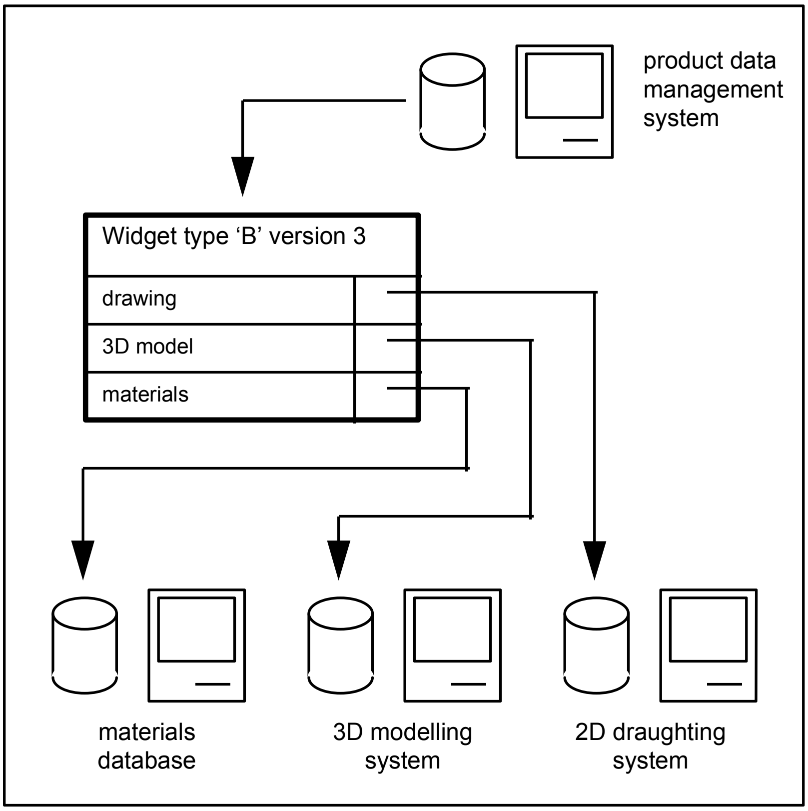

Figure 10: management of data in different systems

Product data management systems fulfil requirements for:

- the *logical* identification of elements of product data within a heterogeneous system;

- the *physical* locations of those data elements within one or more computer systems.

Data models

A data model provides the definition, structure, and format of data. Data models generally form the basis by which we organize and understand information. For example, a standard credit-card slip is based on an underlying understanding of the essential facts that are necessary for a complete transaction:

- the name and reference of the shop,
- the name of the credit card holder, the credit card number and expiry date,
- the amount to be charged to the credit card account,
- a confirmation number, and
- the card-holder's signature.

As well as defining the various boxes and spaces on the face of the credit card form, this set of information also defines how this information is stored or made available for processing in a suitable computer system. The data model does *not* define the specific values of data. In this example, the data model reflects the structure of the *blank* credit card form, not the actual names, numbers and amounts that are filled in for a given transaction.

The roles of data models

A data model is used as the basis for agreement of the structure and meaning of data. Such a model may serve different purposes:

- defining the internal data structures used by a piece of software;
- defining the data that is to be stored and managed by a database management system;
- defining the tables, rows, and columns within which data is stored in a database;
- defining the structure of a file used for data exchange.

Management of data, as discussed in the previous sections of this chapter, requires compatibility of data. For example, to manage information about the employees of a company it is essential to have agreed definitions and descriptions of the different items of information that describe an employee, and how these items are related to each other. A data model for information about employees will provide these definitions and descriptions.

It is a characteristic of data models that their form and content is determined by their purpose: therefore, a data model that defines the structure of an exchange file will not be ideally suited as the basis for a database.

The ANSI/SPARC architecture

The most commonly used basis for understanding different types of data models, and their roles, is the ANSI/SPARC architecture. This three-layer architecture was first published in 1975 by the ANSI/X3/SPARC Study Group on Database Management Systems. This model was proposed as the core of a framework for database management systems, but has since been widely used as a basis for other information technology system architectures.

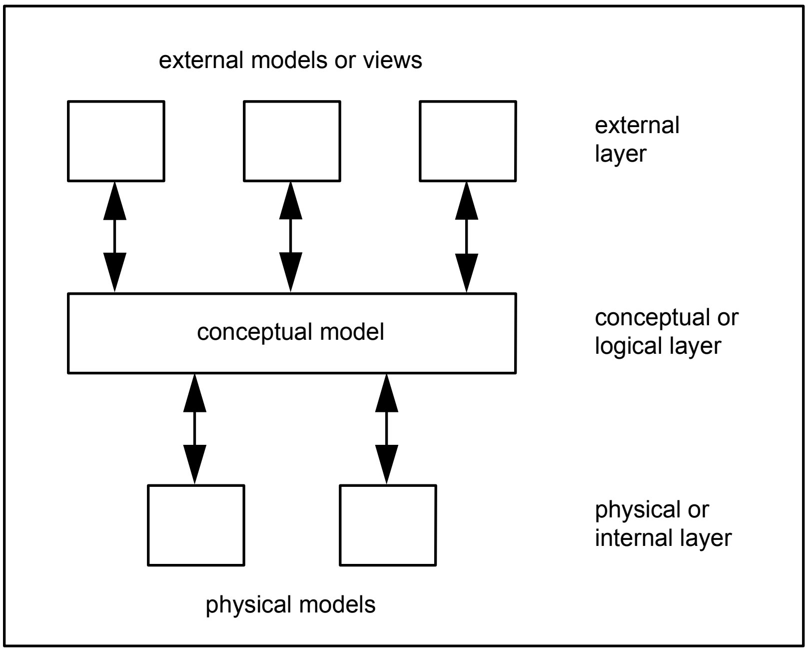

Figure 11: ANSI/SPARC three-layer architecture

The architecture specifies three types of model: external, conceptual and internal (see Figure 11). Each type of model has a specific use and a defined relationship with the other types. The three types of model (corresponding to three layers) work together within the architecture to support the concept of data that is accessible from multiple applications or business perspectives, and that may be stored in and moved between multiple implementation forms.

The conceptual layer contains a single model (within a given context) that is the basis for integration of the data used by different applications or stored in different formats. Models in this layer need to be stable over a long time scale, and so must not embody details that are specific to any application or any storage format, as these details are liable to change over short time scales.

The external layer contains one or more external models or "views". Each of these is specific to a given application or business view and maps to a subset of the conceptual model in such a way that the data described in the external model can be held in the format of the conceptual model.

The internal or "physical" layer contains one or more physical models. A physical model is a complete specification for a data structure which is implemented through statements written in an application-specific Data Definition Language. The data structure must conform to the physical model in the sense that an exact, reversible mapping between them is defined. Each physical model is optimized for a specific implementation scenario (eg. a relational database for archiving or a working form for computer-intensive processing). Any data structure that conforms to an internal model must be able to accept data that conforms to the associated conceptual model.

Data models in STEP

Many of the issues relating to the use of existing standards for CAD/CAM exchange, as discussed in Chapter 2, relate to the lack of formal, agreed models for the data that is to be exchanged. Additionally, the models that do exist within these standards do not separate the structure of the data that is encoded in an exchange file from its meaning.

As we will see in the next and subsequent chapters, the overall structure of the STEP standard is strongly influenced by these problems with previous specifications. STEP also borrows heavily from the concepts embodied in the ANSI/SPARC architecture.

Conclusions

The requirements for exchange and sharing of data discussed in Chapter 1 must be coupled with requirements for the management of data. The tools and techniques of data management, as applied to engineering systems and disciplines, give rise to Product Data Management (PDM) and Engineering Data Management (EDM). Such systems are now widely implemented within industry, providing a level of control over product definition data that is not possible from single CAD/CAM applications.

Data management requires formal definitions and descriptions of data, in the form of data models. Such models may have different roles and purposes. Architectures such as the ANSI/SPARC three-layer architecture allow these different roles to be understood and inter-related. The disciplines and techniques of data modelling enable complete, unambiguous specifications of data formats; the following two chapters, together with Appendix A, describe how these have been adopted and used within STEP.

---

[1] This definition is derived from that developed by the Information & Computing Division, Shell International Petroleum Company.

[2] The complete file name and path for this chapter on my Macintosh is: "LC475:The Book:book text:Ch3. Data management". Some elements of the hierarchy that I use to organize this particular system can probably be derived from this information!

[3] An *Executive Guide to Product Data Management* has been produced by the UK Department of Trade and Industry. This provides an introduction to the requirements and available solutions in this area.

4 The development of STEP

Previous chapters have presented the business and technology requirements from industry for a comprehensive product data standard. This chapter examines the history and status of STEP, tracing its development from initiation in the mid 1980s, to the publication of the initial release of the standard in 1994. Chapter 5 provides a description of the structure and contents of STEP; additional information relevant to this chapter is also to be found in Appendices A, B and C.

What is STEP?

"STEP" is, in fact, an unofficial name; the actual designation of the STEP standard is ISO 10303 "Industrial automation systems – Product data representation and exchange". The term is also frequently used to refer to the development activity that produces the standard. STEP is developed by the International Organization for Standardization (ISO). ISO's work is structured into a large number of technical committees, sub-committees, and working groups. The group responsible for STEP is:

- Technical Committee TC184 "Industrial Automation Systems and Integration",

- Sub Committee SC4 "Industrial Data" (ISO TC184/SC4).

Further information on the working groups within SC4 is given in Appendix C. Although ISO is responsible for a very wide range of standardization activities, those arising from the requirements of the electrical and electronic industries are undertaken by a separate organization: the International Electrotechnical Commission (IEC).[1] A joint working group of ISO TC184/SC4 and IEC TC93 addresses the electrical and electronic industry aspects of STEP.

It is common, particularly in the USA, to see references to the standard as "PDES". Although at one time this acronym stood for "Product Data Exchange Specification", suggesting a separation between the US and international standards, the interpretation of the acronym was changed in the early 1990s to "Product Data Exchange using STEP", and is now taken to refer to the US activities contributing to the international standard development activity.

In all, eighteen countries participate in the development of STEP, drawn primarily from the leading industrialized areas of Europe, North America and the Pacific Rim. These countries are represented by their national standards bodies such as BSI in the UK, DIN in Germany, and AFNOR in France. These bodies are responsible for co-ordinating national input to the ISO standards committees, facilitating reviews of draft standards by industry, and determining voting positions when draft standards are circulated for approval.

National standards bodies generally only undertake these co-ordinating activities. The actual development of the standard is done by technical experts drawn from industry, research organizations, and academia. World-wide, there are several hundred such experts contributing to the development of STEP, with an increasing additional number involved in tracking, implementing and deploying the standard in industry.

Contributors to the standard are drawn from a wide range of backgrounds. Many industrial companies participate in STEP; much of this participation is undertaken in the context of collaborative projects through which companies share resources and risks. Summaries of a number of these projects are given in Chapter 9.

National STEP Centres have been identified in a number of countries and, as well as undertaking development and implementation projects, facilitate training and technology transfer for national industries. These Centres also provide links between industry and the various research and academic institutions who undertake some of the underlying research and development necessary to the standardization process.

The history of STEP

As explained in Chapter 2, initial efforts to create specifications for CAD/CAM data exchange resulted in a number of national standards (IGES, SET, VDA-FS) that achieved limited success in providing interfaces between proprietary CAD/CAM systems. By the mid 1980s, however, it had become apparent that industry's needs would only be properly addressed by a more comprehensive international effort that would not only improve on the existing specifications, but also fulfil requirements for life cycle product data support.

The STEP project was therefore initiated in 1984, with the following objectives.

- The creation of single international standard, covering all aspects of CAD/CAM data exchange.

- The implementation and acceptance of this standard by industry, superseding various national and *de facto* standards and specifications.

- The standardization of a mechanism for describing product data, throughout the life of a product, and independent of any particular system.

- 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.

Work towards these objectives started in mid 1984. Following an analysis of the capabilities of existing specifications, including the results of the ESPRIT CAD Interfaces (CAD*I) and US Air Force Product Definition Data Interface (PDDI) projects, it was agreed that no adequate, interim standard solution was available. The PDES Initiation Effort proposed that the new standard should be based on a "three layer" architecture, significantly influenced by the ANSI/SPARC three-layer architecture for database systems (see Chapter 3).[2] The layers within this architecture were:

- an **applications layer**: data models concerned with individual applications or disciplines such as mechanical products, electrical products, and building & construction;

- a **logical layer**: generic data models describing the common concepts used by all product data applications, such as product structure, shape (geometry and topology), and presentation;

- a **physical layer**: a file format for data exchange.

Several different modelling languages were used to describe the various application models proposed for STEP. This increased the difficulty of integrating these models. It was therefore decided that STEP should develop a computer interpretable data specification language that could be used to describe all the data models within the standard. This language, based on initial work undertaken within McDonnell Douglas Information Systems in the USA, became known as "EXPRESS". The combination of EXPRESS and the three-layer architecture allowed a clear separation between:

- a method for the description of data models (EXPRESS);

- the data models themselves (applications and logical layer);
- a mechanism for the exchange of data (file format).

Over the next three years, these elements of the STEP standard were developed, including the EXPRESS language, the exchange file specification, and the basic data architecture. In addition, a number of data models were developed describing the requirements for product shape (geometry and topology), engineering drawings, and industry-specific needs such as those of electrical applications and shipbuilding.

In 1988 these data models were collected together as an Integrated Product Information Model (IPIM). The IPIM, together with the EXPRESS language and the exchange file specification, was published by ISO as a Draft Proposal in November 1988.[3] This document was reviewed by the countries participating in the development of STEP as part of the ISO balloting process. Given the considerable variation in the stability and completeness of the various elements of the document, it was no surprise that the Draft Proposal was not approved for publication as a standard without further technical work.

Following the rejection of the Draft Proposal, a number of key decisions were made in mid 1989 that have determined the structure, content, and direction of the continued work on STEP. The most important of these decisions were:

- the division of the standard into parts;
- the adoption of Application Protocols;
- the identification of a minimum set of parts as the basis for the first version of the standard.

It is common practice to divide large standards into parts to facilitate managed, phased development and publication within a large and complex technical domain. Given that the Draft Proposal had been some 2,500 pages in length, the division of STEP into parts was seen as an absolute necessity. Details of the document structure of STEP are given in Chapter 5.

Although the data architecture employed in the development of the Draft Proposal recognized a distinction between common "core" models and application-specific models, it had not effectively addressed the problems manifested in other standards such as IGES due to implementations being based on vendor-specific subsets of the standard. Application Protocols were introduced into STEP as a mechanism for the identification of controlled, application-specific views within the standard. These views then provide a basis for implementation, and for the testing of implementations, to meet specific industry needs within the overall scope of STEP.

The “initial release”

Analysis of the contents of the Draft Proposal in terms of the division of the standard into parts and the introduction of Application Protocols resulted in the identification of more than twenty elements of the standard on which development was to be continued. It was recognized, however, that the long-term success of STEP would be supported by identifying a minimal set of parts for “fast track” completion and publication. The requirements for the “first release” of the standard were identified in mid 1990, and much of the effort of the STEP development activity over the following four years was directed to the successful completion and publication of these parts.

The basis for the initial release was the inclusion of at least one Application Protocol, together with all necessary additional parts required to support neutral file exchange based on Application Protocols. In fact, the initial release contains two Application Protocols supporting requirements for exchange of engineering drawings, and for exchange of configuration controlled 3D design data.

A total of 12 parts of STEP comprise the initial release. All had been issued for Committee Draft ballot by November 1992, approved and issued for Draft International Standard ballot by November 1993, and approved for publication as International Standards by September 1994. All 12 parts of the initial release were published by ISO early in 1995.

Beyond the initial release

Although the technical effort in STEP between 1990 and 1994 concentrated on the development of the initial release, continued parallel development of other elements of the standard has been actively pursued. In addition to the 12 initial release parts, more than 60 additional parts of the standard were in development in 1995.

There are three major areas within which further development is being undertaken.

- Development of additional Application Protocols, supporting the needs of areas as diverse as manufacturing applications, shipbuilding, the process plant and petro-chemical industries, and automotive design.

- Development of a standard data access interface as part of the support for shared databases.

- Further development and enhancement of the EXPRESS data definition language.

Over the ten years since its initiation, STEP has not only grown but has also altered in response to the changing needs of industry. In 1984, the key requirements were for a standard that could replace IGES, SET, and VDA-FS as a more effective and efficient mechanism for the exchange of CAD/CAM data. Today, however, STEP has expanded to address industry’s needs for system-independent management of product data across the full life cycle. The success of pilot programmes based on the initial release, as discussed in Chapter 9, and the continued expansion in the range of industries predicating their future information technology needs on the availability and use of STEP, suggests that STEP has and is fulfilling both of these aims.

Why has STEP taken so long?

It is undeniable that STEP has taken a long time to develop – considerably longer, in fact, than envisioned when the STEP project was initiated in 1984. Given the ever-increasing rate of change in the information technology industry, how can this long gestation period be explained, and does it mean that STEP is lagging behind advances in information technology?

It has to be recognized that the development of STEP is unusual in comparison with other standards. Many standards are the result of agreement and codification of accepted industry methods and practices. STEP, by contrast, has been as much a research project as a standardization activity for much of its lifetime. STEP is part of a trend for information technology standards to be proactive, ie. setting standards for the future rather than capturing the common aspects of past or current ways of working.

As a result, there has been a vital need within the development of STEP to receive and incorporate the results of prototype implementations and pilot projects. This synergy between research, prototyping, and standardization activities has inevitably led to delays in the development of the standard. The methodology for the development of STEP has, in effect, been produced in parallel with the standard itself.

In some ways, the current procedures of international standardization work against the needs of industry for continuous improvement and evolutionary change. The processes of development, review, approval, and publication of standards through ISO and national standards bodies are slow and cumbersome. In other areas information technology standards have been developed and published by industry groupings independently of the formal international processes. Although this allows much greater flexibility and responsiveness in the resulting standards, they do lack the authority, stability and legal status of standards produced by ISO and/or IEC.

The experience in the development of the initial release of STEP has resulted in considerable refinement not only in the technical development process but also of the effective use of the ISO/IEC standardization procedures. As a result, the development cycle from the initial identification of an industry need to be fulfilled by STEP, to the publication of one or more parts of STEP meeting that need has been reduced to between two and three years. Although this is a short time in terms of standards development, it does highlight the need for the standard to recognize and accommodate flexibility, so that any changes in industry practices or requirements during the standardization period do not render the contents of the standard inappropriate or unusable.

Summary

The STEP development activity was initiated in the mid 1980s within the International Organization for Standardization (ISO). The development, which is supported by most of the leading nations of the industrialized world, has led to the definition and publication of ISO 10303 "Product data representation and exchange".

The standard is divided into a large number of parts addressing different elements of the total requirements on STEP. An "initial release" of the standard, fulfilling industry needs for exchange of engineering drawings, and of configuration controlled 3D design data, was completed in 1994 and published in 1995. Additional parts of the standard are being developed to meet the product data standards needs of many industry sectors.

[1] ISO and IEC work very closely together, and their Central Secretariats occupy adjacent office buildings in Geneva.

[2] Initial work on the development of STEP undertaken by the IGES/PDES Organization (IPO) in the USA.

[3] The procedures of ISO have been changed during the development of STEP. The term "Draft Proposal" is no longer used, the current equivalent being Committee Draft (CD).

5 The structure and content of STEP

STEP is not a single document. As discussed in the previous Chapter, it is divided into a number of parts, each of which is published separately. These parts are organized according to a defined document structure, which reflects the underlying architecture of the standard. This Chapter uses the document structure as the basis for a "walk-through" of the essential elements of STEP, and also includes a "readers' guide" to STEP, indicating which elements of the standard are relevant and useful to different groups of people with an interest in STEP.

The underlying architecture of STEP is discussed here only as it is relevant to the document structure; further information is given in Appendix A. A complete list of the parts of the STEP standard is given in Appendix C.

The STEP document structure

As discussed in Chapter 4, the STEP standard was divided into a number of parts following review of the 1988 Draft Proposal. Rather than numbering parts sequentially as they are developed, a document numbering scheme was developed that classifies each part of the standard according to its content. Each part is, of course, a component of the STEP standard, ISO 10303 "Industrial automation systems – Product data representation and exchange". STEP has been divided into seven separate classes of parts; each class of parts is associated with a block of part numbers within the ISO 10303 standard. Figure 12 illustrates the major elements of the STEP document structure.

Introductory parts

The first class of parts (not included in Figure 12), is the *introductory* class that describes the overall structure of the standard, and the relationships between its various elements. There is (so far) just one part of STEP within the introductory class:

- Part 1: "Overview and fundamental principles".

which defines the basic principles of STEP, the characteristics of the other parts of the standard, and the relationships between them. Part 1 is part of the initial release of STEP, and is required reading for anyone with interest in using, implementing, or further developing the standard.

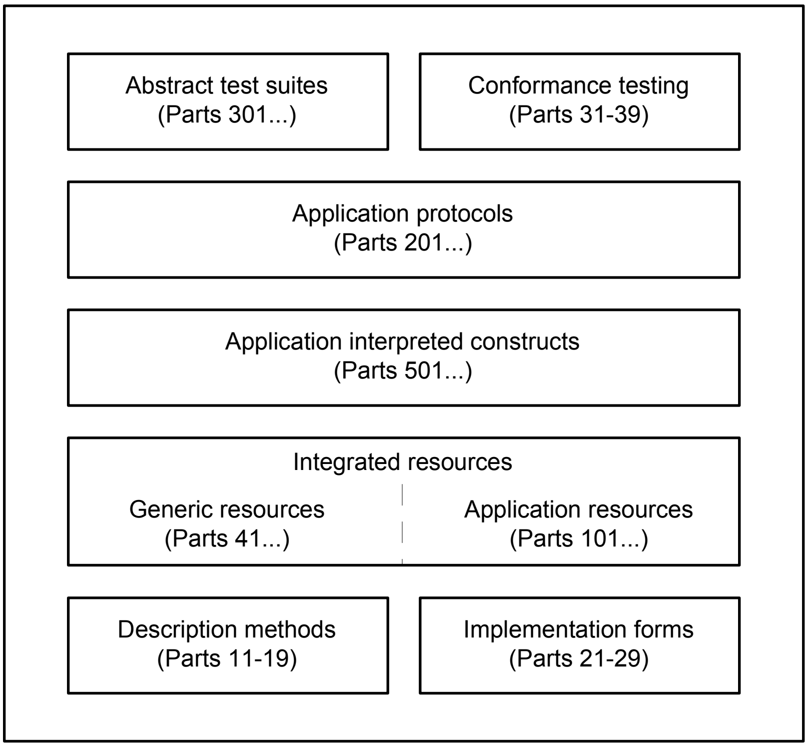

Figure 12: the STEP document structure

Description methods

The scope of the *description methods* class is that of the languages and methods used to create standard representations of product data. The description methods have been allocated part numbers between 11 and 19. To date, three parts have been identified within the description methods class:

- Part 11: “The EXPRESS language reference manual”;
- Part 12: “The EXPRESS-I language reference manual”;
- Part 13: “Architecture and methodology reference manual”.

Part 11 is included within the initial release of STEP.

One of the key objectives of STEP is to provide unambiguous, computer-interpretable representation of product data. This is supported through the use of the EXPRESS language. EXPRESS is a data specification language that is used to represent the structure of data and any constraints that may apply to it. The data models contained in the STEP Integrated Resources and Application Protocols are defined using EXPRESS.

Although EXPRESS resembles some programming languages, it cannot be used to define executable programs; rather, it is used to define the data on which programs operate. EXPRESS supports:

- the definition of data entities, attributes, and relationships;
- the specification of local and global constraints on these;
- the collection of data definitions and constraints in separate schemas, supporting modular development of data models.

Although EXPRESS has been developed as part of STEP, it is now widely used in other standardization, research, and integration projects. Further information about the EXPRESS language is given in Appendix B; Chapter 10 includes a summary of some of the software tools that are available to work with EXPRESS.

The second STEP part in the description methods class is EXPRESS-I. This is a standard data instance definition language, ie. it may be used to specify actual values within an EXPRESS schema. EXPRESS-I is particularly useful in the specification of test data.

Part 13 of the standard describes the detailed structure of STEP and the methods used to develop its various elements. It is intended to provide a definitive statement of the STEP architecture and methods, to enable the application of the methodology, and to serve as a basis for training. A summary of the STEP architecture and methodology is given in Appendix A of this book.

Implementation forms

One of the key differences between STEP and previous standards in the areas of CAD/CAM data exchange or data management is the separation of data definition from implementation. Thus within STEP the data models defined in the standard are designed to be independent from the various ways in which they may be implemented. The *implementation forms* class of parts defines standard formats for data instances and values, and the mappings between these formats and the EXPRESS language.

The implementation forms have been allocated part numbers between 21 and 29. To date, five parts have been identified within the implementation forms class. The most important of these are:

- Part 21: "Clear text encoding of the exchange structure" defines the standard format for encoding of data in a file, and supports the exchange of data between applications.

- Part 22: "Standard data access interface" enables access to product data within an application (an "engineering" application or a database management system) independently of the internal form of data storage within the application.

Part 21 (commonly referred to as the "physical file" format) is part of the initial release of STEP. As well as Part 22 (commonly referred to as the "SDAI"), several additional parts within the implementation forms class define the bindings between the SDAI and specific programming languages including C, C++ and FORTRAN.

An implementation form is not sufficient to define the complete requirements for a conforming implementation of STEP; in addition, a data model is required that defines the structure and semantics of the data to be handled by the implementation. Within the architecture of STEP, these data models are provided within Application Protocols, as described below. An implementation of STEP combines an Application Protocol with an Implementation Form; thus, the same data model is used (for example) for exchange of engineering drawings using files (on disc or tape) or for standard access to drawings within a database.

Chapter 6 provides additional information on implementation forms, and their place within the implementation architecture of STEP.

Integrated resources

In Chapter 3, the idea of a conceptual data model was introduced. Within STEP, a single conceptual data model has been developed that reflects and supports the common requirements of many different product data application areas. This conceptual data model is modular in nature, and is published within the *Integrated Resources* class of parts. Even though the Integrated Resources are documented as a number of separate parts, the data model that they contain logically constitutes a single, integrated conceptual product data model.

For the purposes of publication, Integrated Resources are divided into two separate series of parts:

- the **integrated generic resources** (Parts 41…99) define the components of the conceptual product data model that are independent of applications; for example, Part 42 "Geometric and topological representation" defines the standard representations for the shapes of objects, independent of any specific use of the shape information;

- the **integrated application resources** (Parts 101…199) extend the generic resources to support the needs of specific groups of applications; for example, Part 101 "Draughting" defines the common data requirements of all applications that incorporate or make use of engineering drawings.

The Integrated Resources provide the developers of STEP Application Protocols with standard definitions of product data; they are not themselves intended for direct implementation. In this respect they may be loosely compared to a collection of library routines used by programmers: Integrated Resources define reusable components that are intended to be combined and refined (within an Application Protocol) to meet a specific need.

Six parts in the integrated resources series are included in the initial release of STEP:

- Part 41 "Fundamentals of product description and support";

- Part 42 "Geometric and topological representation";

- Part 43 "Representation structures";

- Part 44 "Product structure configuration";

- Part 46 "Visual presentation";

- Part 101 "Draughting".

Several other Integrated Resource parts have been developed for later publication, as required to fulfil the requirements of new Application Protocols.

Application protocols

By far the largest, and in many senses the most important class of parts is the *Application Protocols* class. Application Protocols are numbered from part 201 onwards; two Application Protocols are included in the initial release of STEP:[1]

- Part 201: Explicit draughting;
- Part 203: Configuration controlled design.

Many others are being developed for later publication.

Application Protocols define and fulfil the requirements of an identified application of product data related to a specific industry need. This should be contrasted with the generic, application-independent nature of the Integrated Resources. It is noted above that Part 42 of STEP defines standard representations of geometry: an Application Protocol will, for example, define *how* these representations are used to exchange the designed shape of the wing of an aeroplane, or the in-service shape of the hull of a ship, or the shape of the access area around a pump within a petro-chemical plant. These are examples of uses of shape that may be supported by appropriately designed Application Protocols.

Part 1 of STEP defines an Application Protocol as:

> *"a part ... (of ISO 10303) ... that describes the use of integrated resources satisfying the scope and information requirements for a specific application context."*[2]

This definition highlights a key characteristic of Application Protocols: they are uses of the STEP Integrated Resources, but do not extend the data model defined in the Integrated Resources. This characteristic ensures a high degree of uniformity across different Application Protocols, ensuring that common or similar requirements are satisfied using a common or similar solution.

The role of an Application Protocol is to provide the basis for implementations of STEP, and to enable the assessment of conformance of implementations. As noted in Chapter 2, Application Protocols were introduced into IGES as a mechanism for defining controlled, meaningful subsets within the standard. STEP has extended and refined this concept considerably, and the development of STEP is now focused on the development of Application Protocols to meet the requirements of a wide range of industry sectors and application areas.

Clearly, this diversity has to be managed: otherwise, STEP would consist of a collection of separate, industry- or discipline-specific Application Protocols. The costs of implementing and supporting these would be very high as new interfaces would have to be written from scratch for each Application Protocol. However, the diversification of Application Protocols is not only managed; it is a conscious design intent of STEP that Application Protocols should be consistent. This consistency is achieved at three levels:

- each Application Protocol is reviewed in detail during its development to identify overlaps in scope and requirements with other Application Protocols;

- where identical requirements exist within the scopes of two or more Application Protocols, these are fulfilled using common data modelling constructs;

- Application Protocols share a basis in the STEP Integrated Resources; this ensures consistency across the totality of all Application Protocols.

Each Application Protocol is divided into a number of separate, inter-related sections. This structure is common to all Application Protocols and not only encourages consistent development but also eases and aids the processes of review, implementation, and use. There are four major components of an Application Protocol.

- The *scope* of the Application Protocol specifies the industry processes and data that it is designed to support. The scope is related to an Application Activity Model ("AAM"), a graphical model of the industry activities that the Application Protocol supports.

- The *information requirements* that the Application Protocol supports: these are specified as definitions of data ("application objects") and the relationships and constraints that apply to data ("application assertions"). The information requirements are specified using the language and terminology of the application area supported, and are therefore designed for understanding and review by relevant industry experts. The definition of information requirements is supported by an Application Reference Model ("ARM"), a graphical model of the data entities and relationships within the scope of the Application Protocol.

- The *Application Interpreted Model* ("AIM") is an EXPRESS data model that defines how the information requirements are satisfied using the STEP Integrated Resources. Constructs from the Integrated Resources are selected and constrained to create a data model that not only fulfils the specific requirements of the Application Protocol, but is also consistent with other Application Protocols. The Application Interpreted Model is supported by a Mapping Table, that specifies how each identified information requirement (application object and application assertion) is satisfied.

- The *conformance requirements* that apply to implementations of the Application Protocol. In order to avoid problems caused by implementations of vendor-specific subsets, as described in Chapter 2 with respect to IGES, STEP Application Protocols require *completeness* of implementation, ie. that an interface conforming to the Application Protocol should support every entity, attribute and constraint specified. However, in recognition of the differences between the computer systems that are used within a given application area, many Application Protocols identify a number of "conformance classes" that specify subsets of the Application Protocol for which conformance may be claimed and assessed. The completeness requirement then applies within each conformance class.

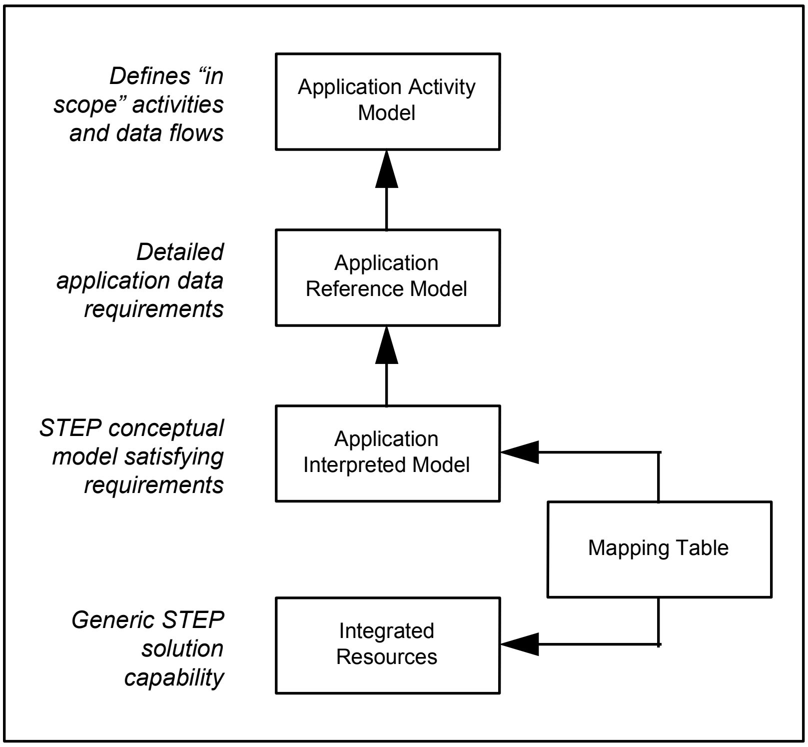

Figure 13: relationship between models

The presence of several different models within an Application Protocol can be confusing to the reader. The need for each model, and the relationships between them, can be explained by reference to Figure 13.

The Application Activity Model depicts the industry activities that are to be supported by the Application Protocol. The Application Reference Model defines the detailed data requirements for the "in scope" information flows defined in the AAM. The Application Interpreted Model fulfils the data requirements specified in the ARM, and is creating by selecting and constraining elements taken from the Integrated Resources.

The Application Reference Model, the Application Interpreted Model, and the Integrated Resources are related and cross referenced through the Mapping Table.

Further information on the relationship between the models contained within Application Protocols is given in Appendix A.

Application interpreted constructs

As noted above, where identical requirements exist within the scopes of two or more Application Protocols, these are fulfilled using common data modelling constructs. These common constructs are published within the *Application Interpreted Constructs* class. Application Interpreted Constructs (AICs) are numbered from Part 501 onwards; no AICs are included in the initial release of the standard. Further information on AICs is given in Appendix A.

Conformance testing

As discussed above, Application Protocols are included within STEP as the basis for implementation and for the testing of implementations. It has taken a long time for the importance of testing and testability to be accepted: for example, explicit conformance requirements were not introduced into IGES until the late 1980s, and conformance testing services for IGES and other current standards for CAD/CAM data exchange have been available only since the early 1990s.

STEP, in contrast, has recognized the importance of testing since its inception. The *Conformance Testing Methodology and Framework* class of parts provides test laboratories, implementors, and end-users with the basis for consistent, comprehensive conformance testing of implementations of STEP. This basis has been derived not only from experience of the development and use of test methods for other CAD/CAM data exchange standards, but also the lessons learned in the creating and maintaining conformance testing procedures and services for the OSI "Open Systems" standards.

Parts in the Conformance Testing Methodology and Framework class are numbered between 31 and 39. One part in this class is included in the initial release of STEP:

- Part 31: General concepts.

This part defines the basic principles of conformance testing for STEP. Other parts in this class define the specific requirements on test laboratories, clients for testing services, and details of the methods used in assessing the conformance of implementations of Application Protocols when combined with both the physical file format (Part 21) and the Standard Data Access Interface (Part 22).

Abstract test suites

One of the requirements defined in the Conformance Testing Methodology and Framework is that an Abstract Test Suite should be available for each STEP Application Protocol. An Abstract Test Suite defines detailed requirements for the assessment of conformance, and includes a number of test cases that are to be used by all test laboratories. The standardization of Abstract Test Suites is recognized as a vital component to the harmonization of testing activities and the acceptance of test results on a world-wide basis. Experience in testing Open Systems products has shown that without standardized test suites the same product can give different results in different test laboratories. Such differences not only fail to give end-users the assurance of conformance that testing is intended to give, but also leads to undesirable trade barriers in the form of national or local requirements for testing and non-acceptance of other test results.

The Abstract Test Suite parts of STEP are numbered from 301 onwards: Part 301 specifies the Abstract Test Suite for Application Protocol Part 201, and so on. There are no Abstract Test Suites in the initial release of STEP.

Do I need to read *all* of STEP?

Even the initial release of STEP comprises several thousand pages of detailed technical specifications, and the totality of the standard – including parts in development – occupies several feet of shelf-space. Readers will be relieved to know that few people will either need or want to read each and every part of the standard. The document structure adopted for STEP allows an easy identification of the class of parts, and the individual parts within a class, applicable to a particular user of STEP.

The only part that *everyone* with an interest in STEP is encouraged to read is Part 1 “Overview and fundamental principles”. This not only provides the definitive statement of the scope and purpose of STEP, but also defines the key terms that are used frequently throughout the remainder of the standard.

- **EXPRESS**: data modellers, including those working on further development of the standard, need to be familiar with the EXPRESS language. System designers and programmers responsible for the development of STEP compliant interfaces also need a strong understanding of EXPRESS.

- **Implementation forms**: the details of implementation forms are of interest and use to system designers and programmers responsible for the development of STEP compliant interfaces.

- **Integrated Resources**: detailed examination of the STEP Integrated Resource models is necessary only for those involved in the continued development of the standard itself. Data modellers engaged in other activities may find these models of interest in understanding and making use of the STEP methodology.

- **Application Protocols**: with the exception of Part 1, Application Protocols are the only class of STEP parts of detailed interest to end-users in industry. Chapter 11 includes a recommended approach to reading and understanding Application Protocols and relating their contents to enterprise requirements. End-users will concentrate their attention on the scope and information requirements, and the supporting Application Activity Model (AAM) and Application Reference Model (ARM). These components of an Application Protocol allow an end-user to assess its suitability for use.

  System designers and programmers responsible for the development of STEP compliant interfaces will make use of the Application Interpreted Model (AIM), the Application Reference Model (ARM), and the ARM-AIM mapping table. These components specify the information requirements for an implementation of the Application Protocol. System designers and programmers will also make use of the Conformance Requirements (including the definitions of any Conformance Classes), in determining the options within the Application Protocol for claims and testing of conformance.

  Test realizers and test laboratories will make use of an Application Protocol in preparing and undertaking conformance testing of implementations.

- **Abstract test suites**: the primary users of Abstract test suites are organizations involved in conformance testing. The contents of an Abstract test suite are also of interest to system designers and programmers, as they provide reference test cases which may be useful during interface development.

- **Application interpreted constructs**: these are of relevance not only to those involved in the continued development of the standard itself, but also to system designers and programmers. AICs may be used as the basis for a "modular" design of applications or interfaces that are intended to support exchange using more than one related Application Protocol.

Summary

The parts of the STEP standard are structured and numbered as a collection of classes, as summarized in Table 3 below.

Part numbers	Class
1-9	Introductory
11-19	Description Methods
21-29	Implementation Forms
31-39	Conformance Testing Methodology & Framework
41-49 101-199	Integrated Generic Resources Integrated Application Resources
201-…	Application Protocols
301-…	Abstract Test Suites
501-…	Application Interpreted Constructs

Table 3: ISO 10303 part classes

Each class of part plays a different role in the overall structure of the standard, and is relevant to different readers and users. For both end-users and system implementors, the important components of STEP are Application Protocols which define standard data models for many different areas of industry requirements, and implementation forms, that determine how the data models are used to achieve data exchange and sharing. The other classes of part are more restricted in their audiences, being applicable to data modellers, STEP developers, or testing laboratories.

---

[1] It is common to refer to STEP application protocols by the abbreviation “AP”; specific applications protocols are then referenced as “AP201”, “AP202”, and so on.

[2] ISO 10303-1:1994

6 STEP implementation

Implementation levels

One of the key technical differences between STEP and previous CAx data exchange standards is the separation of data models (*what* is to be exchanged) from implementation forms (*how* it is to be exchanged). This distinction derives from the ANSI/SPARC model for database systems, as discussed in Chapter 3, and is one of the key design goals of STEP.

Through this separation of data models and implementation forms, different approaches to implementation may be taken depending on the specific requirements that are to be fulfilled. As part of the initial development of STEP, four different levels of implementation were identified:

- level 1: passive file transfer;
- level 2: active file transfer;
- level 3: shared database access;
- level 4: integrated knowledge-base.

Level 1 exchange is the use of STEP to achieve the same process as that described in Chapter 2 (see Figure 5). A pre-processor translates data from the internal format of the sending system and encodes this using the STEP physical file format. This file is transferred to the receiving system, where a post-processor reads the data and translates it to the internal format of the second system. The details of this process, and the relationship of the various elements of STEP to the stages in the process, are explored later in this chapter.

Level 2 is an extension to the first level: here, the data in the sending system is translated into a "working form" that allows selection and modification of the data as part of the translation process. In fact, most file-based implementations of STEP make use of this approach to a greater or lesser extent, and the distinction between level 1 and level 2 implementations has effectively disappeared. The "level 2" approach is also the basis of advanced direct translators that employ the "hub and spoke" architecture described in Chapter 2.

Level 1 and 2 implementations apply the advances of the STEP standard and its associated technologies to the problem of data exchange between systems using neutral files. This level of implementation of STEP may therefore be regarded as a "better mousetrap" with respect to previous standards such as IGES; the solution is better, and provides a higher level of functionality, but is still solving the same problem. Level 3 and 4 implementations, however, represent not only a significant advance in technology, but also address a different set of problems: those associated with data sharing. This is particularly significant given the views of advocates of business process re-engineering who have characterized data exchange as being symptomatic of "broken processes".

A level 3 implementation of STEP combines translation with data access. The translation element is equivalent to that for file exchange, ie. converting from the internal format of a CAx system into the form prescribed by STEP. However, rather than storing the STEP data in a file for the purpose of exchange, an underlying database or repository is used to store the data. The data access element of a level 3 implementation is a standard interface to this underlying database, that allows applications to store, manipulate, and above all share the data in a standard manner. Again, the details of these processes are discussed later in this chapter.

Level 4 implementation (integrated knowledge) envisions the combination of STEP with knowledge-based systems and artificial intelligence (AI). A level 4 implementation may be seen as a component in an "intelligent design environment" in which designers and engineers are supported by advanced information technology. Level 4 implementation remains in the domain of basic research and development, and will not be discussed further.

How does data exchange using STEP work?

Figure 14 below illustrates the process of data exchange using STEP. Comparison with Figure 5 demonstrates that this shows only half the picture: the operation of the pre-processor in translating data from the sending system into STEP. However, the second stage of a complete translation is simply a reversed process.

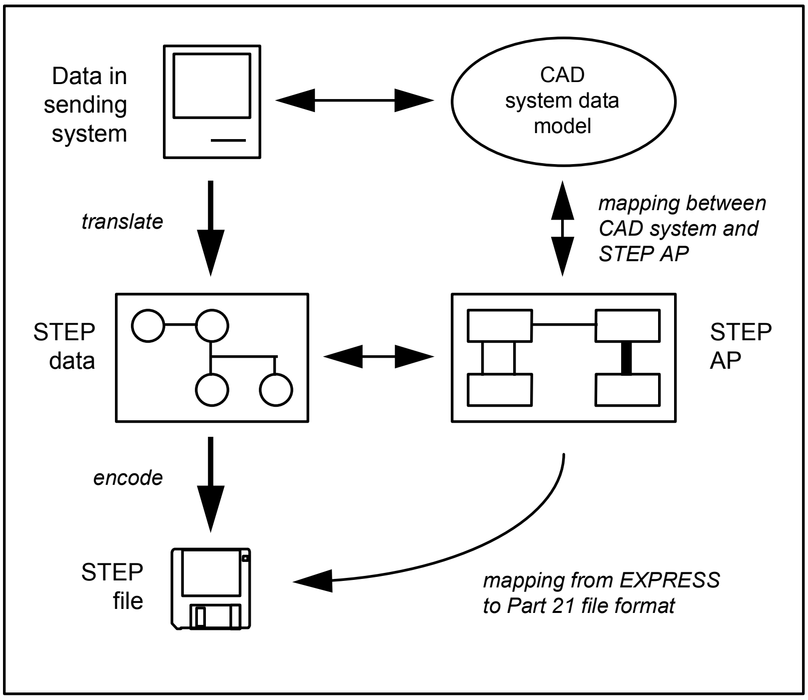

Figure 14: data exchange using STEP

The left hand side of the diagram represents data instances, ie. actual values. The right hand side represents the models of data that are used: it is these models that determine the translation requirements that are satisfied in an implementation. It will be seen that there are two data models that have to be taken into account: the internal data model of the CAx system, and the data model of the STEP Application Protocol that defines the data to be exchanged.[1] In creating a STEP interface for the CAx system, the software designer or programmer will create a mapping between these models, such that even if the structure of data in the two models is different, the meaning of the data is preserved.

For example, the CAx system may store information about straight lines within a model or a drawing using the co-ordinates of the start point of the line and the co-ordinates of the end point of the line. If the STEP Application Protocol that is used to exchange this data requires that straight lines are stored using the co-ordinates of the start point plus the length and direction of the line, then the programmer responsible for the interface must define a transformation of the data from one form to the other. Since this transformation is defined by reference to the two data models, actual data values are always translated in the same way. Using the same example, if a model within the CAx system includes a straight line that starts at co-ordinate position (0,0,0) and finishes at co-ordinate position (4,3,0), then the translation process will create, as part of the STEP data, a line starting at (0,0,0) with a length of 5 units and a direction ratios of 4:3. This is illustrated in Figure 15; note that although the data values are different as the result of the translation, the same line is still described, ie. the information, or the meaning of the data has been preserved.

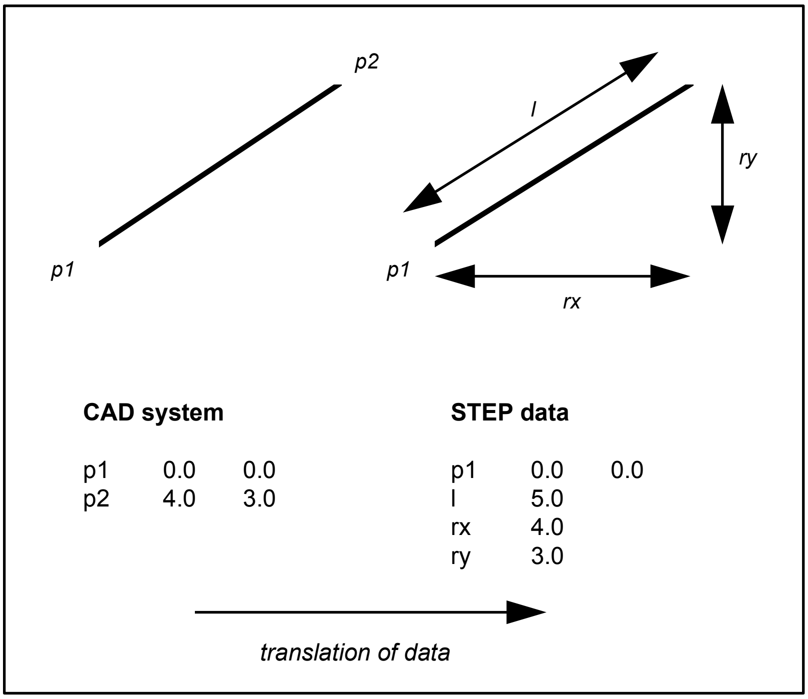

Figure 15: translation of data into STEP

The STEP file format

The process described above is that of *translation*, ie. converting the data from the internal format of the CAx system into that prescribed by the STEP Application Protocol. In order to exchange this data using disk, tape, or EDI, it is necessary to *encode* the data in a suitable form. This is accomplished using the STEP file format, specified in ISO 10303-21. This part of STEP specifies a standard representation form for data instances that conform to a data model that is specified in EXPRESS.

The file format uses a character-based encoding form based on ISO 8859, the accepted standard for representation of characters using 8-bit encoding.[2] The use of a character-based encoding format means that STEP files are portable across all major operating systems and computing environments, and can be transferred across networks using e-mail or EDI. Unlike IGES, which also uses a character-based encoding but in a format based on the requirements of the FORTRAN programming language and 80-column punched cards, the STEP file uses a simple sequential format. Also unlike IGES, the STEP file format has a formal definition, using the Wirth Syntax Notation (WSN), and a defined mapping from EXPRESS to the file structure. This mapping means that *any* data model defined in EXPRESS can serve as the basis for the exchange of data using files; it is not necessary to define a file mapping for each data model.[3]

The following example illustrates how the STEP file format is used to encode data. An EXPRESS data model includes the following definitions.

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

ENTITY person;
  first_name : STRING;
  last_name  : STRING;
END_ENTITY;
```

Instances of this data model that represent a 1989 Ford Orion, owned by the author, are encoded using the STEP file format as follows:

```
#1 = CAR ('Ford', 'Orion', 1989, #2);
#2 = PERSON ('Julian', 'Fowler');
```

This example has been formatted as separate lines for instances of each entity type; as noted above, the file format is essentially sequential in nature and so this division into lines is artificial. For the purposes of exchange, the same data would appear in a file as follows:

```
#1=CAR('Ford','Orion',1989,#2);#2=PERSON('Julian','Fowle
r');
```

Data exchange software architecture

The specification of the STEP file structure was one of the first elements of the standard to reach stability, and has therefore supported the development of many advanced prototypes and commercial tool kits and translators. Figure 16 below illustrates a typical software architecture for STEP data exchange.[4]

This illustrates how a modular approach to developing interfaces can greatly reduce the overheads in creating STEP interfaces:

- only the pre-processor "front-end" and post-processor "back-end" are specific to a given CAx system;

- the intermediate data structures are derived directly from the EXPRESS data models specified in STEP;

- conversion libraries and data validation utilities allow the data to be manipulated and checked as part of the translation process.

The creation of pre- and post-processors for another system therefore requires development only of the "front-end" and "back-end" elements. Advanced architectures for STEP data exchange include not only the ability to create and maintain the internal data structures directly from an EXPRESS data model, but also the creation of mappings between different EXPRESS data models. If the internal data model of the CAx system is available in EXPRESS, then the development of a translator resolves to the definition of the mapping between the CAx system internal data model and the STEP Application Protocol data model. Software tools that embody this approach are described in Chapter 10.

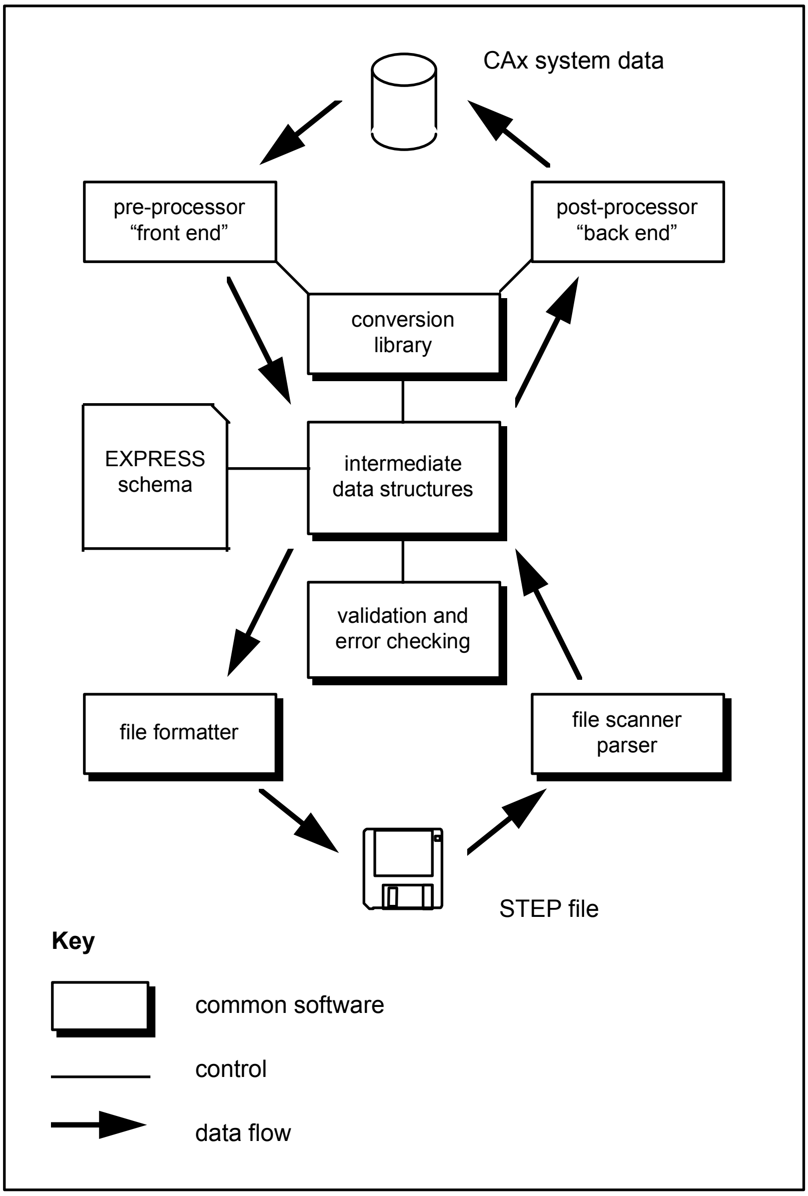

Figure 16: software architecture for STEP file exchange

Data access using STEP

Earlier in this chapter, the distinction between data exchange and data access (as the basis for data sharing) was made. Level 3 implementations of STEP are concerned with the latter, and are supported within STEP by a second standard implementation form: the Standard Data Access Interface, or SDAI (ISO 10303-22). Figure 17 illustrates the use of the SDAI in providing access to shared data.

It will be noted that the "translation" element of this picture, and the roles of the data models, are identical to those depicted in Figure 14 for data exchange. The difference, however, is that rather than encoding the data in a form suitable for file exchange, a standard interface is used that allows the data to be stored in a database. This standard interface (the SDAI) allows applications, such as CAx systems, to store, access and share data.

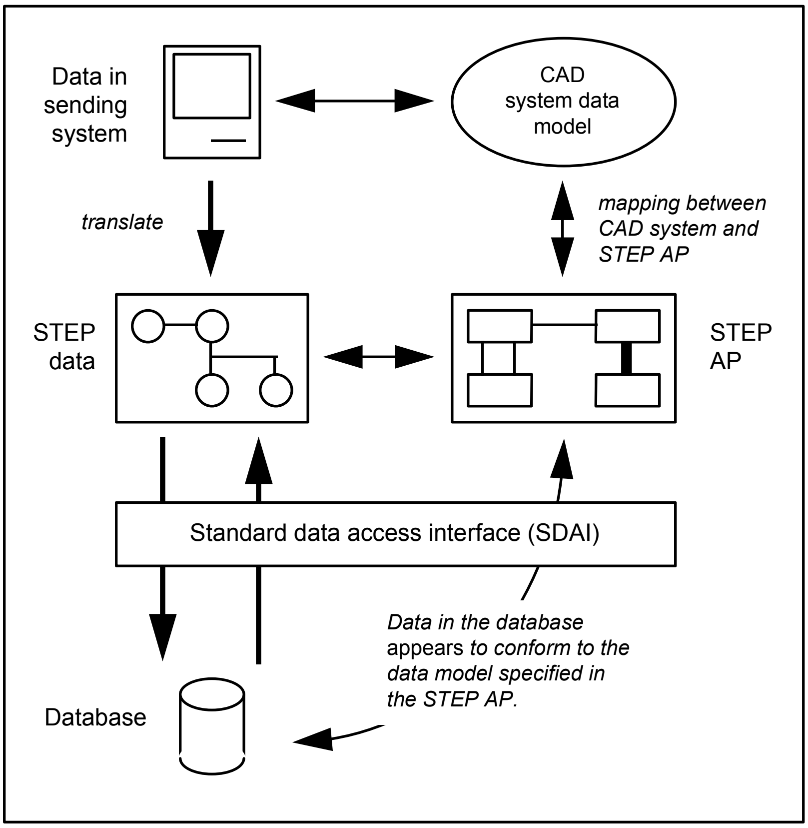

Figure 17: data access using STEP

The purpose of the SDAI is three-fold, in that it provides the following.

- A standard Application Program Interface (API) for data that is defined in EXPRESS. In Chapter 3, the distinction between a conceptual model and a physical model was made with reference to the ANSI/SPARC architecture for database systems; the SDAI provides the facility to access data *as if* the physical model used for data storage is identical to the conceptual model.

- A consistent data access environment for use in software development. If a programmer can assume that data is available via the SDAI, then the details of the underlying database are hidden and not of concern. Similarly, the developer of a database system that provides an SDAI need not worry about the details of each individual application that will use the database.

- Independence from the underlying database technology. Since the interface defined by the SDAI is based on a conceptual data model, different database technologies can be used for data storage without the need to alter the interfaces used between the applications and the database.

The SDAI specification is split into a number of different parts. The core specification (ISO 10303-22) specifies a language-independent access interface. Other parts provide the additional detail that is need to implement the SDAI using specific programming languages such as FORTRAN, C, and C++.

These additional details, known as “language bindings”, allow two different modes of operation for an implementation of the SDAI. An “early binding” implementation is specific to one EXPRESS data model, and will provide access functions only for the constructs defined in that data model. A “late binding” implementation is independent of any specific data model, and allows access to data defined by *any* EXPRESS data model. Late binding implementations are clearly more flexible, but offer lower performance than early binding implementations due to the overheads in maintaining a complete data dictionary.

The example below shows a fragment of ‘C’ language code, based on a late-binding SDAI. The data model is the same as that used for the exchange file example above. Here, however, rather than encoding information about my car in a file, the same information is to be stored in a database.

```
person1 = sdaiCreateInstanceBN(amodel,"person");
sdaiPutAttrBN(person1,"first_name",sdaiSTRING,"Julian");
sdaiPutAttrBN(person1,"last_name",sdaiSTRING,"Fowler");
```

These three function calls create an instance of the entity **person**, and allocate values to the attributes **first_name** and **second_name**.

```
car1 = sdaiCreateInstanceBN (amodel,"car");
sdaiPutAttrBN(car1,"make",sdaiSTRING,"Ford");
sdaiPutAttrBN(car1,"model",sdaiSTRING,"Orion");
sdaiPutAttrBN(car1,"year",sdaiINTEGER,1989);
```

These function calls create an instance of the entity **car**, and allocate values to the attributes **make**, **model**, and **year**.

```
sdaiPuAttrBN(car1,"owner",sdaiINSTANCE,person1);
```

This last function call creates the relationship between the car and the person.

SDAI implementation architecture

Later chapters of this book include some specific architectures based on the use of SDAI and some of the possible advances that are being made through the combination of SDAI with other new technologies. Here, however, the basic architecture of an implementation that uses SDAI is presented. Figure 18 below illustrates the key elements of an SDAI implementation.

Several points can noted from this diagram. Firstly, two data stores are shown, one labelled “database” and one labelled “dictionary”. The former is where actual data is stored, while the latter provides storage for “meta-data”, ie. data about data. In an SDAI implementation, the dictionary will contain information about the data models that are supported, and therefore the interfaces that are available for use by applications.

In the diagram, two different interfaces to applications are shown (labelled ‘A’ and ‘B’). This indicates that the dictionary contains at least two different data models, and can provide access to data to applications that support either ‘A’ or ‘B’. These data models are, of course, those specified in two STEP application protocols, and therefore the system depicted in Figure 18 can store and provide access to data described in both application protocols. As shown by the ‘A’ and ‘B’ type interfaces in the diagram, a CAx system for which a suitable interface exists can “plug in” to the SDAI in order to store or access data.

The process of storing data from a CAx system in a database via the SDAI involves three distinct stages, as shown by the labelled arrows in Figure 18.

1. **Translation**: this is exactly the same process as that described above for file-based exchange. As before, the result of this process will be data stored in "in-memory" data structures within the interface software.

2. **Access**: SDAI function calls are used to create and populate data structures within the underlying database. From the viewpoint of the interface, these data structures are precisely those described by the STEP data model (application interpreted model).

3. **Storage**: the SDAI implementation translates each "standard" function call into the operations necessary to create and populate data structures within the physical database, eg. tables within a relational database, or objects within an object-oriented database. At this point optimization of the physical data storage can be introduced to increase the efficiency of data storage and access.

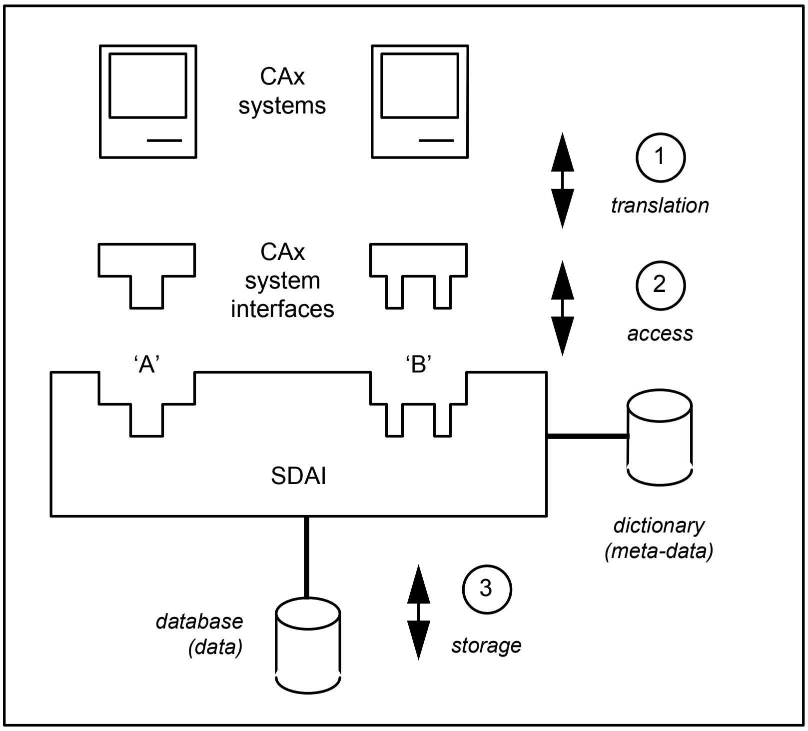

Figure 18: SDAI implementation architecture

Two additional points should be noted.

- The "database" and "dictionary", shown as distinct items in Figure 18, may be just different data sets in the same physical database.

- The SDAI provides no *direct* support for either sharing of data conforming to the same data model between multiple applications, or for sharing of data that conforms to different data models. The scope of the SDAI does not include database management functions such as record locking, control of concurrent access, or functions for multi-schema access to data. Nonetheless, the SDAI provides some of the basic standard elements upon which a data sharing environment can be built.

Conclusions

The separation with STEP of data models from implementation forms has enabled the identification of several different "levels" of implementation. In practical terms, these cover two basic areas of functionality, data *exchange* using files, and data *access* based on a standard interface to shareable databases. The formal specification of the exchange file format permits a greater degree of commonality and automation in the development of interfaces than previous standards such as IGES. The SDAI data access specification opens the door to the development and use of shared product databases.

Advanced prototypes, pilot demonstrations, and commercial products based on both implementation forms have been successfully developed. Examples of these implementations, and the tools that are used to build and maintain them, are provided in Chapters 9, 10 and 11.

---

[1] In this chapter, it is assumed that there is a single data model specified in an Application Protocol. However, as discussed in Chapter 5, there are in fact two data models: the Application Reference Model (ARM) and the Application Interpreted Model (AIM). References in this chapter to *the* data model are to the AIM; Appendix A provides additional information on the roles of the ARM and the AIM with respect to implementation.

[2] ISO 8859 is effectively the 8-bit versions of the character encoding standard known as "ASCII".

[3] In order to reduce the size of STEP files, the mapping of entity names can make use of abbreviated versions. For example, an entity type named **product_definition_relation-**

ship is mapped to the physical file as PRDFRP. These "short names" are standardised for each STEP data model.

[4] This architecture is based on that developed in the "CADEX" project, part of the ESPRIT programme of the European Community. Many other projects and commercial products have adopted similar architectures as the basis for their development.

7 STEP and other standards

STEP does not exist in a vacuum. The total information needs of a business encompass much more than "product data" of the type supported by STEP. Other types of information used within a business include:

- commercial information: orders and invoices;
- production information: resource plans;
- logistics information: maintenance plans and spares management records;
- geographical, geological, and geophysical information: site plans and seismic survey results;
- scientific information and analysis results;
- product and part catalogues;
- financial information;
- documents of all kinds: drawings, data sheets, technical manuals, and reports;
- management information.

Unfortunately, information does not generally fit within such simple categories: most information relates to many different disciplines and users. None of the types of information described above is generally considered to be within the scope of STEP. Each is supported by at least one other standard.

This chapter examines some of these other standards that exist in related areas to STEP, and discusses how they may be used alongside STEP within a corporate information standards strategy. The discussion covers:

- two other standardization efforts within the same ISO sub-committee as STEP;

- standards for other types of information that are used in conjunction with STEP, such as those for electronic commerce (EDI) and electronic publishing;
- the US Department of Defense's "CALS" program and its mandated standards;
- standardization efforts in industries that have chosen not to make use of STEP for product data standards.

Other standards developed by ISO TC184/SC4

Appendix C describes the committee structure of ISO TC184/SC4, and identifies two Working Groups that are not involved with the development of STEP.

- ISO TC184/SC4/WG2 is developing a standard for Parts Libraries.
- ISO TC184/SC4/WG8 is developing standards for manufacturing management data.

Parts libraries

ISO 13584 "Parts library" (sometimes referred to as "P-LIB") defines a common structure for neutral, exchangeable libraries that hold information about "families" or "classes" of parts. Such parts families often refer to a parametric design; for example, ISO standard screws and bolts are specified by reference to tables of values of thread pitch, diameter, thread length, head size, etc. Most manufactured products include some element of standard parts: a typical car will include many components within their electrical, hydraulic, and mechanical systems that are procured "commodity" items, rather than being designed specifically for each make, model, or variant.

The use of such standard parts within a design is commonly supported by the use of parts libraries. These may take the form of paper catalogues or of computer-based systems that allow a designer to select components from a database of all available parts using criteria such as form, fit, function, performance, or price. P-LIB defines standard mechanisms for the creation and use of such parts libraries. Figure 19 below illustrates the fundamental concepts of P-LIB.

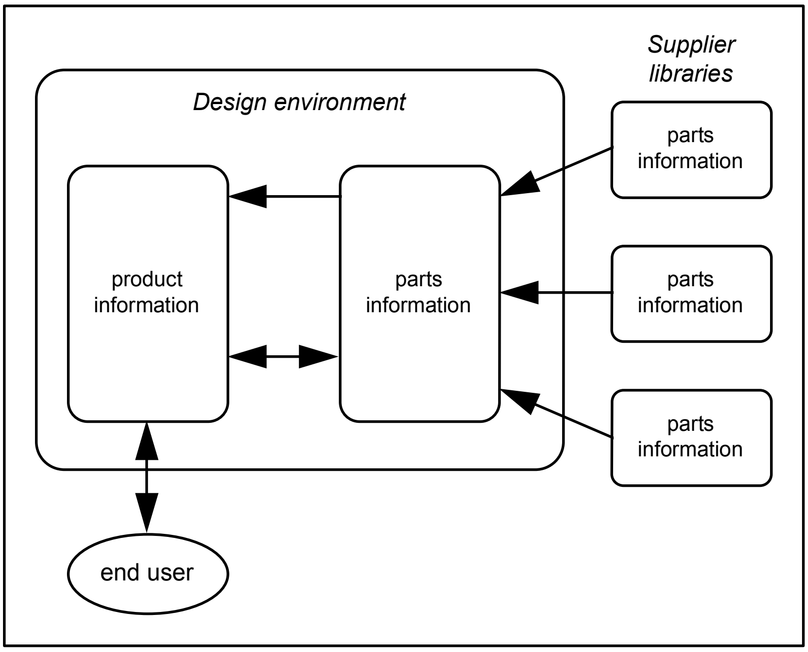

Figure 19: P-LIB overview

Here, an end user works in a design environment that provides CAD and other tools that support the design process, including a "user library" system that contains information about all the "standard" parts that are available to the designer. This user library is created using supplier libraries, ie. libraries created by the manufacturers or suppliers of the standard parts.

A User Library contains two basic types of information:

- information that allows the identification and selection of standard parts;

- information that describes the properties or characteristics of the standard parts.

These two types of information correspond to the two arrows between "CAD" and "User library" in Figure 19. The lower arrow represents the use of the library system, through dialogues and queries to search for and select parts that meet criteria determined by the user. The upper arrow represents the transmission of property information from the library to the CAD system. For example, once an architect has selected a specific type of window frame from the parts library, a drawing of the window (representing its shape and dimensions) is transmitted to the CAD system so that it can included within the design of the building.

Within the environment depicted in Figure 19, P-LIB specifies the following:

- a neutral interface between the CAD System and the User Library;
- a method for defining the structure of the data to be held in the User Library;
- a standard identification system for parts;
- a neutral interface between the User Library and Supplier Libraries.

The parts of the ISO 13584 standard are listed in Appendix C.

Relationship to STEP

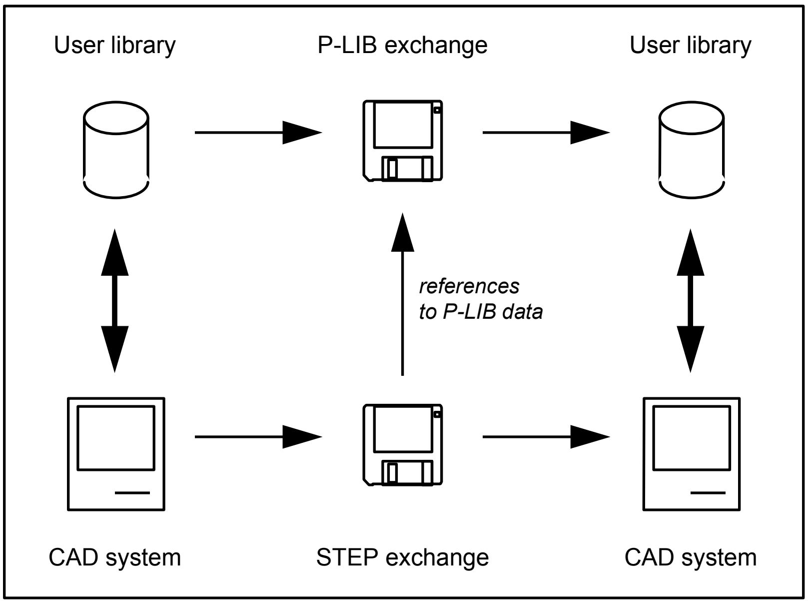

Figure 20: co-operative use of STEP and P-LIB (1)

The Parts Libraries standard has a very close relationship to STEP. It is a common industry requirement that product data (STEP) and parts library data (P-LIB) should be exchange, shared, and managed together. Figure 20 and Figure 21 illustrate two different scenarios for the co-operative use of STEP and P-LIB.[1]

In Figure 20 two organizations are working on a common project. Each has a different CAD system and a different catalogue system. The P-LIB standard allows the exchange of the parts library that is to be used throughout the design project, eg. all the types of doors and windows to be used in a building. Each organization works on the design, making use of this common parts library. Successive versions of the design are then exchanged using an appropriate STEP Application Protocol; the STEP files include references to any standard parts included in the design.

Such co-operative use of the two standards acts as an enabler to increased design quality by ensuring that designers are making use of a common, consistent set of standard parts. It also increases the efficiency of the data exchange processes, since each model or drawing of the building includes just *references* to the standard parts included in the design; this has significant potential impact on the size of the exchange files and on the time taken to process, translate and validate each file.

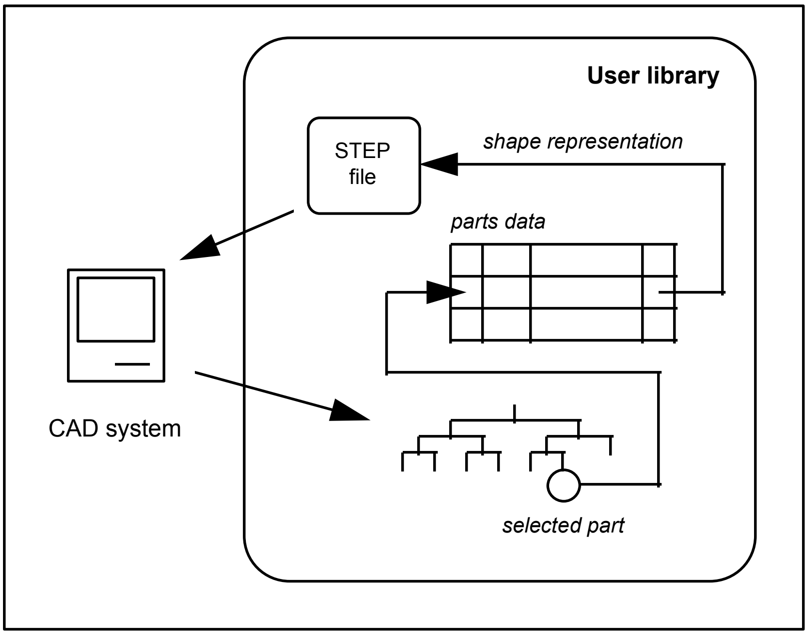

Figure 21: co-operative use of STEP and P-LIB (2)

In Figure 21 STEP is used *internally* within the User Library as a format for the representation of the properties of parts. The designer uses the library system to select a part from within the parts library hierarchy. Each part in the library has an number of associated data items (shown in tabular form in the diagram); one of these items denotes the three dimensional shape of the selected part. This then includes a reference to a STEP file, stored within the library system, that represents this shape. In some cases the library system may include the capability to *generate* this STEP file based on the parameter values stored for the part.

This STEP file is then available for use within the CAD system. Although P-LIB does not mandate the use of STEP in this manner, it will be seen that this offers considerable advantages. The CAD system need not be provided with a specific translator for shape information stored in the library; rather, a "general purpose" STEP translator (conforming to an appropriate Application Protocol) may be used instead.

Status of the P-LIB standard

The development of the P-LIB standard was initiated within ISO in 1991; preliminary work was undertaken in a European standards group with the CEN organization. An initial group of seven parts of ISO 13584 were released for Committee Draft ballot in 1995. At this point in the development of P-LIB support for the "co-operative use" of STEP as described above is not complete; it is nonetheless a high priority for the developers of both standards.

Manufacturing management data

The third area of work within ISO TC184/SC4 deals with data that is used in the management of industrial manufacturing plants. The work is undertaken within ISO TC184/SC4/WG8 "MANDATE"; this working group has responsibility for three projects:

- development of a standard for the exchange of manufacturing management data;
- development of a standard for manufacturing resource databases;
- development of a standard for data that relates to the flow of materials within a manufacturing enterprise.

MANDATE has considerable overlap with work on standards in the EDI field, and is making use of the Basic Semantic Repository (BSR) concept developed jointly by ISO and the United Nations. The work on MANDATE was initiated in 1993; at the time of writing, none of the projects within WG8 has yet produced a draft standard for wider review.

STEP and electronic commerce: EDIFACT and ANSI X.12

Electronic data interchange (EDI) has become widespread for the exchange of commercial and financial information such as orders, invoices, and cash transfers. EDI uses "messages": standard formats for each type of information to be interchanged. There are two major standards for EDI: EDIFACT, developed by the United Nations, and ANSI X.12, an America National Standard.[2]

UN/EDIFACT defines a file format for messages, together with syntax for messages, data segments, and data elements. The actual messages are developed by various industry groupings. Two factors have led to a growing awareness of the inter-relationship between EDI and STEP, and of the need for the two standards to be used together.

The EDI messages used in finance and commerce are relatively simple in terms of their structure and content: many are essentially electronic versions of the paper forms previously used to interchange information between organizations. The ever-increasing rate of change of business processes has, however, led to an understanding that encoding of paper forms can lead to restrictions on the ability of enterprises to form new relationships, and new types of relationships.

An initiative within UN/EDIFACT, called Business and Information Modelling (BIM), is taking a more formal, structured approach to the development of EDIFACT messages. The disciplines of modelling used within BIM are very similar to those used in STEP; indeed, the use of EXPRESS as a language for modelling of EDI requirements and EDIFACT messages has been considered. The BIM approach has already been adopted within a number of the industry sectors working within the UN/EDIFACT framework, including health care, finance, and construction.

The second factor that is influencing the need for synergy between EDI and STEP is a growing requirement within industry to be able to include technical data within EDI messages. For example, a simple EDI message can be used to place an order for a number of components, based on a reference to the component from a catalogue or other "external" information source. However, the disciplines of "lean" or "just-in-time" manufacturing can give rise to a need to include complete product information within the order. For example, it is advantageous to a petro-chemical plant engineering contractor to be able to procure a pressure vessel from a specialist manufacturer, using a single message or transaction that includes the technical specifications of the vessel as well as the relevant purchasing information.

This requirement has been characterized as "multi-format exchange" (MFE), ie. the inclusion within an EDI message of additional data in other formats. The ANSI X.12 standard includes such a capability: it allows the inclusion of "binary data" within a transaction. This binary data could be a CAD model in native format, an IGES file, or a STEP file. However, this capability in ANSI X.12 is limited, since there is no indication within the EDI transaction of the content or intent of the binary data.

Some initial prototypes of a hybrid STEP/EDIFACT capability have been developed. For example, the ESPRIT "PISA" project has demonstrated the ability to exchange messages in an extended EDIFACT format that combines "commercial" data in EDIFACT syntax, and product data in STEP physical file format.

The key requirement to be fulfilled, however, is the recognition and management of the potential redundancies and ambiguities within such "hybrid" messages. STEP's focus on *complete* product data means that every STEP Application Protocol embodies the concept of product identification. Many also include "management" data such as the identification of organizations, approvals, and security classifications. This information is likely to be present in EDI messages as well. An effective solution to the co-operative use of EDI and STEP will therefore have to address how such information is managed: this is a major challenge for both standards groups in the future.

STEP and electronic publishing: SGML

The Standard Generalized Mark-up Language (ISO 8879) specifies a language for document representation, based on system-independent mark-up and encoding. It is widely used in electronic publishing, particularly in the defence industry (see section on CALS below). SGML can be used to represent any type of document, but is especially suited to the development, publication and distribution of technical documents such as manuals and data sheets.

This usage of SGML introduces a strong relationship to STEP. STEP includes the concept of documents as carriers of additional information related to products. For example, a STEP file may include references to documents that contain material specifications, operating conditions, or contractual information. Without a link to SGML (or another suitable documentation standard) these references are by name only; a link from STEP to SGML would enable references to specific chapters or sections *within* a document. A link between STEP and SGML also permits the management of documents as "information products" within a wider product data management context, eg. the management of installation guides and technical manuals *as parts of* a computer system.

The capability of SGML to structure textual information through the assignment of mark-up "tags" also offers a capability to manage such information more effectively within STEP. For example, SGML tagging of textual information presented in an engineering drawing (represented within STEP), would allow the same information to be re-used within the technical documentation (represented within SGML) for the same product.

Work on the inter-relationship between STEP and SGML has been undertaken by the "SWEDCALS" project in Sweden.[3] There is also strong interest in this work within other groups, and the development of solutions for the use of STEP and SGML as part of a single, standards-based information management strategy is a high-priority activity for STEP in the mid 1990s.

STEP and CALS

STEP has for many years been seen as a key standard within the US Department of Defense's CALS program.[4] The pragmatic nature of CALS has resulted in a strong focus on the use of those standards within common use in industry; therefore, established standards for the exchange of technical information (IGES, SGML) have been adopted. As STEP moves towards maturity, however, it is likely that the use of STEP Application Protocols will eventually supplant the subsets of IGES currently specified in military specification MIL-D-28000. The Department of Defense has adopted a policy of direct use of national and international standards, rather than the development and publication of military specifications. This will allow more rapid uptake of STEP within the CALS environment.

Several STEP Application Protocols reflect requirements from the US defence industry and therefore embody a number of CALS concepts for data management and electronic commerce. An Application Protocol for Technical Data Packs has been proposed, that will support the information that is typically delivered by a customer to a contractor, or from a prime contractor to a sub-contractor, for the purposes of procurement.[5] The proposal identifies this information as:

> *"... (the) engineering definition sufficiently complete to enable a competent manufacturer to produce and maintain quality control ... to the degree that the physical and performance characteristics interchangeable with those of the original design are obtained without resorting to additional product de-*

sign effort, additional design data, or recourse to the original design activity."

Such data encompasses drawings, documents, bills of materials, associated lists, and product data sets (including CAD models). The scope of this proposal may be compared with that of MIL-STD-1840, which plays a similar "data packaging" role for the original CALS standards. It is therefore likely that the Technical Data Packs Application Protocol represents a migration path from the current CALS standards to a future basis in STEP. The inclusion of technical documents within the scope of the proposal also introduces a clear overlap with work on SGML, as discussed above.

STEP also fits within CALS technical framework, as a key enabler of the concept of shared data that is created once, used many times, and resides in many places. This technical framework forms one of the key drivers to the inter-relationships between STEP and other standards: it is the articulation, by a major customer organization, of the need for integrated information management across the discipline and technology boundaries between standards. Since 1990, "CALS" has become less a defence initiative, and more a standard-bearer for international, government-industry activities aimed at increasing efficiency and competitiveness through electronic commerce. Thus, although the "peace dividend" may have reduced the visibility of CALS as a primary driver of STEP development, it still represents a vital source of requirements for the future development of the standard.

The importance of STEP to CALS is demonstrated by the conclusions of a key CALS study.[6]

"… the integrated data environment envisioned by CALS cannot be achieved without a product data definition standard such as STEP. Without STEP, CALS will remain in Phase I. STEP … is critical to the manufacturing element in the CALS environment. There must be appropriate and correct implementations (application protocols) of STEP available for the specific requirements of the environment."

Design of electrical & electronic systems

In Chapter 2, EDIF (the Electronic Design Interchange Format) was mentioned as one of the current standards for product data exchange. The development of standards in the electronic and electrical domain as a whole is one that introduces requirements for synergy between STEP and other, established standards. The following is an incomplete list of the standards that are available or in development for data exchange in the electrical and electronic domain.

- AP210 (STEP) supports the exchange of design, manufacture and assembly information for printed circuit boards.

- AP211 (STEP) supports the exchange of diagnostics and remanufacturing data for electronic systems.

- AP220 (STEP) supports the exchange of manufacturing planning information for printed circuit boards.

- EDIF is an EIA standard that supports the exchange of design information for printed circuit boards and assemblies.

- VHDL (VHSIC Hardware Design Language) is an IEEE standard that supports the exchange of design and simulation information for integrated circuits.

- IPC D350 is a standard developed by the Institute for Interconnecting and Packaging Electronic Circuits that supports the exchange of information used in the manufacture of printed circuit boards.

- AP212 (STEP) supports the exchange of design and installation information for electrical systems.

- VNS is a German standard for the exchange of information related to electrical systems and installations in the power generation and distribution industry.

Figure 22 illustrates the overlaps between some of the standards from this list that are used in the field of circuit board design and manufacture.

The key point to be drawn from this diagram is not that STEP is competing with existing standards, but that it has the potential to act as an *integrating* standard across others. Thus, EDIF, VHDL and IPC D350 do not themselves have obvious or significant overlaps: STEP AP210, however, includes elements from the scope of all three. The challenge to the standards bodies here is to develop solutions that allow information to be exchanged using the existing standards, to be exchanged across these domains (using STEP), *and* to be integrated in a design database environment that handles all the data shown in Figure 22.

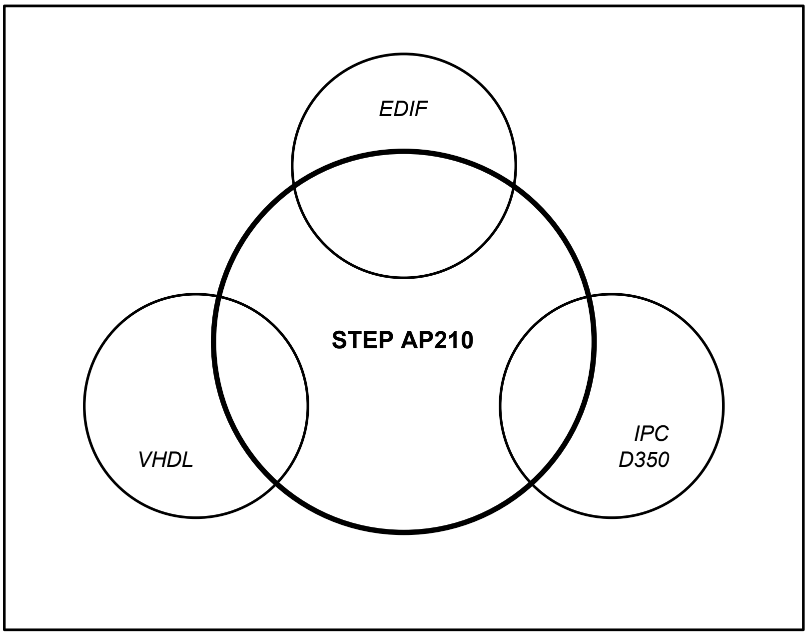

Figure 22: overlaps between electrical/electronic standards

This challenge is recognized within the STEP committee structure: the capabilities of STEP for electrical and electronic applications are developed by a *joint* working group between STEP (ISO TC184/SC4) and IEC TC93 ("Design automation"), the committee responsible for the development and international standardization of both EDIF and VHDL. Primary attention has been given to the harmonization of STEP AP210 with EDIF: this is made easier by the fact that EDIF now makes use of EXPRESS as its data specification language.

POSC: Standards for oil and gas exploration & production

The Petrotechnical Open Software Corporation (POSC) is a consortium of major oil companies and others involved in the exploration and production (E&P) sector of the oil and gas industry. POSC was formed in 1990 as a non-profit, vendor neutral organization to address issues related to the effective use of information technology in the E&P sector. POSC differs from STEP not only in its technical scope, but also in the fact that it has chosen *not* to make use of the ISO standardization process; rather, through consensus building by the major industry companies, and the involvement of major information technology vendors, POSC seeks to establish *de facto*, industry standards.

The POSC software integration platform

There are many similarities between the STEP standards and the POSC specifications. Both include standard data models, file formats, and data access specifications. Both use EXPRESS as a standard language for defining data models. There are also differences, particularly in scope. STEP is intended to support the product data exchange requirements of all relevant industry sectors; within this scope, data models are defined that capture consistently the requirements of different classes of applications.

POSC, however, is restricted to a single industry sector – oil and gas exploration and production (E&P) – but addresses a greater “vertical” scope within that sector, covering other technical computing needs such as user interface and base technology standards. Together, these comprise a Software Integration Platform (SIP), made up of the following components:

- base computing standards;
- the EPICENTRE data model;
- data access and exchange specifications;
- exchange format;
- E&P user interface style guide.

The EPICENTRE data model has been developed by POSC in order to provide a conceptual, detailed logical data model. The scope of EPICENTRE is primarily that of geological and geophysical data, and it is intended as a common reference model for the integration of E&P applications using shared databases and data exchange. The EPICENTRE model has been developed using an extended version of EXPRESS. Although the methodology used in the development of EPICENTRE differs from that of STEP, the resulting data models are nonetheless comparable, and EXPRESS-based tools can be used in both environments.

The POSC specifications support two forms of implementation:

- data access, based on an Application Program Interface (API);
- data exchange, based on the POSC Exchange Format (PEF).

POSC may offer to STEP additional concepts and ideas in the field of implementation. The POSC exchange format is a binary format that may enable more efficient processing of large data files than the STEP physical file (which uses an ASCII encoding). The POSC API is similar to the STEP standard data access interface (SDAI); experience of implementations of the API within the POSC Industry Pilot Projects (IPP) may provide valuable feedback in the refinement of the SDAI as it completes the standardization process.

STEP and POSC compared

There is one clear overlap in the scopes of STEP and POSC: the engineering equipment associated with oil wells, rigs, and platforms. This is clearly within the scope of POSC (and is partially covered in version 1 of the EPICENTRE model), and is also in the scope of STEP. STEP Application Protocols currently being developed in the areas of process plant, shipbuilding, and civil engineering all include elements with clear relevance to POSC. This area is being investigated by a joint project between POSC and the Norwegian offshore industry.

This overlap is of more than academic importance to the major oil companies, who not only seek, develop and exploit sources of oil and gas, but also engage in refining, processing and distribution of petrochemical products. It is difficult to see the benefits to these companies if POSC and STEP independently, and inconsistently, develop standards for data associated with E&P engineering functions.

STEP and POSC each represent the results of many hundreds of man-years of research, development and standardization efforts. Individually, they provide solutions to specific problems. Together, they can provide a single, consistent set of solutions to the data exchange, sharing and management issues faced by the process and petrochemical industries.

Conclusions

STEP is not – indeed cannot be – an isolated standard. The customers for STEP are also customers for standards in many other areas of information technology. This chapter has explored this situation, using examples drawn from a number of different industries, technologies, and standards. The common theme between these examples has been that there is a real, strategically important industry need for STEP and other standards to work together, and that this need has not yet been fully fulfilled. Much good work has been initiated, and positive results achieved; however, just as the “initial release” of the standard faced the challenge of establishing STEP as an effective enhancement to standards such as IGES, SET, and VDA-FS, future work must demonstrate that STEP can become just one piece in a larger “jigsaw” of standards-based information integration strategies.

---

[1] The term “co-operative use of standards”, denoting the consistent usage of several different standards to meet a single industry goal, was introduced into the work on STEP by Bernd Wenzel (EuroSTEP GmbH, Germany).

[2] Within the ANSI X.12 standard, the term “transaction set” is used as a synonym for “message”.

[3] SWEDCALS is a Swedish joint effort to explore the potential benefits of CALS and electronic commerce to Swedish industry and government agencies.

[4] Continuous Acquisition and Logistic Support. For an introduction to CALS, see Joan M. Smith, *An introduction to CALS: the strategy and the standards*.

[5] *Proposed Application Protocol Summary Sheet: Technical Data Packaging Core Information and Exchange*. Document ISO TC184/SC4/WG3/N430, June 1995.

[6] Robert Willis, *Implementing the Vision – A Case Study Embracing CALS Technologies and Philosophies.*

8 The potential benefits of STEP

In this chapter the benefits of STEP are examined.[1] Previous chapters have presented the background to the development of the standard, its structure and content, and the technologies that are available to implement it within computer systems. However, the mere existence of STEP and the ability to create software that conforms to it does not of itself bring *any* benefit. Rather, STEP needs to be related to the *business* context in which it is used, and to the business problems that it can help to solve and the business opportunities that it creates.

The potential benefits of STEP can vary greatly: in some cases, STEP provides an incremental improvement in computing technology, whilst elsewhere it may be a key enabler for competitive advantage. The benefits gained from STEP by an organization depend in part on the goals that the organization sets for STEP. Some of the potential goals for the implementation and use of STEP relate to improvements to current techniques:

- improve the quality, accuracy and completeness of the exchange of data between CAD systems;

- increase the re-use of design information in engineering, manufacturing, operations and support functions;

- combine CAD models and drawings with other data in a managed product data environment.

Others relate to new or improved business processes:

- share complete product data with customers, collaborators, suppliers, and sub-contractors;

- implement concurrent engineering;

- create a shared product database that is used by many different disciplines and applications.

Each of these goals is valid; indeed, there are many organizations world-wide who are already committed to achieving one or more of these by using STEP.

Another factor in assessing the potential benefits of STEP is the current "state of the art" in data management. Table 4 below illustrates some of the changes that STEP enables.

	Current situation	Opportunity offered by STEP
Availability of data	Stored in heterogeneous, proprietary formats	Freely exchangeable between different systems
Accessibility of data	Restricted by quality and costs of translation processes	Accessible through standard interfaces
Re-usability of data	Substantial loss and/or recreation of data between life cycle phases and across enterprise boundaries	Enables the creation and maintenance of shared data environments
Quality of data	Reduced by inaccuracy, incompleteness, ambiguity and redundancy of data	Enhanced by the use of standard data models and interfaces

Table 4: opportunities created by STEP

The third factor that influences the potential benefits of STEP is the changing cultures of business. Today, few products are designed, manufactured and maintained by a single organization. Most businesses therefore operate in an environment of parallel collaboration *and* competition with other similar companies. Within this environment, the ability to manage and exploit information becomes critical to the development of competitive advantage. STEP's role as an enabling standard focuses on:

- better availability and accessibility of information;
- more effective and efficient use of information;
- the deployment of systems that simplify and improve the processes within organizations and the relationships between organizations.

Where do standards fit in an enterprise?

In examining the benefits of a standard such as STEP, it is useful to understand the role that technology and standards play within an enterprise. One view of this role is illustrated in Figure 23.[2]

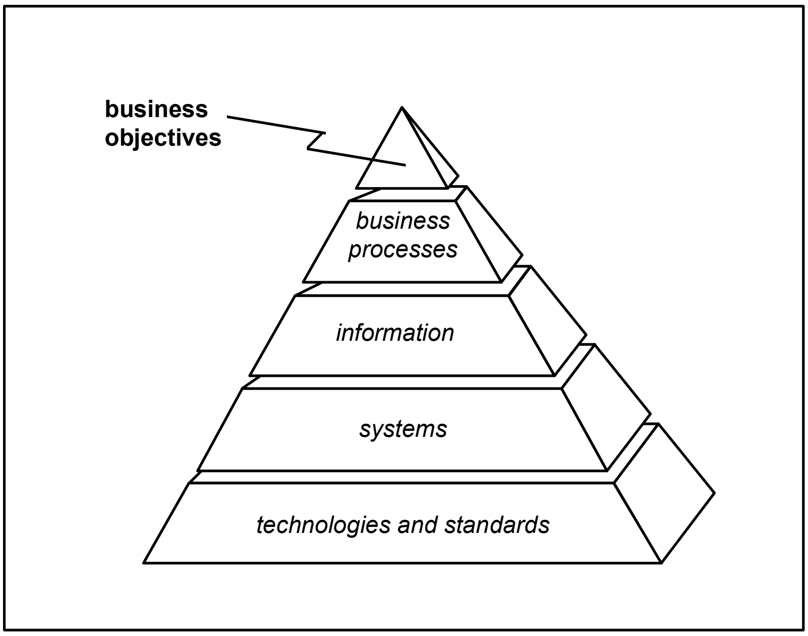

Figure 23: the role of standards

This diagram depicts various "layers" within an overall information architecture for a company. The apex of the pyramid is the objectives of the enterprise; examples of such objectives are:

- increased market share;
- improved profitability, earnings per share;
- agility in response to customer needs;
- military readiness.

These objectives are fulfilled by the processes that the enterprise executes; these processes are underpinned by specific activities, including decision making, and the information that is created or used in these activities. These activities are, in turn, supported by systems and information processes. It is only within the infrastructure of technologies and standards upon which such systems are built that we encounter STEP.

Therefore, we can see that the benefits of STEP have to be analysed in terms of each of the successive layers above it to see the eventual impact on business objectives. STEP of itself has *no* such impact: rather, the long-term benefits from STEP result from the improvements to business processes which allow decisions to be made on the basis of accurate, timely information. The role of STEP is the creation, management and sharing of such information in a form that is adaptable to different systems and enterprise needs.

Increased benefits from information technology

A major area of benefit to be gained from the effective implementation and use of STEP is through better use of information technology to meet business objectives.

In increasingly competitive markets, delays in responding to customers' needs can be a major factor in lost business. The use of STEP can improve this responsiveness in several ways:

- by providing effective interfaces between systems, a supplier can receive and deliver technical information related to tenders, orders and contracts more effectively;

- by integrating product data in a shared environment, the impact of design or manufacturing changes on costs, production and delivery schedules can be assessed and communicated to the customer more rapidly.

The time taken to deliver a new product to the market can be reduced through the effective use of STEP. This reduction may be achieved through:

- improved interfaces and integration between systems, enabling support for concurrent engineering teams;

- system-independent access to previous designs, encouraging re-use of existing information;

- improved interfaces with suppliers and sub-contractors, enabling incorporation of appropriate procured parts and components within a design.

The quality of products, across their entire life cycle, may be improved by the use of STEP. This quality improvement may be achieved by:

- fewer errors and design defects resulting from incomplete or inaccurate information;

- continuous improvement as a result of data feedback: data produced in manufacturing or operations can be made available in a standard format to the design and engineering functions;

- improved interfaces between systems, reducing or eliminating the amount of information lost in exchanges, and therefore the potential for errors or inaccuracies in identifying and recreating missing information;

- improved integrity of information across the life cycle of the product, through the implementation of shared product data environments.

One of the most significant benefits that can result from the implementation and use of STEP is the removal of restrictions on business practices. The new ways of working that may be enabled by STEP include:

- greater choice in selection of computing systems, based on functionality and business needs rather than compatibility with existing systems;

- greater choice of suppliers and sub-contractors, based on capability and business fit, rather than the use of common systems;

- more effective re-use of information by project partners, based on the availability of standard interfaces and exchange capabilities between systems;

- automation of quality control, safety and environmental checks: computer systems to undertake these tasks may be developed and implemented, based on the use of STEP to provide standard data models and interfaces, and therefore access to all necessary information.

STEP can create new business opportunities in several ways. The ability to create new business partnerships with information technology vendors and with suppliers or sub-contractors is discussed above. Other opportunities created by STEP include:

- inclusion of suppliers and sub-contractors in the “virtual enterprise”, enabled by STEP’s capability for exchange and sharing of product information across enterprise boundaries;

- increase the use of specialist suppliers of skills and services, again based on the shareability of product information;
- "24-hour" working across time-zones: STEP can be used as part of an information infrastructure that ensures that engineers working in London, San Francisco, Tokyo, or Sydney are able to work co-operatively on one project, sharing and updating product information in real time.

Driving out costs

The second major area in which STEP delivers benefits to industry is by reducing the costs associated with key processes.

Analysis of the processes of design and engineering reveals many activities that are undertaken that do not contribute to the value of the final product. Such activities include:

- looking for data;
- translating data between different formats;
- recreation of data lost in translation;
- duplication of data by different disciplines.

STEP is a powerful enabler to the reduction of these activities, through the capability to exchange and share information between systems and disciplines. This in turn results in the elimination of duplicate or redundant activities, and potential reductions in project effort and staffing costs.

Many companies have made significant investments in information technology, not just in terms of hardware and software, but also the training of the staff who use these systems. It is not unusual that such investment requirements are increased by the acquisition of multiple systems offering the same functionality, in order to be able to meet compatibility requirements of different customers. This diversity is an overhead that is ultimately passed on to the customer. Through STEP, a company can retain and invest in the systems that meet business needs, without compromise in performance or functionality.

The use of STEP to enable seamless exchange and sharing of data means that companies need not be tied, long-term, to proprietary hardware or software, and can select "best in class" specialist applications without significant integration overheads. This also applies to the integration of "home gown" applications, as well as to integration needs that may arise from organizational changes.

Measuring the benefits

The discussion above demonstrates that the potential benefits of STEP vary greatly, and result from the achievement of many different business objectives. It is therefore impossible to identify a single mechanism that measures the benefits of STEP. Indeed, the dependence of "STEP benefits" on the higher level business objectives means that they cannot be measured independently.

For example, suppose a manufacturing company uses a CAD system to receive design data from a customer, to add further detail to the design, and return the results to the customer. The current method of doing this is to use an IGES translator, whose quality is such that every file received from or sent to the customer has to be checked and possibly corrected. This checking takes 30 minutes per file, and the average number of exchanges per week is 20. If the cost of this checking is £25 per hour, then the weekly overhead cost associated with the data exchange is £250. Suppose a STEP interface becomes available and, after testing and acceptance, it is agreed between the company and its customer that the quality of the exchange has improved to the extent that it can be maintained through sampling, and that only one in twenty exchange files need to be checked in detail.

This suggests that the benefit of STEP to the manufacturing company is a relatively modest £225 per week. However, if, as a result of the improved information exchange between the company and its customer, the relationship changes such that the company is given complete design responsibility for these parts, then the benefit may be measured in many thousands of pounds a week.

This example demonstrates how cost-benefit analysis, and the metrics used in the analysis, should be determined by the strategic goals of the organization. Such a "top-down" analysis will help to identify the activities and processes in which there are opportunities to apply STEP and achieve one or more of the benefits described above. Once these activities have been identified, it is possible to assess the quality, cost and benefit metrics of current ways of working, and therefore to estimate the improvements resulting from STEP. The top-down nature of the analysis means that the impact of "low level" benefits, such as error-free data exchange, can be related to higher level benefits such as new business opportunities or reduced project costs.

There are very few published case studies on the actual benefits of STEP achieved in practice. This is not unexpected since, as will be seen in the next chapter, STEP is only just entering routine, production use in a small number of industry sectors. This is not, however, the only reason for the relative lack of quantitative information in this area. In assessing the potential benefits of STEP, many companies have analysed the actual costs associated with data exchange and data management for the first time. The results of these analyses have frequently shown that these costs are considerably higher than had previously been estimated; this clearly means that the potential benefits of STEP are increased, but also means that these companies are less willing to reveal the true costs that have been identified.

Benefits to systems developers and vendors

The discussions above have concentrated on the potential benefits of STEP to users of information technology, and some of these benefits have been shown as resulting from increased return on investment, or lower initial investment, in information technology. Does this mean that STEP does not offer benefit to system developers and vendors?

In fact, STEP offers potential benefits to the information technology industry as significant as those for the end-user. Some of the longer-term aspects of these benefits are described in Chapter 12. STEP also offers short and medium term benefits. As is shown in Chapter 10, there is wide availability of software tools to assist a developer in the development of STEP interfaces; the costs of developing such interfaces is therefore no more than that for IGES or other translators.

STEP does create both threats and opportunities for systems vendors. By removing data barriers between systems, users are much less likely to continue acquiring systems from a single vendor purely on the basis of data compatibility. This therefore creates opportunities for vendors to win business from their competitors, and equally means that vendors may have to work harder to retain their customer base. However, this analysis does not account for the significant opportunity for vendors to use STEP in creating closer, long-term relationships with customers, working as a partnership to develop and implement STEP-based information technology strategies.

There are also a number of technical benefits offered to systems developers by STEP. Firstly, STEP represents a massive, public domain collection of knowledge of both CAD technology and end-users' requirements. Vendors who ignore the potential offered by this information source do so at their peril! Similarly, systems integrator can make use of STEP as reference models within integration projects.

Secondly, the systems architectures enabled by STEP (as discussed in Chapter 12) allow application developers to concentrate on the functionality of their products, relying on the existence of STEP's standard data models and interfaces to provide for data storage and management functions.

The final technical benefit of STEP to system vendors applies to those producing specialist applications, particularly those in the engineering analysis or manufacturing domain. Such systems often make use of the geometric information created in CAD systems, and this may result in the need to create and maintain interfaces to *each* CAD system with which the analysis or manufacturing tool is required to work. STEP's capability to deliver high quality interfaces between tools should mean that such developers need only implement an appropriate STEP interface in order to allow users to couple the tool with any CAD system.

Barriers to the implementation of STEP

This chapter has concentrated entirely on the potential benefits of STEP, and may have implied that there are no reasons for *not* implementing STEP. This is, of course, not the case. As with any new standard or technology there are many potential barriers to the implementation and market take-up of STEP. Some of these barriers result from myths or misunderstandings of the nature of STEP and the benefits that can be achieved from its use.

- *"My system vendor will supply STEP to me".* The earlier parts of this chapter have demonstrated that STEP is not just a new piece of software added to a CAD/CAM system, and that no benefit results from STEP unless it fulfils a business need.

- *"STEP will reduce competitive advantage."* There is a common misunderstanding that STEP will standardize *systems*, ie. that all CAD/CAM systems will become the same, and users will gain no benefit from the use of one system over another. In fact, the reverse is true: the existence of STEP as a standard for *data* will result in diversification of systems differentiated by the functions and performance provided. Since users will no longer have to consider compatibility as a major factor in system selection, systems can be chosen that reflect and enhance the business processes through which advantage is gained.

- *"STEP will solve all data management problems".* As much as this might be desirable, STEP is an enabler of improved data management, providing *part* of the framework around which a data management strategy can be created, and providing for the integration of systems and data around this framework.

- *"STEP will make confidential information publicly available".* STEP does not enforce specific business processes or practices, or remove the responsibility on companies to retain control of confidential information. Indeed, the requirements fulfilled by many STEP application protocols include the *maintenance* of data security.

Other barriers to the implementation may be characterized under two headings: barriers that may prevent or delay vendor implementation of the standard, and those that may prevent its adoption in industry.

Until the publication of the initial release and the success of a number of high-profile industry pilot projects, a major barrier to the implementation of STEP was the perceived lack of real customer demand. Interest in STEP was high in the standards development community, in research projects and in academia, but few potential industry users were demanding of their information technology suppliers firm commitments to implement STEP and make it available as commercial products.

Since 1993-94, however, this barrier has largely disappeared. As discussed in Chapters 9 and 10, not only have a number of major projects successfully demonstrated the use of STEP in production environments, but most major CAD vendors have announced STEP interface products. Nonetheless, the fact that these pilots and products are focused on a small number of application protocols (AP203, AP214) and their use in two industry sectors (aerospace and automotive) means that barriers of this kind are still very real in other areas.

These are linked to the other major barrier to vendor implementation: the perception that, despite ten years development, STEP is neither complete nor mature, and that considerable investment on the part of systems developers is required to turn the standard into usable products. Whilst there is an undoubted element of truth in this perception, it stems primarily from a lack of understanding of the role of STEP: systems developers who see STEP *only* as a CAD exchange standard, designed to replace IGES, are likely to construe the complexity necessary in STEP to manage life cycle product data as an unnecessary overhead. Such misunderstanding can be addressed only through a combination of education, awareness, and participation in pilot projects.

The largest single barrier to STEP adoption and use in industry is a lack of knowledge of STEP and its associated technologies and capabilities. STEP, as a set of standards documents, is a daunting prospect to a potential user. As mentioned above, users expecting STEP to be a direct replacement for IGES or SET may perceive the new standard as unnecessarily complex.

This barrier is slowly being overcome by the growing realization of the potential power of STEP as an enabling technology for concurrent engineering and process improvement. This higher profile for STEP is leading to a much greater degree of company commitment to development and use of the standard, and is typified by the senior management presence on the boards and steering committees of many of the projects discussed in the next chapter.

Summary

STEP offers many potential benefits to both users and developers of CAx systems. Realization of the benefits of STEP depends in the greater part on the understanding that STEP is much more than a new and better mechanism for CAD/CAM data exchange. The linkage between STEP as an enabler for improved data management, and the benefits to be gained in terms of improved business processes and fulfilment of business objectives, places STEP in a context that requires the attention and support of senior management.

---

[1] The material in this chapter makes use of a number of sources, some unpublished, that cover different aspects of the benefits of STEP, and their analysis and measurement. One of the earliest studies undertaken in this area (on behalf of the UK "PI-STEP" project) revealed a remarkably small amount of publicly-available information on the benefits of STEP and other data management or exchange technologies. As the number of companies and projects implementing STEP increases, and more hard information becomes available, then more detailed analyses of the benefits of STEP will become available.

[2] This diagram is inspired by several similar pictures included in reports or presentations from a number of sources.

9 STEP projects and pilots

The size and complexity of STEP means that few single organizations can countenance the costs of developing, validating and implementing any component of the standard in isolation. It is not surprising, therefore, that most significant contributions to the development of STEP have come from collaborative projects. Such collaborations have several distinct advantages:

- the costs, and therefore the risks, of developing the standard can be shared between the collaborators;

- consortia can more easily achieve the “critical mass” of technical knowledge and expertise needed to develop STEP capabilities, either from internal resources or by contracting specialist consultants;

- the proposed standard reflects the requirements of a group of companies;

- existing business partnerships within the consortium allow the identification of exchange scenarios for detailed validation of the standard.

This chapter provides a survey of just some of the projects that contribute to the development and implementation of STEP.[1] The survey is not intended to be comprehensive; projects have been selected to demonstrate the different contributions made.

European STEP projects

The collaborative research supported by the Commission of the European Communities within programmes such as ESPRIT and EUREKA has made many significant contributions to the development of STEP. These projects include:

- CAD*I, CADEX and PRODEX: development of the geometric modelling capabilities of STEP, together with demonstration of software prototypes for geometric model exchange and sharing.

- NEUTRABAS, MARITIME: development of STEP for the shipbuilding industry.

- ProcessBase, CIMSTEEL, ATLAS: investigations of the use of STEP in the process plant, structural steel work, and large-scale engineering sectors, respectively.

Several of these projects are described in further detail below. These transnational projects are complemented by national initiatives, such as ProSTEP (Germany), PI-STEP (UK), SPI-NL (the Netherlands), and Caesar Offshore (Norway). Several national “STEP Centres” have been established; these are linked through the European Product Data Exchange Network (EPDEN), as well as through liaison with similar centres in the USA and Asia.

ProSTEP

One of the largest European initiatives is focused on the automotive industry. ProSTEP, initially a German national project, has now widened its scope to cover several countries.

ProSTEP was originally set up as a collaborative project involving industry end-users, systems vendors, and the German Federal government. A two year project, with a budget of DM5.7 million, was started in 1992. The focus of this project was on developing STEP capabilities in the mechanical and electrical engineering areas.

The ProSTEP project identified requirements for two STEP application protocols:

- AP214 “Core data for automotive design processes”, and
- AP212 “Electrotechnical design and installation”.

The development of both application protocols has been led by ProSTEP. The project also encouraged the development of prototype software based on a common toolkit. STEP translators for ten different CAD/CAE systems were developed and demonstrated at a major industry forum in Berlin in October 1993.

Following the successful completion of the initial two-year project, ProSTEP activities have been continued by two organizations:

- the ProSTEP Association is a not-for-profit membership organization that is the focus for continuing standards development and collaborative research;

- ProSTEP Product Data Technology GmbH is a commercial venture that continues the development and marketing of the ProSTEP software and associated training and consultancy services.

Within this second phase ProSTEP has widened its membership to include automotive companies and information technology vendors from other European countries. ProSTEP has strong liaisons with automotive trade associations in the USA and Japan. ProSTEP facilitates a “round table” for users and vendors to discuss and resolve detailed STEP implementation issues.

PDES, Inc.

PDES, Inc. is a US-based consortium formed in 1988 to accelerate the development and implementation of STEP. Although the consortium is largely drawn from US aerospace and automotive companies, together with systems vendors and government agencies, it also includes two major UK aerospace companies.

PDES, Inc. has been one of the most significant driving forces behind the development of STEP, and has taken a leadership role in the development of several STEP Application Protocols, including:

- AP202 “Associative draughting”;
- AP203 “Configuration controlled design”;
- AP207 “Sheet metal die planning and design”;
- AP210 “Electronic printed circuit assembly: design and manufacture”.

Following its major contribution to the completion of the “initial release” of STEP, PDES, Inc. has focused on the deployment of STEP within its member companies. It has supported and encouraged the development of STEP implementations by many major CAD vendors, and has facilitated a number of major implementation and demonstration projects. These include:

- AWS (Advanced Weapon System) STEP project: a joint initiative with the US Department of Defense to demonstrate the use of STEP for configuration management of 3D design data and engineering drawings.

- Harrier: a joint project involving British Aerospace, McDonnell Douglas, Rolls-Royce and a number of software vendors, using STEP to support the exchange of 3D geometry and configuration management information, within a concurrent engineering environment.

- AeroSTEP: exchange of product definition data during the development of commercial aircraft design. The AeroSTEP project is described in more detail in the final section of this chapter.

Through such projects PDES, Inc. is actively encouraging the transition from research and development to the routine use of STEP for data exchange.

STEP in Japan

Japan has been an active contributor to STEP since the development of the standard started. Much of this activity has concentrated on the dissemination of information about STEP within Japanese industry, although in recent years a number of Japanese companies and collaborative projects have become involved in transnational initiatives through links with ProSTEP, PDES, Inc., and PlantSTEP.

The Nippon STEP Center was established in August 1990 as an industry cooperation with government funding, over a four year period. The companies involved in sponsoring the STEP Center were drawn from the mechanical, electrical and shipbuilding sectors, together with a number of information technology vendors and Universities. The STEP Center had a full time technical staff supplemented by resources seconded from member companies.

The Nippon STEP Center was responsible for detailed technical reviews of many STEP parts, and undertook the task of translating the "initial release" into Japanese prior to its adoption as a national standard. The STEP Center also undertook a number of implementation projects, developing a toolkit containing an EXPRESS compiler and SDAI interface generator, and demonstrating prototype implementations of AP201 "Explicit draughting" and AP205 "Mechanical design using surface representation" at NICOGRAPH '93 in Tokyo.

On completion of the Nippon STEP Center project, a number of further projects have been initiated in Japan, including a STEP Promotion Center, and several industry-specific activities in the automotive, ship-building, and process plant sectors.

STEP in the process industries

Although the origins of STEP lie in the CAD data exchange requirements of the aerospace and automotive industries, the period since 1990 has seen a rapid and dramatic increase in the involvement of the process industries. This has been fostered by a number of separate national and international projects within this sector.

The Process Industries STEP Consortium ("PI-STEP") project was established in 1991 by a consortium of UK plant owner/operators, engineering contractors, and systems vendors. The project, part-funded by the UK Department of Trade and Industry, sought to demonstrate the effective implementation of STEP for exchange and sharing of engineering information across the life cycle of process plants.

Progress towards this goal was achieved within three major themes:

- market awareness: the PI-STEP project organized two successful conferences in order to disseminate information about STEP to industry, as well as producing an introductory STEP video and booklet;

- standards leadership: although PI-STEP did not itself undertake the development of STEP Application Protocols, it provided co-leadership of the STEP Process Plant committee, as well as being instrumental in the creation of EPISTLE (see below);

- demonstration: PI-STEP undertook two major demonstrations of STEP capabilities – a "proof of concept" demonstrator (1993), and an advanced demonstration using six major CAD systems (1995).

PI-STEP has spawned several other smaller projects to continue technical development and implementation of STEP in the process industries.

PlantSTEP is a US Consortium, formed in 1994, with the goal of developing STEP capabilities for the design, fabrication, and operation of process plants and other large facilities. The primary work item of PlantSTEP has been the development of AP227 "Plant spatial design", an application protocol that enables the exchange of 3D design data for piping systems, equipment, and other elements of process plants.

This development is accompanied by the development and demonstration of software implementations. PlantSTEP has active liaisons with related projects in Europe and Japan

SPI-NL is a Dutch consortium of plant owners and engineering contractors formed to promote the standardization of electronic information exchange between partners in the process industry.[2] The consortium successfully completed an initial project ("SPIN-OFF") in November 1994, demonstrating the exchange of process and instrumentation diagrams (P&IDs) between different systems. A second project ("SPIN-OFF 2"), targeted at a "data warehouse" implementation based on AP221 "Process plant functional data" is due to be completed in 1996.

The work of POSC, the Petrotechnical Open Software Corporation, is discussed in Chapter 7. In 1994 POSC came together with an existing Norwegian national activity to form the POSC/CAESAR project. POSC/CAESAR represents a significant overlap between the worlds of POSC and STEP, and is developing and applying standards for exchange and sharing of information related to offshore installations and facilities. The project is supported by a number of major oil and gas producing companies.

The various projects working on STEP in the process industries are brought together by EPISTLE – the European Process Industries STEP Technical Liaison Executive. EPISTLE was formed in 1993 following informal liaisons between several European projects working in this area. The basis of EPISTLE is that collaborative projects and individual companies gain benefit from working together on shared problems. The success of EPISTLE is demonstrated by the fact that, from a initial membership of six projects, by mid 1995 more than 30 organizations had participated in EPISTLE's technical workshops and other activities.

STEP in shipbuilding

Shipbuilding has long been established within STEP, having been the subject of one of the application models contained within the "Tokyo draft" of the standard in 1988. After an inactive period, work in the shipbuilding area was re-initiated in the early 1990s, and is now being carried out by several national and international projects.

These projects include:

- NEUTRABAS and MARITIME within the European ESPRIT programme;
- NIDDESC, the Navy-Industry Digital Data Exchange Standards Committee (USA);

- ShipSTEP, an industry project led by Lloyd's Register of Shipping;
- ITiS, a German national project.

The European activities in this area are co-ordinated by EMSA, the European Marine STEP Association, founded in 1994.

These projects are contributing, individually and in collaboration, to the development of several STEP Application Protocols:

- AP215 "Ship arrangement";
- AP216 "Ship moulded forms";
- AP217 "Ship piping";
- AP218 "Ship structures";
- AP226 "Ship mechanical systems".

There are a number of overlaps between these Application Protocols and those being developed or planned in the process plant and offshore sectors.

STEP in building and construction

Like shipbuilding, building and construction is an area that was almost dormant within STEP during the development and completion of the "initial release" of the standard. Since then, however, this has become a very active area, with three Application Protocols in development:

- AP225 "Building elements using explicit shape representation";
- AP228 "Building services: heating, ventilation and air conditioning";
- AP230 "Building structural frame: steel work".

These application protocols have been developed within European collaborative projects such as ATLAS (ESPRIT programme), COMBINE and COMBINE 2 (JOULE programme) and CIMSTEEL (EUREKA programme). There is also strong interest in Australia, which has a very active STEP group in the building and construction area.

Case study: STEP in aerospace

So far, this chapter has presented very brief overviews of a number of projects that have contributed to STEP across different industry sectors. This final section examines a part of one of these projects in more detail. As mentioned above, the PDES, Inc. consortium in the US has been active in fostering pilot implementations of STEP by its member companies. One of the most significant of these pilots is AeroSTEP: the use of STEP in the digital pre-assembly of commercial aircraft engines.

The basis of the AeroSTEP project lies in the commercial relationships between an aircraft manufacturer (Boeing) and its engine suppliers (General Electric, Pratt & Whitney, and Rolls-Royce). Until relatively recently, the only method to check the fit between an aircraft engine and the airframe has been to construct a full-size physical mock-up. These mock-ups are used to check the various interfaces between the airframe, the engine, and the aircraft systems. As the use of advanced CAD technology for both engine and airframe design became standard practice, the need to create these physical mock-ups was becoming an increasingly critical bottleneck in the design process.

The Boeing 777 is the first airliner to be designed completely using CAD. Similarly, the various engines that may be fitted to the 777 are designed using CAD. This therefore creates the opportunity to compare and analyse the designs of the airframe and the engines based on the respective CAD models, and to eliminate the need for the physical mock-up. This use of the CAD models is referred to as “digital pre-assembly” (DPA). Complex data translation is needed to accomplish this requirement, since Boeing and its three major engine suppliers use three different 3D CAD systems: CATIA (Dassault Systemes), CADDS5 (ComputerVision) and Unigraphics (EDS). Previous attempts to exchange data between these systems using IGES, SET or direct translators had not delivered the necessary completeness or accuracy of exchange: this therefore formed the basis for a trial of STEP’s capabilities in comparison to previous methods.

The basis of the AeroSTEP project is the use of one of the STEP “initial release” application protocols – AP203 “Configuration controlled design” – to exchange data between Boeing and its engine suppliers in the context of digital pre-assembly. The usage scenario for the project is illustrated in Figure 24.

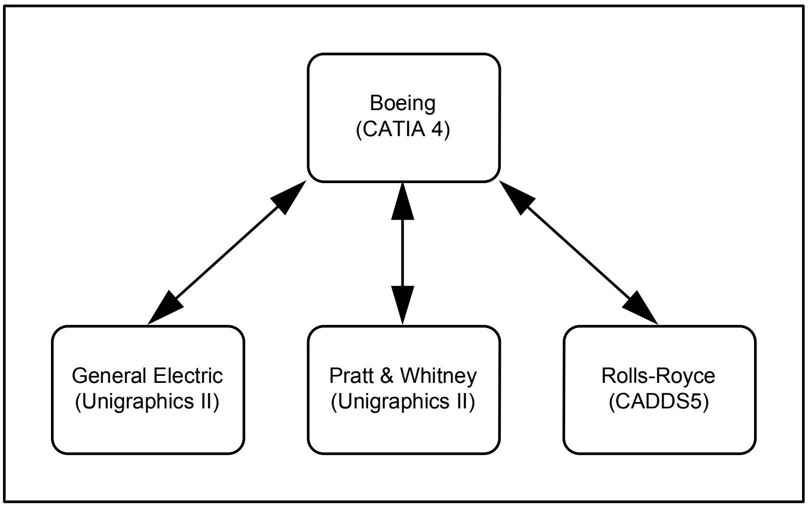

Figure 24: AeroSTEP project scenario

AP203 has a relatively large scope, as shown in Table 5.

Configuration management *authorization* version & revision control effectivity *release status* *security classification* *supplier identification*	**Geometric shape** *advanced BREP solids* faceted BREP solids manifold surface with topology wireframe with topology surfaces and wireframe without topology
Product structure *assemblies* *bills of materials* *part* substitute part alternate part	**Specifications** surface finish material design process CAD filename

Table 5: scope of AP203 (with AeroSTEP subset shown in italics)

However, to facilitate the rapid development and demonstration of the AeroSTEP pilot implementations, a limited subset of the AP203 requirements were selected as the basis for the exchange of data. This subset is shown in italics in the table.

Translators for this subset were developed for each of the systems used, and trial exchanges conducted.

The results of these trial exchanges were very promising, showing real improvement over the previous data exchange methods attempted. A number of issues were raised by the results of the exchanges. The most significant of these concerned the mis-matches between the geometric tolerances of the different systems: this resulted in the same data being regarded as a closed solid in some systems, and as a set of disjoint surfaces in others.

As a result of this experience, a capability was added to AP203 to define, for each solid model being exchanged, the geometric tolerance of the sending system. This allows the receiving system to compensate for any geometrical inconsistencies whilst maintaining the integrity of the exchanged model.

A second and more potentially interesting set of issues arose from the exchange of configuration management data. The ability to exchange such information is unique to STEP among CAD/CAM exchange standards. The analysis carried out of the configuration management data in the various companies revealed a number of significant differences in understanding of terms such as "part", "version", and "assembly". STEP was therefore playing an unexpected role – as a neutral language forming the basis for alignment of working practices and terminology.

The success of the AeroSTEP project may be judged from the fact that complete, production implementations of AP203 are to be used for digital pre-assembly exchanges from late 1995, and that elimination of physical mock-ups is now a real possibility. The shared benefits to the aircraft manufacturer and the engine suppliers from the effective use of STEP are:

- improved data integrity;
- reduction of cycle time;
- greatly reduced effort in data exchange;
- improved quality;
- savings in configuration management.

Together, these represent the potential to save many millions of dollars on each aircraft design or revision.

Summary

The development of STEP, and the demonstration of its benefits through pilot and prototype implementations, has been facilitated by the contributions of many collaborative projects. These projects increasingly reflect commercial alliances and allegiances, and therefore form the basis for "production strength" demonstrations of STEP capabilities, such as those shown by the AeroSTEP project.

The existence of such projects has greatly broadened the scope of industries that are seeking benefits from STEP: by sharing costs and risks through collaboration, companies in these industries are able to reap the full benefits of STEP on the basis of relatively modest investments.

---

[1] It is inevitable that this chapter represents a "time slice" on world-wide STEP activities: at the time of writing, some of the projects discussed are complete, others are at some mid point, others are being planned or initiated. Annex D includes contact details for some of the projects and consortia discussed; readers are advised to contact these organizations for up-to-date information.

[2] 'Samenwerkingsverband voor de Proces-Industrie in Nederland' – Co-operative Association for the Process Industry in the Netherlands

10 STEP software

In Chapter 6, details of STEP implementations have been discussed in terms of the operation of interfaces for both file exchange and access to shared databases. One of the features of STEP that distinguishes it from other CAx exchange standards is the fact that so much of the standard is specified in *computer interpretable* form. A software developer producing an IGES translator has to read the IGES manual (some 600 pages), and base the translator on interpretation of the natural language (American English) definitions found in the standard.

With STEP, however, the standard is based on data models defined in EXPRESS, and on formal syntaxes such as that of the exchange format. This therefore opens up the possibility of automation of software development processes, based on the use of tools that process STEP data models and file syntax, and therefore both ease and accelerate the software development effort.

This chapter presents a brief survey of the types of tool that have been developed for use with EXPRESS and STEP, and discusses how they are used. It includes the results of some informal market surveys of STEP translators carried out in 1994-1995; in a rapidly changing market, this information will quickly become obsolete. Readers are strongly advised to seek up-to-date information from systems vendors and consultants to supplement the results reported here.

Types of STEP software

STEP software tools and products fall into several categories, determined by the uses to which they are put.

- Model development and validation tools are used by data modellers, including the developers of the STEP standard itself, as aids to the production and checking of EXPRESS data models.

- Implementation tools are used by software developers and systems integrators to support the development and testing of translators, interfaces, and other software components that conform to the standard.

- Interfaces, translators, and other tools used by CAx systems users.

- Other "STEP-based" software, such as migration tools and direct translators.

The following sections discuss each type of STEP software in turn.

Modelling tools

Model creation tools are used in developing EXPRESS data models; they include EXPRESS editors ("language sensitive" text processing tools that have templates for the EXPRESS language built in, and are often linked to language parsers), and EXPRESS-G tools that support the development of models using the graphical form of the language. In both cases, these tools may be simple "stand-alone" applications, or may form part of a more comprehensive model development environment through links to other tools.

EXPRESS-G modelling tools, as well as providing capabilities for creation and modification of EXPRESS-G diagrams, often support the generation of the "lexical" form of the language based on diagrammatic input, or allow automatic or semi-automatic generation of EXPRESS-G diagrams from lexical input.

The earliest EXPRESS tools developed were those that take an EXPRESS data model as input and check it for errors in syntax, semantics, and cross-references. Such tools, commonly referred to as EXPRESS parsers, may themselves be generated automatically from the syntax rules of the EXPRESS language. EXPRESS parsers form the "core" of all other EXPRESS-based software tools, since they undertake the initial checking on an EXPRESS data model (as an ASCII text file), and make it available to other tool components for further processing.

The third category of tools used by data modellers are those used to process a model to generate other models. Appendix A describes the development of a "short form" EXPRESS model within a STEP Application Protocol, and the generation from this model, plus the STEP Integrated Resources, of a "long form" model that is a complete, self-contained single schema. The development of long form schemas may be automated through the use of suitable tools.

Many of the tools used in the development and implementation of STEP offer equivalent functionality to those in the domain of CASE – Computer Aided Software Engineering. The fact that STEP has created its own "cottage industry" of tool developers and suppliers results from a lack of awareness and support for EXPRESS as a modelling language within "mainstream" CASE tool vendors. It is likely, however, as EXPRESS gains wider acceptance outside the STEP development activity then it will be supported by major CASE tool vendors. Some initial studies have already taken place, and the possible future links between EXPRESS and CDIF should allow EXPRESS models to be created, managed and used from within powerful CASE environments.[1]

Implementation tools

The second major category of STEP software covers the tools that are used by implementors of STEP translators, interfaces, and applications. The "starting point" for these tools will be a complete, validated data specification in the form of one or more EXPRESS models, plus associated documentation.

The term "compiler" is used for several different types of implementation tool.[2] All share the basic, characteristic resemblance to a programming language compiler, in that an input in one language (EXPRESS) is processed and converted into some other form. Compilers that are used in the development of STEP interfaces and translators may perform one of several functions.

- Process an EXPRESS model into data structure definitions of a programming language, and create programming language functions that may then be used to read, write, and manipulate data within those structures.

- Process an EXPRESS model into data definition language (DDL) and data manipulation language (DML) statements, suitable for use with an appropriate database management system.

- Process an EXPRESS model into appropriate function calls for use with the Standard Data Access Interface, SDAI.

In each case, the compiler delivers to the programmer a form of the original EXPRESS model that is suitable for inclusion, through conventional software engineering techniques, within a translator or application package. This use of EXPRESS "compilers" is a key factor in the increased efficiency of translator development that is offered by STEP, since it allows developers to produce substantial amounts of the code needed in a translator automatically from the EXPRESS models provided as part of each STEP Application Protocol.

Some developers have gone beyond the use of compiler technologies that process the standard EXPRESS models supplied by STEP, by developing mapping tools that operate on an EXPRESS representation of the *internal* data structures of a CAx application. A typical configuration for the use of mapping tools is shown in Figure 25 below.

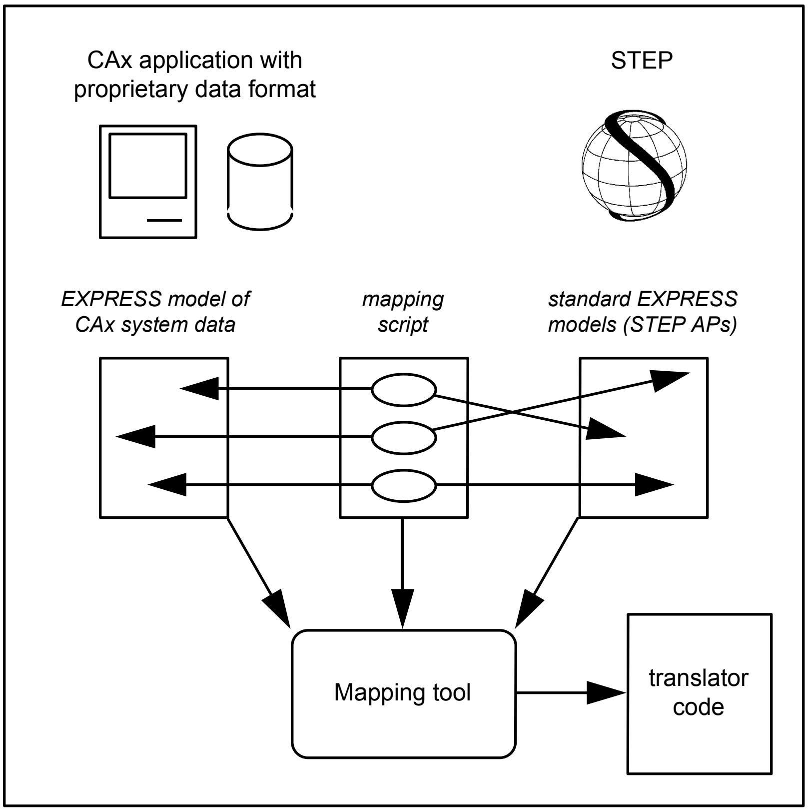

Figure 25: use of mapping tools

Here, a "mapping script" is used to describe the mappings between the system EXPRESS model and the STEP EXPRESS model(s); the mapping tool processes these to produce complete code for a translator. This type of tool changes the task of translator development from one of software engineering to one of data analysis: given an EXPRESS representation of the internal data structures of the CAx system, and a mapping between this representation and STEP, translator development may become completely automated.

Several of the approaches to STEP implementation that make use of mapping tools embody extensions to the EXPRESS language that are used to construct the key mapping script. These extensions are included in the proposals for the development of a second edition of the EXPRESS language (see Appendix B).

Compilers and mapping tools are just two of the components that may be found in a typical STEP development "toolkit". Other components may include:

- scanner/parsers for STEP physical files, that convert the STEP ASCII file representation into a “working form” suitable for further processing;

- file formatters, that convert a working form into STEP physical files;

- class libraries for specific STEP application protocols, ie. data structure definitions and associated functions, potentially generated directly from an EXPRESS model;

- data browsers for use in testing and debugging translators.

Interfaces and translators

As noted in the introduction to this chapter, it is impossible to capture the current “state of the art” of STEP interfaces and translators without immediately becoming out of date. Therefore, this section presents a primarily historical survey of the development of STEP implementations.

The first phase of STEP implementation (1988-1991) developed early prototypes using interim versions of EXPRESS and the STEP data models. These prototypes were generally produced within collaborative research and development projects, including:

- ESPRIT projects such as CADEX, IMPACT, NEUTRABAS, and NIRO;

- The National PDES Testbed at NIST;

- The initial prototypes developed by PDES, Inc.;

- The US Navy’s RAMP programme.[3]

Information on most of these projects has been given in Chapter 9. Many of these prototypes were based on software developed by previous projects, such as the ESPRIT CAD*I project and the USAF GMAP and PDDI programmes.

The primary focus of these implementation projects was the testing and validation of the STEP methodology and models, rather than the development of “production quality” software. This aspect of prototyping has proved highly effective in improving the quality of the standard during its development. Indeed, the development of such prototypes is now required as part of the application protocol development process.

From 1991 onwards, the contents of the "initial release" of STEP were reaching stability and maturity, and second generation implementations became possible. The focus of this phase of STEP implementation projects was the detailed evaluation of specific STEP application protocols (primarily AP203 "Configuration controlled design") against end-user requirements and commercial CAx systems.

The AeroSTEP project, discussed in detail in the previous chapter, is a major example of such implementation projects. Others included:

- assessment by other PDES, Inc. member companies of the capabilities of AP203;

- development and demonstration of AP203 and AP214 prototypes within the first ProSTEP project;

- prototype implementations and demonstrations of AP201 by the Japan STEP Center.

These advanced prototypes were significant not only in the detailed feedback provided to the final development of the "initial release" of STEP, but also in enabling powerful demonstrations of the potential capabilities of STEP in real industrial environments. The incremental nature of STEP development, with application protocols being developed and published on a phased schedule, means that some of the industries not supported by the "initial release" application protocols are now developing such advanced prototypes. For example, ProSTEP is facilitating the development of production-quality implementations of the Committee Draft version of AP214, and various projects developing STEP for the process industries are planning major demonstrations in the period 1996-97.

A second aspect of this phase of STEP implementation was the evaluation of early versions of the Standard Data Access Interface (SDAI). More than any other component of STEP, the SDAI represents the results of major pre-standardization implementation initiatives.

	AP201	AP202	AP203	AP214
ComputerVision	☆		☆	☆
ConCAD GmbH	✪			
Dassault Systems			☆	☆
EDS Unigraphics	☆	☆	✪	☆
Grumman Data Systems			✪	
Hewlett Packard			✪	☆
Intergraph	☆	☆	✪	☆
MCS			☆	
Parametric Technology			✪	
SDRC			☆	☆
Sherpa			☆	
STEP Tools, Inc.			✪	
Team SCRA			✪	
Theorem Solutions			✪	

Key: ☆: prototype implementation
✪: commercially supported implementation

Table 6: STEP implementations, 1994-95

Before the publication of the initial release of STEP, few CAx system vendors had released, or committed to release, STEP translator products. This situation has, however, changed rapidly. Surveys undertaken in late 1994 and early 1995 showed that fourteen suppliers had announced STEP translators, including five of the top seven CAD vendors in 1994.[4] A summary of these surveys is given in Table 6 above.[5]

It is notable that many of the companies developing STEP implementations are "third parties", ie. they are not the originators of the CAx systems for which the translators are developed. This parallels the situation with the development of IGES translators; end-users will now often have a choice of sources for translators. This choice and competition can only lead to an increased and maintained high level of quality in translators.

All the software described above aims to be "conforming", ie. to comply with the requirements of ISO 10303. However, there are other software products that are being developed and used that make use of STEP as an underlying technology without claiming conformance to a particular part of the standard.

A key requirement for many companies is to be able to migrate current product information into STEP-based data management environments. Where this data is held in a CAx system that has an appropriate STEP interface, this may be readily accomplished. However, there are cases where such migration is not so simple.

- The data is held in a "legacy" system for which STEP interfaces are not being developed.

- The data has been archived in an alternative neutral format, such as IGES or SET.

Since most "legacy" systems will have interfaces to standards such as IGES or DXF, tools have been developed that convert from these "older" formats into STEP. This cannot be a complete conversion: as noted in previous chapters, STEP includes concepts of product identification and product structure that are not present in IGES. The migration tools developed for such requirements therefore supply this "additional" data, so that the result of the conversion may be successfully managed in a STEP environment.

The second class of "STEP-based" tools is that of direct translators. In Chapter 2, a "hub-and-spoke" architecture for direct translators was described. Some developers of such systems have made extensive use of STEP as the basis for the internal format in the "hub". This usage of STEP illustrates another of the standard's key benefits: the availability, in the public domain, of world-class information on product data and product data modelling.

Summary

STEP is supported by a wide range of software tools, covering the development and review of data models, the creation of translators and interfaces, and those translators and interfaces themselves. STEP has benefited greatly from major prototype implementation projects, which have not only validated the structure and content of the standard, but also given the developers of STEP software the experience necessary to bring commercially supported products to the market.

---

[1] The CASE Data Interchange Format – see Appendix B.

[2] The term “compiler” is also used to refer to the type of tool here called a “parser”. The normal distinction is that a parser analyses an EXPRESS file, reports any errors, and makes the structure and content of the EXPRESS model available for further use. A compiler provides additional functionality, in that as well as parsing and checking the EXPRESS file, the compiler will convert the EXPRESS model into some other form.

[3] Rapid Acquisition of Manufactured Parts, a programme that makes use of STEP coupled with flexible manufacturing cell technology to enable “production on demand” of spare parts from digital product data.

[4] Source: Daratech, Inc.

[5] Sources: ISO TC184/SC4 Secretariat; CADDETC.

11 STEP implementation strategies

This chapter identifies some of the strategies that may be employed in introducing STEP within a company or project.

Exploiting the potential of STEP

In Chapter 8 many of the potential benefits of STEP are identified and discussed. Clearly, there will be very few companies to which every possible STEP benefit is applicable. In developing a strategy for the implementation of STEP, it is therefore necessary to identify the specific benefits that are applicable, and to plan acquisition and deployment of appropriate software tools in this context.

One of the results of the analysis of the potential benefits of STEP presented in Chapter 8 is that the major, significant benefits are to be gained through improvement of business processes and realization of key business objectives. This implies that a strategy for the implementation of STEP must involve more than the engineering computing function. If the benefits of STEP include those that result from the integration of design and information data with that produced in manufacturing and operations, then the decisions pertaining to STEP need to be taken at the level of a corporate information technology policy.

STEP makes extensive use of the tools and techniques of activity modelling and data modelling in the development of the standard. These models are used to gain insight into the business processes that the standard should support, and into the structure and content of the data used in these processes. Similar insight can be gained by applying the same techniques to individual enterprises.

Development of an effective implementation strategy for STEP can be significantly enhanced by such analyses, through which the potential benefit of STEP to an enterprise may be identified. Since it is precisely these analyses that form the basis of business process re-engineering programmes, it may prove effective to conduct a STEP benefits analysis alongside, or as part of, such programmes.

STEP policy statement

None of the analyses described above are likely to result in success if there is not effective "buy-in" to STEP at a senior level in the company. Without such commitment, it is likely that proposals for STEP implementation will fail. The development of a STEP Policy Statement can have many benefits:

- simply asking the question "What is our policy on STEP?" can be effective in bringing STEP to the attention of the Board or other senior managers;[1]

- identification of the need for such a policy statement may prompt the need for wider awareness of STEP within the company, through distribution of appropriate reading material, organization of seminars, and participation in conferences;

- once drafted, a suitable policy statement can be used as part of the process of evaluating proposals for STEP implementation.

A typical approach to the development of such a policy statement is that recommended by the UK PI-STEP project:[2]

> *Set your management the question:*
>
> *If all the information about my plant, process and products was available in a form that was independent of any computer hardware and software, or indeed of the organization that produced it …*
>
> *… then what opportunities would exist for changing the way we work, to produce a breakthrough in reducing the time to bring new plant on-line, reducing plant down-time for major refits and maintenance, improving safety and quality, and reducing costs?*

Planning for STEP implementation

Once a STEP policy statement has been developed and agreed upon, there is still much to do if benefits of the kind described in Chapter 8 are to be realized.

The analyses described above allow the identification of the business processes and information flows that may benefit from improved exchange or sharing of data. The next stage is to identify how STEP may relate to these needs.

Given that STEP consists of many different documents, comprising many thousands of pages, one cannot begin at page 1 of Part 1 of the standard, and keep reading until one find something relevant to one's needs! Fortunately, the structure of the standard allows much more rapid analysis of the capability of STEP. As discussed in Chapter 5, the essential components of STEP for the end-user are the application protocols. Each application protocol defines the standard data model to be used for communication of data in a specific industrial application context.

The title of each application protocol provides the first indication of this intended use. Look at the list of application protocols included in Appendix C: a shipbuilding company would, for example, recognize five application protocols as immediately pertaining to its business activities. Clearly, however, it is not sufficient to match your company's needs to the capabilities of the standard just through the title of the parts. Again, the structure of the STEP documentation within each application protocol supports assessment of its relevance to your requirements.

- The scope statement provides a textual description of the intended use and applicability of the application protocol.[3]

- The application activity model (AAM) presents in graphical form the activities and information flows covered by the scope of the application protocol. This may be compared with any activity models or business models developed within your company.

- A data planning model, normally given as part of the scope statement, will provide a high level description of the types of data that the application protocol supports.

- The application reference model (ARM), and the associated definitions of units of functionality and application objects, describe in detail the types of data that the application protocol supports. This description will use the perspective and terminology of an expert in the field.

On the basis of reading, understanding, and analysing these elements of the application protocol, you will be able to make a decision as to the relevance of the application protocol to your needs. The other elements of the application protocol: the mapping table, the application interpreted model (AIM), and the conformance classes, are equally important but are not relevant at this stage.

The results of this survey of STEP application protocols will have one of five different results:

1. My requirements are completely satisfied by one application protocol.

2. My requirements are satisfied by a combination of two or more application protocols.

3. My requirements are partially satisfied by one or more application protocols.

4. My requirements are fulfilled by one or more application protocols, but these include data for which I have no need.

5. My requirements are not satisfied by any application protocols.

If the result is (1) or (2), then implementation of STEP in your company can focus on the acquisition of appropriate STEP interfaces and translators that meet your needs. If the result in (4), then it may be worth revisiting analysis of your own requirements: the organizations responsible for developing the application protocol had good reasons for inclusion of this data.

Similarly, if the result is (3), you may wish to consider whether the application protocol delivers real benefits to you even if it does not support *all* your data requirements. Otherwise, you may wish to review, possibly with others in your industry sector, how the application protocol may be enhanced or extended to meet your needs.

This illustrates the high importance of becoming involved in the development and review of STEP. You cannot expect that STEP will automatically meet your needs. It may seem costly to invest in the necessary time and training to allow engineers to participate in review of standards documents; however, if the resulting standard works to the benefit of your competitors, then the impact of not reviewing and seeking improvement to the documents will be much greater.

This also applies if there is no application protocol that meets your needs. There may, of course, be other standards or technologies outside of STEP that do meet these needs. If not, you may wish to consider the benefits of joining with others to foster the development of the capability within STEP. As discussed in Chapter 9, few companies will be likely to bear the large costs of such development alone; however, it is likely that there will be other companies in your industry sector with similar requirements and interests, offering the potential for collaboration and reduction of both costs and risks. Once again, the costs of participating in the development must be offset against the risk of the standard not being appropriate to your business requirements.

Preparing for implementation

Usage scenarios – descriptions of the intended use of STEP – play an important part in the initial development of application protocols. Similarly, they can be used as part of the process of evaluating the potential use and benefits of STEP within a company. Such scenarios can also form the basis for prioritising phased implementation, and should consider both exchange of data (data communication), and shared access to common data (data integration).

Development of such scenarios should consider at least some of the following:

- exchange of data between different systems used in the same department or function within the company;
- exchange of data between different departments or functions within the company;
- exchange of data with a partner, supplier or sub-contractor;
- sharing of data between different systems used in the same department or function within the company;
- sharing of data between different departments or functions within the company.

In some organizations it may also be appropriate to consider sharing of data with a partner, supplier or sub-contractor.

Consideration of these scenarios allows an implementation strategy to be planned. This may be “incremental” in nature, starting with simple file-based exchange within the company and moving on to more advanced implementations, or may take the form of a complete implementation supporting all the above scenarios through a STEP-based information infrastructure.

Acquisition and migration planning

Once a suitable STEP application protocol has been identified, it is necessary to acquire the interfaces or translators that implement this application protocol. As the use of STEP becomes more widespread, it will be increasingly likely that such software will be available from application vendors, as well as from specialist suppliers.

It is, of course, not sufficient to seek a “STEP translator” from suppliers. On the basis of the analysis of your business needs for STEP, it is necessary to determine from suppliers of STEP interfaces for your systems:

- which application protocols are supported?
- which conformance classes of these are supported?
- which implementation forms are supported (exchange file, SDAI)?

From the answers to these questions, it will be possible to determine whether there are products on the market that meet your needs.

As with all computer systems, it is essential that STEP interfaces are rigorously tested and validated before they are used in production projects. This testing and validation needs to address several issues.

- **Conformance**: does the software comply with all the requirements of the standard?
- **Interoperability**: does the software enable exchange of data with other implementations?
- **Performance**: does the software make reasonable demands on processor time, memory and disk space in terms of the benefit achieved?
- **Robustness**: does the software respond in a controlled and predictable manner to erroneous or incomplete data?
- **Usability**: is the software easy to use, and is it well integrated with the other components of the system of which it is a part?

Acquisition plans for STEP interfaces should include all these aspects of testing and validation. As discussed in Chapter 5, STEP places a strong emphasis on testing and testability, particularly in the area of conformance testing. If there is sufficient user demand for it, testing laboratories world-wide will be able to provide conformance assessment services; certification and “branding” schemes may be developed on the basis of conformance test reports.[4]

The abstract test suites that are used in the formal assessment of conformance may also be useful in developing test cases for other aspects of testing and validation. It must be remembered, however, that conformance testing is concerned with assessment against the requirements of the standard; the other types of testing listed above are concerned with the specific requirements of a user.

Through the application of appropriate selection, testing and validation techniques, appropriate STEP interfaces may be acquired. The final hurdle to full implementation of STEP is then that of migration: “how do we get there from here”. The answer to this often seems to be that of the classic joke: “Well, I wouldn’t start from here!”. Even with comprehensive acquisition policies, implementation of STEP is not just a question of installing the software and allowing users to work with it. As previously discussed, the real benefit of STEP comes from *how* it is used, and it may therefore be necessary as part of a STEP implementation plan, to implement process changes and improvements in parallel with the installation and use of STEP software.

Clearly, this can not be done easily in the middle of a project: it would not be advisable, for example, to introduce STEP across the board mid way through the development of a new product. Implementation and migration plans should therefore identify those projects in which the introduction of STEP is not only beneficial, but can be implemented without major impact on the schedule or resources of the project. These plans should consider the possible need for migration tools, ie. software that will assist in moving data from an “old” (pre-STEP) environment into one in which STEP is to be used. This may include translators from “legacy” systems to STEP, or from other neutral formats such as IGES to STEP.

It is difficult to demonstrate the benefits of STEP unless some means is found to measure those benefits. Implementation plans should therefore define the metrics and criteria against which the success of the plan is to be judged. The form of these metrics will vary considerably between companies; as identified in Chapter 8, the benefits of STEP are company-specific and so, therefore, are the metrics for benefit measurement. These metrics should, however, be both realistic and useful, and should be relevant to the *highest* level objective set for the implementation. If the benefits of STEP have been sold within the company on the basis of increased profits or reduced time to market, then successful implementation reported in terms of errors per exchange file may not be considered as a success.

Summary

Successful implementation of STEP can be a complex process, requiring not only detailed technical evaluation of the standard and the software that implements it, but also positioning of the standard within information technology strategies and the development of a corporate policy for STEP. The development of comprehensive evaluation, acquisition, implementation and migration plans for STEP is a necessity: without them, many of the potential benefits of STEP may be lost.

---

[1] This may particularly be the case if competitor companies have already developed such a policy statement!

[2] Taken from “Executive Guide to STEP for the Process Industries”

[3] Every part of STEP (indeed, every ISO standard) includes a Scope statement as its first normative clause.

[4] After an uncertain start, this demand seems to be building momentum. For example, one of the outcomes of the AeroSTEP project (see Chapter 9) is the requirement by Boeing on its CAD vendors, and those of its engine suppliers, to have independent conformance testing of their STEP (AP203) translators.

12 Conclusions: a glimpse of the future?

This chapter draws together various themes from the earlier parts of the book, and summarises the key impacts of STEP on business improvement, contractual relationships between customers and suppliers, the continued development of information technology systems, and strategies such as concurrent engineering and total quality management.

STEP offers the potential for complete, accurate exchange of product information amongst engineering applications, as well as forming the basis for the implementation of shared product databases. Initial implementations of STEP has shown that it is already providing a robust, reliable alternative to previous standards such as IGES, as well as offering significant performance benefits. The computer-interpretable basis of STEP has enabled the development of many modelling and implementation tools that have brought high degrees of automation to the development of STEP interfaces. This not only increases the quality of the software delivered to end-users, but also, by encouraging the repeated re-use of software components, reduces the costs of acquiring and maintaining translators.

The impact of STEP as a standard

STEP is delivering standard representations of product data to industry. As with all standards, these are used as contractual requirements on suppliers. Within STEP, one class of parts (application protocols) specify the constructs required for communication of product data in a stated industrial context. It is these parts of the standard that therefore play a role in contracts.

Customers are able to specify precise requirements for the delivery of, for example, engineering drawings, and various forms of geometric model and configuration controlled design data, simply by referencing the appropriate STEP application protocols. The key difference here between STEP and a standard such as IGES is that the latter specifies only the syntax of the exchange representation; STEP specifies the syntax and the semantics, ie. the precise meaning of all data constructs. Coupled with the increased quality and reliability of translator software, this reduces the need for negotiation between partners before data exchange can take place, or for manual intervention in the data exchange process.

The "portfolio" of STEP Application Protocols will increase with time, increasing the scope of applicability of the standard as a contractual requirement. It is important to note here that the content and schedule of additional Application Protocols is determined by the international industrial community, not by the "standards experts" within the ISO committees.

The size and scope of STEP may suggest that it is primarily a standard for large companies. This view is captured in the following statement:

> *"(STEP) ... will achieve critical mass only when a major partnership enterprise ... commits to its goals through changing the engineering paradigm,* and actually does so.
>
> *"In all probability, this will not happen until a major technology vendor ... who can operate at the level of the total partnership enterprise, contracts to achieve the goals of the enterprise through integrating STEP into the enterprise transaction infrastructure."*
>
> Daniel Appleton
> D. Appleton Company, 1992[1]

As discussed in Chapters 8 and 11, however, STEP also offers short-term benefits at the business level and the technology level. These benefits can be realized by any company that makes use of computer-aided engineering technologies; even if the only benefit gained from STEP is the effective exchange of data with a customer or supplier, this represents a major step forward for many organizations.

Impact on customers and suppliers

The discussion above outlines the role of STEP in contractual relationships between customers and suppliers. However, STEP will have a wider impact than this, in that it also supports new and emerging technologies which will change the very nature of the relationship between a customer and a supplier.

In recent years the application of EDI (Electronic Data Interchange) standards and technologies has introduced the concept of "paperless trading". This has proved highly effective in areas such as travel, banking, retail and others characterized by the need to exchange high volumes of relatively small and simple messages, such as orders, invoices, and payments. The success of EDI in these areas has created the possibility of using similar methods for exchange of technical information. There are, however, a number of problems that have impeded progress in this area; in particular, the capabilities of commercial telecommunications and value-added network providers are not designed to cope with the lower volumes but significantly higher message sizes associated with product data interchange (PDI).

However, improvements to the communications infrastructure through the provision of services such as the Internet and ISDN, and enhancements to the EDI standards such as EDIFACT and ANSI X.12 mean that the "how" of PDI is rapidly becoming available. It only remains for STEP to provide the "what", ie. the standardized messages for product data, for a new level of "paperless trading" to emerge.

In this environment, PDI using STEP and related standards will allow suppliers to make complete, system independent catalogues of their products available to customers and to potential customers. These catalogues will include not only pricing and ordering information but also specifications and other appropriate product data. This will allow customers to access catalogue part data (suppliers may make "read-only" access available freely, while charging for data that is used by a customer) for use in CAD or CAE, as well as automatically generating necessary ordering information from bills of materials.

Just as PDI and STEP will enable suppliers to make product information available in standard form, it will enable customers to circulate invitations to tender to potential suppliers, or to supply appropriate product definitions to subcontractors for manufacture.

These technological innovations not only provide improved communications between customers and suppliers, but also, in the short to medium term at least, imply that suppliers will compete not only on the ability to supply products but also, and possibly more importantly, on the ability to supply information about those products. The role of STEP here is that the supplier need only supply this product information in one format, rather than those of a large number of proprietary CAD systems.

These technological changes will support the creation of highly flexible trading networks, using PDI as a support to lean manufacturing and just-in-time supply, reflecting common industry requirements for such improvements to business processes.

Impact on information technology systems

The most significant impact of STEP may be that on advanced information technology (IT) systems. This impact may be summarized in two words: *open data.* The trend over the past ten years towards Open Systems has had significant impact on the information technology market in that customers are, in some cases at least, able to "mix and match" system components in the knowledge that interoperability between these systems is possible. However, this capability extends only to hardware, communications and some system software. In general, applications software is not portable across different environments, and applications are closely and indivisibly coupled to the data upon which they operate.

Considered in terms of applications used in design, engineering and manufacturing, this means that users become tied to single suppliers for tools in areas such as CAD, CAE and CAM. The close link between the application software and the application database results in the need for complex integration tasks; even if the major CAx tools are acquired from a single supplier, most (if not all) manufacturing enterprises will have a number of "home grown" applications which must be integrated with the supplier's database environment.

It must be recognized that the initial level of STEP capability (file exchange) will help to alleviate this problem by providing improved, unambiguous communication between proprietary and home grown applications. However, as the basis for the specification for shared databases STEP will enable a division between application software and the data upon which applications work. Users will then support (in an "Open Systems" distributed, heterogeneous environment) design, engineering and manufacturing data, where the content, structure and organization of this data is specified by STEP. Applications software, both proprietary and home grown, will create, access, modify and act upon this data through a standardized data access interface. Applications software developers and suppliers will concentrate on the functionality of their systems rather than on core database capabilities.

The impact of STEP on information technology systems will be fully realized by using STEP alongside other key standards and technologies. Figure 26 below illustrates the basic data sharing paradigm supported by STEP: applications share data in a common database, access to which is provided by the SDAI and appropriate application protocols.

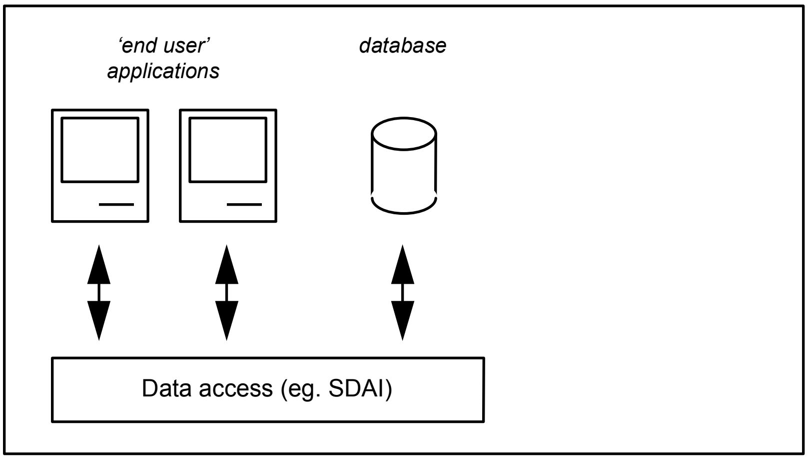

Figure 26: data sharing using SDAI

However, this does not recognize the fact that the information to be shared may reside in different databases. This may be for reasons such as performance or ownership of data. Adding further databases to the information technology environment will support this need, as shown in Figure 27; however, this places considerable overheads on the applications, the database management systems, or both.

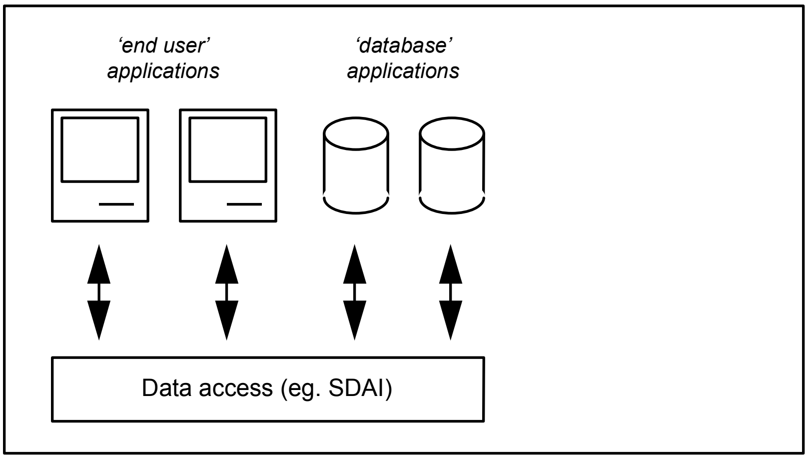

Figure 27: multiple databases

An extension to this architecture may be developed by reference to the "client-server" paradigm, as shown in Figure 28. Here, the applications are classified as "clients" of the infrastructure; the databases are classified as "servers", as are input-output functions such as STEP physical file and EDI facilities.

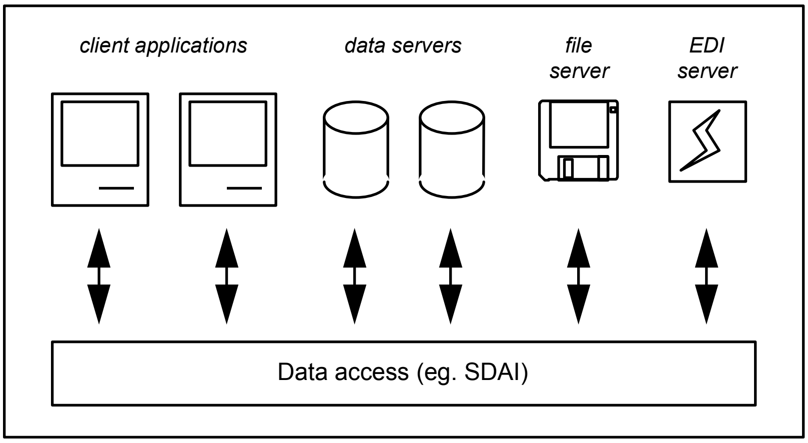

Figure 28: STEP in a client-server architecture

The addition of database servers and file servers creates additional requirements on the SDAI. Because the SDAI is focused on access to data, it assumes that the applications that make use of it are able to maintain references to the locations of data. In the environment shown in Figure 28, an application will have to track data across several different servers. It is likely that the effective use of such an environment requires an engineering data management system, of the type discussed in Chapter 3.

A potential alternative is offered by another standard: CORBA, the Common Object Request Broker Architecture.[2] CORBA provides to both clients and servers facilities that manage the location and transport of data. As shown in Figure 29, the combination of SDAI and CORBA enables a data management infrastructure in which:

- the communication of data between systems (client-client, client-server, or server-server) is accomplished using STEP application protocols;
- access to data within an application or database is provided by the SDAI;
- management of the location of data, and its availability to clients and servers, is managed by CORBA.

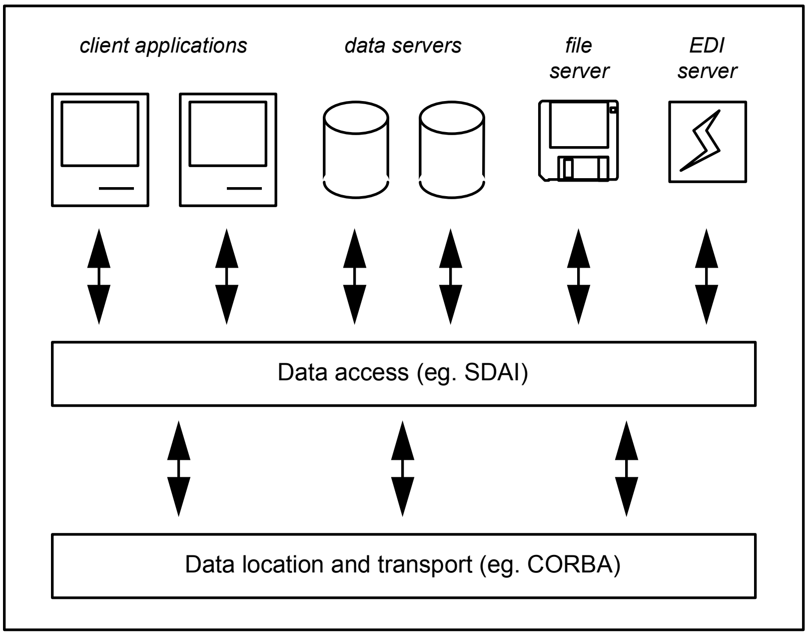

Figure 29: information technology infrastructure using SDAI and CORBA

A user interacting with this environment need have no knowledge of the format or location in which data is stored: these functions are managed "invisibly" by the SDAI/CORBA combination.

The use of such an infrastructure has been investigated in several collaborative research and development projects.[3] These investigations have revealed several issues related to the interaction between SDAI and CORBA: once resolved, this combination offers a powerful set of tools for enterprise data integration.

The evolution of infrastructures described above represents a potential impact of STEP in the area of databases and data management. It will be noted, however, that no changes are assumed to the applications that are used by end-users. This means that the integration of applications is still achieved by translation and exchange of data between systems.

STEP and its related technologies also open the door to a new generation of software products, based on a modular "plug and play" architecture. The basis of such applications is illustrated in Figure 30. This depicts a "platform" providing integration services to different information technology modules; SDAI and CORBA, as discussed in the previous section, may be part of the platform technology.

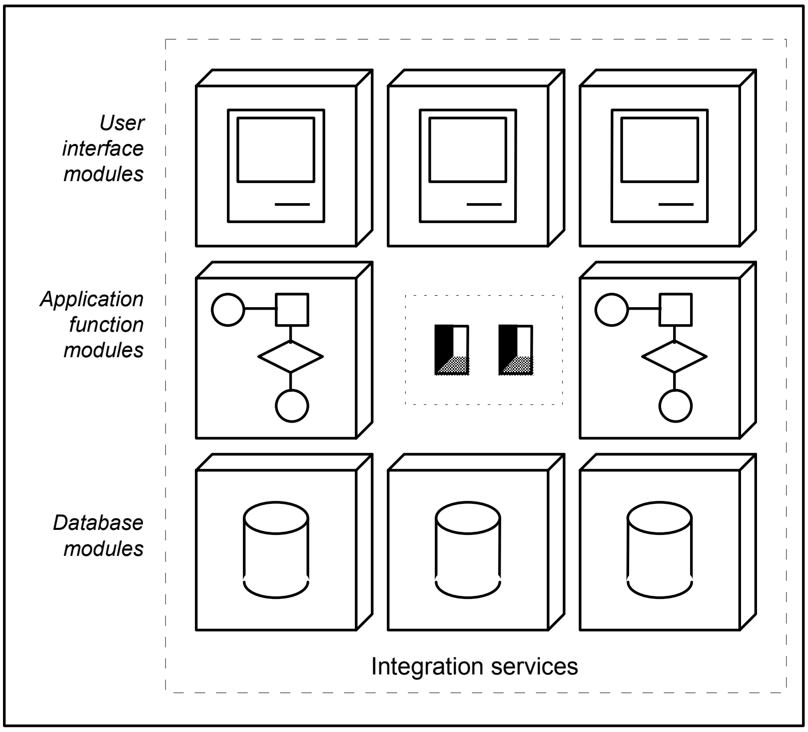

Figure 30: modular, user-configured software

In such an architecture, applications developers will be able to concentrate on the core functionality of their products, on the basis that users will choose the user interface and database tools that they will combine, with the integration services infrastructure, to create a complete system. Such tools and systems are being developed in the POSC environment in the oil industry: it is only a matter of time before equivalent products are developed for STEP.

STEP and Concurrent Engineering

STEP will support Concurrent Engineering (CE) by providing integration of product data across disciplines and applications. It is one of the key issues that have emerged during the development of STEP that there is a need not only for the exchange of data between similar systems but also for the sharing of data between dissimilar systems. Thus, STEP Application Protocols are developed in such a way that the shareable information between (say) boundary representation models for design and finite element models used in analysis may be identified. In a file exchange environment, this enables an FEM system to extract relevant information from a solid modelling system; more importantly, this supports shared database implementation and appropriate interworking between disciplines in a concurrent engineering environment.

STEP and Total Quality Management

In terms of total quality management (TQM), many organizations have found that in rationalizing working practices and documenting procedures in order to meet the requirements of quality standards such as ISO 9000 the management of product data has become a major quality issue.

Electronic data exchange, whether it be used as an external interface to customers or suppliers, or as an internal tool as part of a CIM environment, can represent a major area both of information loss, due to failings in data exchange technology, and of management problems where responsibility for data is not clear. While proprietary engineering data management (EDM) systems may help in this situation, the fact that multiple versions of the same data may exist across a number of different environments represents a major quality problem.

In the shared database environments discussed above, EDM applications will provide a “shell” to the environment ensuring that only one active, up to date copy of each dataset exists, and that this is accessed and/or modified only by authorized individuals or applications.

Summary and conclusions

This book has discussed the management of complex engineering data, and has identified many of the resulting requirements from industry. Current approaches to data exchange, sharing and management, and the support for these requirements offered by ISO 10303 “STEP” have been discussed.

It has been demonstrated that while STEP is not of itself a solution to data management requirements, it is a key enabling standard for such solutions, and will play a key role in the future in supporting data management in the context of process improvement, concurrent engineering and quality management.

At this time, the potential offered by STEP is only starting to be realized, and there are many problems to solve and pitfalls to be overcome before many of the approaches discussed in this book can be adopted as “best current practice” in industry. However, it is clear that many industry sectors world-wide are investing now in development and demonstration projects that are slowly but surely moving STEP from a research-oriented standards activity to a practical and effective long-term solution to industry data management, exchange and sharing requirements.

---

[1] This quotation is taken from a presentation entitled "The Business Case for PDES", presented to an IGES/PDES Organization meeting held in Salt Lake City, Utah.

[2] This standard is developed by the Object Management Group (OMG), a consortium of major suppliers of object-oriented technologies.

[3] The diagrams used to illustrate this section are derived from material produced by ESPRIT project 6874.

Appendix A: The STEP architecture and Methodology

The high level structure and contents of the STEP standard are described in Chapter 5. This high level structure is based around the division of the STEP standard into a number of different classes of parts. This appendix, however, describes the underlying architecture of STEP, and the methods for product data specification developed and used in ISO TC184/SC4. These are the basis for the continuing development of STEP, and are intended to be applicable to the development of other standards and specifications where integration or compatibility with STEP is required or desired.

Methods have been developed within the STEP committees for the following.

- **Data architecture**: underlying the STEP data models and the document architecture is a data architecture that identifies the roles of different data models within the overall context of STEP.

- **Model integration**: methods have been developed and used that allow data models to be developed independently by industry experts, and then integrated with the existing models in STEP.

- **Interpretation of application requirements**: methods have been developed that support the creation of standardized models (Application Interpreted Models) that fulfil the requirements of Application Protocols.

- **Documentation**: detailed guidelines for the preparation of STEP parts have been produced.

- **Quality assurance**: procedures and guidelines have been developed that are used to review, improve and approve STEP documents; the use of these methods ensures a high degree of consistency and quality across all parts of STEP.

Some of these methods are only applicable to the development of the standard, and are developed and maintained within the standardization committees. Others, however, may be applicable to other standardization activities, to research in industry or academia, or to system development and integration projects. Wider availability of documentation of the methods in this second category will further encourage the wide use of STEP and its methods; these aspects of the STEP architecture and methodology are being standardized as ISO 10303-13.

Requirements

The STEP methodology has been developed to meet industry requirements for standard data specifications that support:

- long term storage and retention of product information, in a form that enables continued use;
- reduction in the development and maintenance costs of interfaces between software systems;
- independence of data from the software tools which create or consume information;
- communication of product information between departments, disciplines, and enterprises.

In addition, the fact that STEP is a standard introduces additional requirements, in that the specifications developed to fulfil these needs should be stable, extensible, and publicly available.

Principles

The STEP methodology is based on a small number of fundamental principles.

1. STEP defines an architecture for product data, providing stability and extendibility.
2. STEP supports and requires traceability of data to industry needs.
3. The role of STEP is the standardization of industry application semantics.
4. STEP defines the requirements for implementation of product data exchange, based on a separation of data specifications from implementation forms.

5. STEP defines the requirements for the assessment of conformance of implementations.

These principles are the basis for the solutions provided by STEP to the industry needs articulated above.

The STEP Architecture

The architecture of STEP results from the principles stated above. The complete architecture of STEP covers all elements of the standard, including the EXPRESS data definition language (ISO 10303-11) and implementation forms such as the STEP Physical File (ISO 10303-21) and the Standard Data Access Interface (SDAI, ISO 10303-22). The complete architecture of STEP is described in the ISO 10303 Architecture and Reference Manual (ISO 10303-13).

The standard data specifications that result from the use of the STEP architecture and methodology fall into two categories:

- **Application Protocols**: data specifications that satisfy the specific product data needs of a given industrial application;

- **Integrated Resources**: generic data specifications that support the consistent development of Application Protocols across many application areas.

This distinction is reflected strongly in the structure of the STEP standard, ISO 10303, and its division into several series of parts. The following discussion, however, examines the key elements of the STEP architecture in terms of the roles that they play in satisfying industry needs. The final part of this section ties the architecture to the structure of the STEP documentation.

Within this appendix, the architecture and methodology are described as they relate to the normative data specifications that are included in STEP. Figure 31 below provides a high level summary of these elements of the STEP architecture.

The direction of the arrows in the diagram specifies “existence dependence”, ie. the object at the “tail” of the arrow is dependent on the object at its “head”.

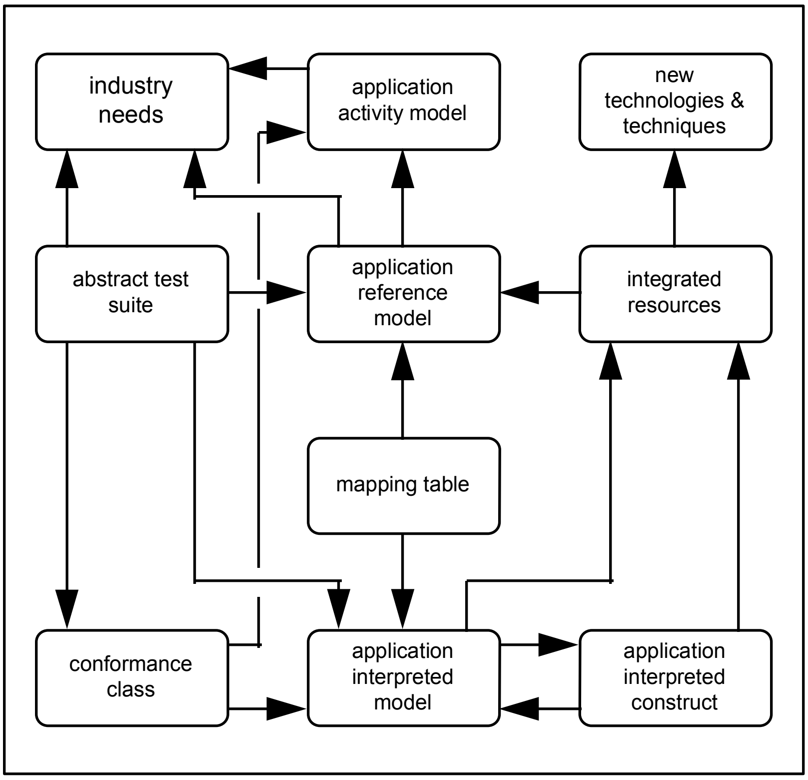

Figure 31: key elements of the STEP Architecture

Industry needs

It will be seen from Figure 31 that the key element upon which all elements of the architecture is **industry needs**. This illustrates that STEP maintains dependence of data on industry applications (what people do) together with independence from computer applications (the tools that are used by people). This linkage to industry needs means that all data that complies with STEP data specifications has an explicit, dependent link with the reason or purpose for its existence. For example, STEP does not support the representation of geometry (such as points, curves, and surfaces) without linking such a representation to a specific product, discipline, or life cycle phase. Therefore, complete information about the shape of a product can be exchanged between organizations without the need for additional communication by phone or fax to indicate the purpose of the exchange.

Application activity model

Industry needs are described by reference to an **Application Activity Model** (AAM); this model, created using activity modelling techniques such as IDEF0, supports the analysis of the activities and information flows within the scope of the industry application. Further detail analysis and design of data specifications within STEP is linked back to the “in scope” activities and information flows. It should be noted that the role of the application activity model is to capture the activities (“what is done”) within an industry application, not the detailed processes (“how it is done”), that are likely to vary between organizations, or with time as the result of continuous improvement or business process re-engineering activities.

Application reference model

The second element of the STEP architecture that results from detailed analysis of the requirements of the industrial application is the **Application Reference Model** (ARM). This is a detailed specification of the data objects (entities and attributes), and the relationships between them, that are required to support the activities within the scope of the industry application. This specification is prepared through analysis of requirements identified by experts in the industry application (sometimes referred to as “domain experts”). These requirements are therefore described using the terminology of the application, and form the basis not only for further development, but also for review and validation. As Figure 31 illustrates, the ARM is dependent on the AAM: it is a detailed description of the data that supports and flows between the activities described in the AAM.

Application interpreted model

The Application Reference Model defines the information requirements of an identified industrial application. These requirements are fulfilled by a second model: the **Application Interpreted Model** (AIM). The AIM fulfils the requirements stated in the ARM through selection and constraint of standard data constructs; this re-use of standard constructs across a wide range of industry requirements results in a high degree of consistency and integration across models, and enables potential reuse of the software code used in interfaces and the potential sharing of common data across application domains. The AIM specifies the data constructs to be used in achieving exchange of information between computer applications; the AIM is defined using the EXPRESS language, and therefore enables file-based exchange in conjunction with the STEP Physical File format, or data access using the SDAI.

Mapping table

The **mapping table** forms the link between the requirements of the industrial application (ARM) and the standard data specification that fulfils them (AIM). The mapping table describes how the standard product data constructs of the integrated resources (IRs) are used in the development of the AIM. The mapping table also includes constraints on the valid population of the AIM (ie. the permitted values of the data).

Integrated resources

The standard data constructs used in the creation of an AIM are specified in context independent models: **Integrated Resources** (IRs). The Integrated Resources logically constitute a single, conceptual model for product data, that supports the common requirements of many different product data application areas. Although the Integrated Resources are used as the basis for developing AIMs, they are not themselves intended for direct implementation: they define reusable components that are intended to be combined and refined to meet a specific need. Integrated Resources are specified using EXPRESS.

Integrated resources are developed and extended to meet two areas of need:

- the requirements contained in the Application Reference Models of Application Protocols;
- new technologies and techniques.

These represent short- and long-term drivers on the capabilities of STEP, and allow the standard to address not only the needs of users of current information technology systems, but also to anticipate the requirements that emerge as new types of system achieve market penetration and acceptance.

Application interpreted construct

The final key element of the STEP architecture is the **Application Interpreted Construct** (AIC). As is discussed below, the process of interpretation within the STEP methodology is the selection and possible constraint of integrated resource constructs to meet an identified industry need. When a common requirement is identified across two or more industry applications, an AIC may be identified as being the shared fulfilment of this requirement. An AIC explicitly identifies the potential for shared data between industry applications. Application Interpreted Constructs are specified using EXPRESS.

The role of EXPRESS

Each of the last three elements of the STEP Architecture: AIMs, IRs, and AICs, are specified using the EXPRESS language. The complete specification of each of these elements has, in fact, two elements: the data specification (in EXPRESS), and the specification of the meaning of the data (in English). The structures of the EXPRESS language themselves, of course, provide a partial definition of the meaning of the data, but the complete, unambiguous semantics are conveyed only by the combination of both specifications.

To aid understanding of the structure and inter-relationships of these models, a third element is included in their documentation: a graphical presentation of the data specification using the EXPRESS-G notation.

Conformance classes

An Application Interpreted Model, as discussed above, provides the normative specification for data to be exchanged between computer applications. This provides the scope and boundaries for implementations of product data exchange that conform to STEP, and also the scope and boundaries for testing implementations. In order to meet the needs of differing computer systems used within a given industrial application, whilst maintaining consistency of implementation and testing, two or more **Conformance Classes** may be defined for an AIM. A conformance class defines a subset of the AIM that may be used as the basis for implementation and testing. These subsets define the minimum conforming implementation based on the AIM; implementations based on any other subsets are not considered to be conforming. Conformance classes are developed on the basis of "usage scenarios" that result from analysis of the activities captured within the Application Activity Model (AAM).

Abstract test suites

The importance of testing and testability within STEP is reflected by a standardized framework and methodology for conformance testing. Although this is not within the scope of this appendix, one aspect of testing – the **Abstract Test Suite** – is the manifestation of the needs of testing within the STEP architecture. An abstract test suite specifies, in non-specific or parameterized form, the test cases that will be used in assessing the conformance of an implementation to the data specification contained in an AIM and the other elements of the STEP architecture upon which an AIM depends. Experience in other domains, such as the OSI standards for Open Systems, has shown that standardization of Abstract Test Suites is an essential prerequisite to repeatability and consistency of testing, and therefore of mutual recognition of test results across regional or national boundaries.

Standards documentation

Figure 32 below shows the relationship between the elements of the STEP architecture described above and the documentation of STEP as a standard. The elements of the architecture that are specific to an industrial application form the basis for **Application Protocols**: parts of STEP (200 series) that standardize the data specification for defined industry needs. Although Abstract Test Suites are specified for each Application Protocol they are (for historical reasons) published separately as parts in the 300 series.

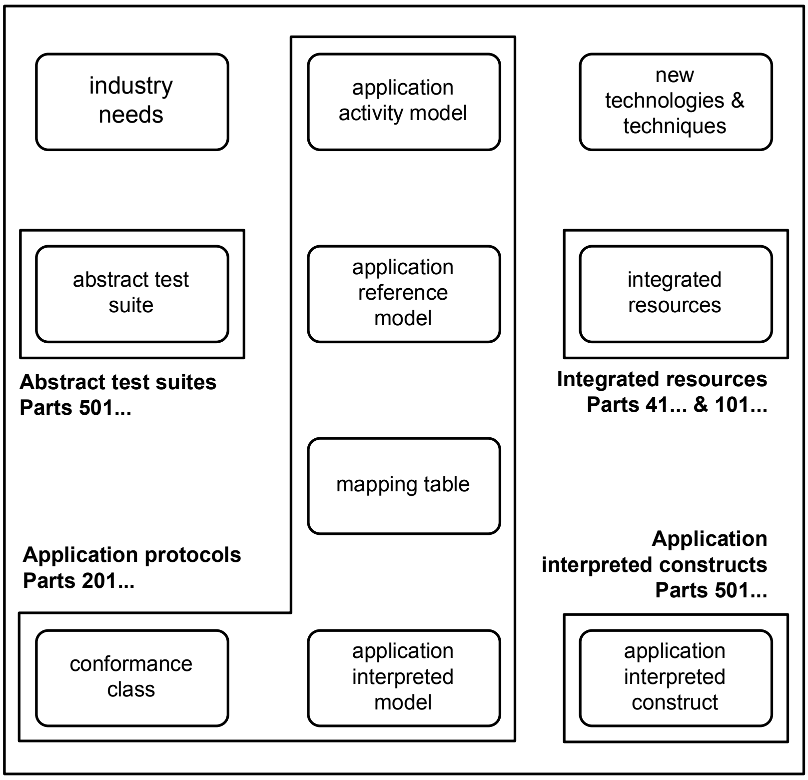

Figure 32: relationship of the STEP architecture to the documentation of the standard.

The elements of the architecture that are shared between applications are standardized either as Integrated Resources (40 series and 100 series) or as Application Interpreted Constructs (500 series).

The basis of STEP data specifications

As discussed above, the consistency of data specifications within STEP for a wide range of industry applications (Application Protocols) is ensured by the reuse of common Integrated Resources. The Integrated Resources themselves are based in a formalized framework for product data, sometimes referred to as the Generic Product Data Model (GPDM). This framework defines the basis of all the data specifications that are standardized within STEP.

Within STEP, elements of data specifications (or "constructs") are taken to be the representation of facts about objects in the real world. The basis of STEP data specifications lies in a framework for product data modelling that is based on a classification of the types of data that describe products. This classification identifies five major types of data, as follows.

- **Application context**: data that defines the purpose for which product information is created, and the types of product, disciplines, and life cycle stages for which such information is valid. The use of an application context allows data that represents an "as designed" product to be distinguished from that for as "as built" configuration.

- **Product definition**: data that identifies products, including variants and categories, and the defines life cycle "views" of products. Product definition data also includes that which relates to the structure of products, in terms of assembly structures, configurations, effectivities, or bills of materials.

- **Product property definition**: data that characterises products by their properties, independent of the representation of properties. For example, it is possible to identify the shape of an object, or aspects of the shape, as a property of the object, without providing a detailed description of shape using a CAD model or engineering drawing.

- **Product property representation**: data that represents the properties of a product, including multiple representations of the same property. For example, the shape of an object may be identified, and then described in many different ways: a 3D CAD model, a physical mock-up, an engineering drawing, and a technical illustration are *different* representations of the *same* shape.

- **Product property presentation**: data that defines the presentation of product information to support human communication. The shape of an object (the property) is *represented* by co-ordinate values, curves and surfaces; this representation is *presented* by assigning colours, and line fonts and displaying the resulting picture on a workstation.

This classification of product data is the basis for all STEP data specifications. It is the framework upon which all the Integrated Resources are built, and is reflected in the Application Interpreted Models (AIMs) of all Application Protocols. The models that capture this framework embody the principle of existence dependence, which ensures that all product information is related to an identified product and ultimately to an application context. This structure is summarized graphically in Figure 33 below.

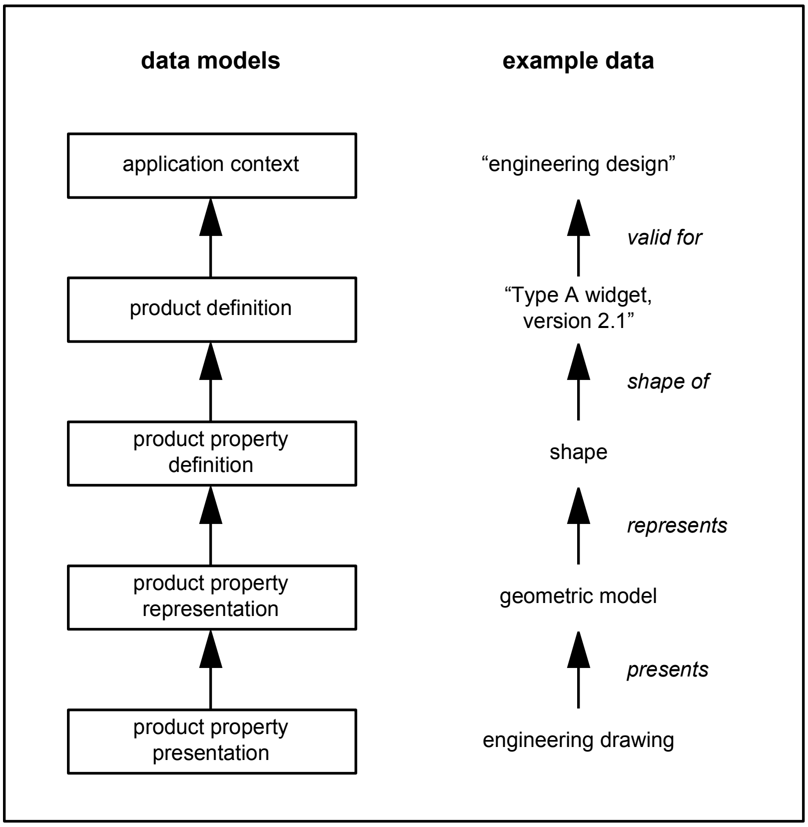

Figure 33: existence dependence of STEP models

This principle can lead to models that are at first sight counter-intuitive: rather than stating that a product having a shape, a shape is "of" a product. However, simple analysis of this example shows that the existence dependent form of the model *requires* that a shape is *always* the shape of a product. Similarly, at a lower level in the structure, STEP does not allow the existence of "geometry" data as collections of points, lines, and curves. Through the existence dependent structures in the STEP models, such a collection of geometry data *must* be the representation of some property, that is related to the definition of a product, that has validity in some application context. Thus the basic structure of the STEP models satisfies and enforces one of the principles of STEP identified above: that all product data should be traceable to an industry need.

Walk through of the STEP methods

Thus far, this appendix has discussed in some detail the architecture of STEP. This discussion is a necessary prerequisite to an exposition of the methods that are used in developing STEP, since these methods are specific to the needs of creating and maintaining each element of the STEP architecture and the relationships between elements.

The walk through presented here is essentially a "procedural" viewpoint on the methodology of STEP, concentrating on *what* is done rather than *how* it is done. Just as the description of the key elements of the STEP architecture above starts with the high level industry needs, the methodology starts at the same point. The development of STEP is driven by the need to fulfil the diverse needs of many industry sectors in a consistent, cost effective manner. The STEP development methodology therefore not only defines the process by which such needs are fulfilled, but also determines how this process is to be managed.

Definition of scope and requirements

As stated above, the STEP methodology is closely focused on the development of Application Protocols (APs), ie. standardized data specifications that satisfy identified industry needs. The development process for an Application Protocol is initiated by the identification of such a need; this may arise from collaborative projects in industry, trade associations, standards bodies, or individual companies. Some Application Protocol proposals arise from Application Protocol Planning Projects (APPPs) within the ISO STEP committees; these projects generally focus on the needs for multiple Application Protocols (or very large Application Protocols) within an industry sector.

The first stage in articulating this need is the definition, at a high level, of the scope of the proposed Application Protocol and the requirements that it is intended to fulfil. This definition not only enables validation of the proposed Application Protocol by potential users and implementors in industry, but also its assessment for overlaps and redundancies with other Application Protocols.

The basis for the definition of the scope and requirements for a proposed Application Protocol is the identification of the Industry Application Semantics that are to be standardized. This is characterized by the types of products to be described, the kind of product data used, the disciplines that make use of the product data, and the life cycle stages in which the data is created and used. This characterization is extended through the definition of a Application Activity Model (AAM), that elaborates the activities and information flows that are to be in the scope of the Application Protocol. Although other methods for activity modelling are permitted, the IDEF0 methodology is almost universally used in STEP. The formal specification of this activity model is often accompanied by the formulation of usage scenarios: informal (but detailed) descriptions of the intended use of the Application Protocol.

At this stage in the development of an Application Protocol, a high level data model (or "data planning model") may be produced as an aid to understanding and analysis of the scope of data to be supported. Such a model attempts to capture the subject areas, or major groups of data, that are in scope. No specific method for the development of such models is mandated; most are documented using simple, informal graphical presentations (often similar to those in Figure 31, Figure 32, and Figure 33 above).

On the basis of the initial statement of scope and requirements, the proposed application protocol is submitted as a Preliminary Work Item Proposal under ISO rules; if approved, the initial development of the Application Protocol as a part of STEP is mandated.

Information requirements

The second phase in the development of an Application Protocol is the discovery and documentation of the detailed information requirements that are to be fulfilled. It is important to note that these requirements are *discovered*, rather than defined: the requirements already exist as the data that underlies industry practices, processes, and systems. These requirements are analysed and documented through the development of an Application Reference Model (ARM). The term "ARM" may be used to refer to two elements of the Application Protocol:

- English language statements of the information requirements, in the form of defined application objects, attributes, and relationships; these application objects may be grouped into Units of Functionality (UOFs);

- a graphical presentation of these requirements, using notations such as IDEF1X, NIAM, or EXPRESS-G.

The development of an ARM, and the specification of information requirements, is one of the weaker elements of the STEP methodology, in that the guidance provided to Application Protocol development teams is little more than that presented above. Significant advances are, however, being made in this area, particularly within projects that are addressing a broad spectrum of requirements within an industry sector. Improved techniques for ARM development are being employed in Application Protocol projects in the automotive, process plant, shipbuilding, and building & construction sectors; harmonization and acceptance of these techniques is likely to lead to their incorporation into the “core” methodology of STEP.

Once the ARM is complete to the satisfaction of the Application Protocol development team, full documentation of the scope (AAM) and requirements (ARM) are distributed as a New Work Item Proposal; this process is designed to ensure adequate and effective review and validation of the Application Protocol by experts in industry. If approved, the Application Protocol is allocated a part number within ISO 10303; at this point, the ISO procedures of standardization require that a Committee Draft (CD) should be produced within two years, and a Final Draft International Standard (FDIS) within three years.

Application interpretation

The first two phases in the specification of an Application Protocol are undertaken by the project team responsible for its development. From this point onwards, however, the further development of the Application Protocol is undertaken through synergy between the project team and the “core” functions of STEP: AIM Development, AP Integration, Resource Integration, and ATS Development. This interaction may be seen as part of a “matrix” management approach to the development of the standard: each Application Protocol results from the definition of requirements by industry or application experts, the fulfilment of those requirements by “STEP” experts, and the validation of solutions by the industry experts.

The first of these synergistic phases is the development of the Application Interpreted Model (AIM – see above): the creation of a data specification based on the STEP Integrated Resources that meets the requirements stated in the

Application Reference Model. This phase begins with analysis of the Application Reference Model by the AIM Development team: this analysis focuses on gaining deep understanding of the application requirements, and relating this understanding to the underlying concepts of the STEP Integrated Resources, as outlined above.

A second part of this analysis, undertaken by the AP Integration team, is the identification of overlaps with other, existing Application Protocols. Where such overlaps correspond to shared requirements across two or more Application Protocols, the development and use of Application Interpreted Constructs, ie. a shared solution to the common requirements, is enabled.

The third aspect of this analysis is the identification of requirements that are not supported by the STEP Integrated Resources, and therefore give rise to a need for extension to the Integrated Resources. It is an important principle of the STEP methodology that Application Protocols do not themselves define extensions to the resource models. These extensions are developed according to the Resource Integration method, as discussed below.

Mapping

Following these analyses, the process of application interpretation involves the identification of the mapping from each information requirement (application object, attribute, or relationship) to one or more constructs from the Integrated Resources. This mapping results in the creation of two elements of the documentation of an Application Protocol:

- the Mapping Table, that specifies the precise mapping of each application requirement;
- the Application Interpreted Model (AIM).

As each application requirement is mapped, the result of the mapping is incorporated into the AIM: the interpretation process results in the creation of a *new* data model (EXPRESS schema) from the Integrated Resources. Within this new data model, each construct acquires the context of the Application Protocol; in many cases, the requirements of this context are fulfilled by applying constraints to the constructs mapped from the Integrated Resources. However, even when a constraint is not explicitly specified, this additional contextual information means that an entity definition in an AIM is *not* the same as an apparently identical definition in the Integrated Resources. The definition in the AIM represents a *usage* of the resource construct, and refines its meaning for the context of the Application Protocol.

Application interpreted model

An Application Interpreted Model is specified as a "short form" EXPRESS schema: this consists of the EXPRESS interface statements that select constructs from the Integrated Resource schemas, together with the additional specializations and constraints that are defined by the mapping process. A second specification (the "annotated listing") provides refinements to the natural language definitions of the constructs selected from the Integrated Resources.

The schema interfacing capabilities of EXPRESS mean that, through the use of suitable software tools, all references in the AIM short form can be resolved to create a single schema form of the AIM, known as the "long form". Algorithmically:

```
Short form + Integrated Resources --> Long form
```

This long form AIM is provided in electronic form as part of the Application Protocol documentation, and is the basis for implementations based on the STEP Physical File format.

Application interpreted constructs

The potential for creation and use of Application Interpreted Constructs (AICs) is described above as part of the initial phase of the application interpretation process. AICs arise where different applications share functional requirements, which may in some cases relate to the use of common or similar computer systems; many of the AICs identified to date relate to common uses of geometric representation (B-Rep, Surface models, etc.), across different application areas. For example, if automotive design, sheet metal tooling, and shipbuilding all use surface models to represent product shape, then a common AIC for this functional requirement can be developed and used.

The existence of an AIC identifies the *potential* for reuse of implementation code, and for the sharing of data between applications. This latter point is particularly important: since the data specification for surface models is common across the three applications identified above, data instances may be shared between them. However, just because they may be shared does not necessarily mean that it is *useful* to do so. In this example, it is easy to see that the same surface model might be used by the automotive design and sheet metal tooling applications; similar sharing between automotive design and shipbuilding is less likely to be useful.

The method for the development of Application Interpreted Constructs is broadly similar to that for Application Interpreted Models, ie. the selection and constraint of Integrated Resource constructs. The only significant difference is that each AIC includes one or more "root" entity, that carries the constraints relevant to the AIC; a root entity acts as a scoping mechanism for the applicability of these constraints when the AIC is used within an Application Interpreted Model. The method for use of Application Interpreted Constructs is simple: an AIC is used through inclusion, without modification or constraint, within an AIM. It is this lack of modification or constraints in the use of an AIC that ensures compatibility across the AIMs that use an AIC.

Resource integration

As discussed above, analysis of the requirements specified in an Application Protocol may identify the need for extension to the STEP Integrated Resources. This approach to Integrated Resources represents the "mature" phase of STEP development; previously, complete, existing models proposed as STEP resources have been "integrated" with the core Generic Product Data Model; it is this "creation" phase that has given rise to the Integrated Resources as they are today. As with AIM development, resource integration is a synergistic process, involving the interaction between the experts in the discipline covered by the resource model (geometry, finite element analysis, etc.), and the STEP integration team.

The creation phase requires a more involved and complex method, since the requirements for extension to the Integrated Resources have been large, and often expressed as complex, mature EXPRESS data models. However, where these models have been developed without the use of the basic STEP architecture, considerable restructuring of these "Draft Resource Models" has proved necessary.

The resource integration method involves several phases; these may be summarized as follows:

- **analysis**: comparison of the requirements underlying the draft resource model (or those identified within an Application Reference Model) with the concepts of the STEP Integrated Resources; where a draft resource model exists, this analysis also includes comparison of specific data model structures;

- **restructuring**: where a draft resource model is the source of the requirements for extensions, this model is restructured so that it fits semantically and structurally with the existing Integrated Resources, and conforms to the EXPRESS usage practices adopted within the integration process;

- **verification**: the model that results from the integration process is verified by appropriate application and discipline experts to ensure that requirements are accurately and completely fulfilled.

Where the requirements for extension to the Integrated Resources arise from the application interpretation process, the restructuring phase is trivial, since the extension is designed and created by the integration team itself (ie. in this case “restructuring” could be replaced by “creation”).

Requirements for implementation and testing

The various methods described above relate to the creation of data specifications within STEP. It must not be forgotten, however, that these specifications are useful only as the basis for implementation of data exchange or sharing, and that such implementations are required to be testable. Requirements for implementation and testing and fulfilled through the specification of conformance classes within an Application Protocol, and of an Abstract Test Suite for the Application Protocol.

Conformance classes

Conformance classes are developed through analysis of the usage scenarios identified in the initial phase of the development of the Application Protocol, and through understanding of the capabilities of the computer applications that are expected to support the Application Protocol. Each conformance class defines a fixed boundary for the scope of an implementation; this is determined on the basis of defining a subset of total capability of the Application Protocol that is practical to implement whilst not comprising the industry needs that define the purpose for the existence of the Application Protocol.

Abstract test suites

Abstract Test Suites (ATS) are developed through analysis of the requirements specified in the Application Reference Model and the Application Interpreted Model of an Application Protocol. An ATS has several constituent components:

- test purposes: formal statements of the aspects of an Application Protocol; these are derived directly from the requirements specified in the Application Protocol;

- verdict criteria: the basis for determination of success, failure, or uncertainty with respect to the results of testing;
- abstract test cases: parameterized forms of simple (yet representative) test cases that exercise one or more test purposes.

Initial Abstract Test Suites are to be published for review during 1995.

The role of ARM and AIM in implementation

In the discussion of STEP implementation in Chapter 6, it was assumed that a mapping may be identified between the internal data structures of a CAx system and the data model contained in the STEP Application Protocol on which the implementation is based.

However, as discussed in the earlier sections of this appendix, each STEP Application Protocol contains *two* different data models: the ARM and the AIM. Although there is no standard approach to implementation defined for STEP, the fact that the ARM is stated in the terms and terminology of the industrial application means that it is often easier to map the internal structures of the CAx system to the ARM rather than to the AIM. Several developers of STEP interfaces, or of tool kits that support the development of such implementation, have made use of this approach. This is particularly attractive where a developer is producing interfaces to the same application protocol for a number of different CAx systems: the mapping from the ARM to the AIM (as specified in the Mapping Table) is encoded within the translator software, which can then be linked through ARM concepts to each application that is to be supported.

Approval and publication

The completion of an Application Protocol, Integrated Resource, or Abstract Test Suite initiates the formal processes of review and approval as ISO Committee Drafts (CD) and Draft International Standards (DIS). Response to comments raised during these reviews gives rise to iterative application of the methods outlined above.

Conclusions

The STEP development methodology governs the development and standardization of data specifications which, when combined with STEP implementation forms, are suitable for neutral file exchange as well as providing the basis for shared product databases and archiving. The methodology is designed to fulfil a number of high level industry requirements, and is based on a number of fundamental principles that in turn give rise to the architecture of STEP described above.

The methods by which the various elements of the architecture are created constitute a complete, proven methodology for the creation and maintenance of standard data specifications. This methodology is well established and understood within the STEP development activity and, on the basis of the comprehensive documentation currently in preparation within ISO TC184/SC4/WG10, offers the opportunity to other standardization and industry projects to achieve the same benefits as those gained by STEP, in a manner that is fully consistent with STEP.

Appendix B: EXPRESS: An Overview

This appendix provides a brief overview of the EXPRESS data specification language, ISO 10303-11. As described in Chapter 5 and Appendix A, EXPRESS is not only a part of STEP, but is used to define the normative data specifications contained in STEP Application Protocols, Integrated Resources, and Application Interpreted Constructs.

It is *not* intended that this appendix should form a tutorial in the EXPRESS language: rather, it is provided to give an initial "feel" for the language and its capabilities to readers intending to develop, implement, review or use data specifications written in EXPRESS. Readers seeking additional information are referred to Wilson & Schenk's introductory text on EXPRESS, or to the many tutorials and courses that are available from STEP Centres and other sources of expertise.

Overview

EXPRESS is a textual data specification language. It is based on the Entity-Attribute-Relationship of data, includes generalization and constraint specifications, and embodies some "object oriented" characteristics. The standard version of EXPRESS is ISO 10303-11:1994. Software tools that support the use of EXPRESS (see Chapter 10) mostly conform to this version of the language, although some may refer to earlier versions, or may provide support for language extensions.

One of the key aspects of EXPRESS is that it is both computer-interpretable and human-readable. It conforms to a formal syntax, so that models can be validated and processed by computer software, but can also be presented to a human reader or reviewer in a form that allows a data specification to be readily understood. This latter aspect of EXPRESS is supported by the existence of a graphical subset of the language (EXPRESS-G) that allows data specifications to be developed and reviewed in pictorial form. There is also a related instance language (EXPRESS-I, ISO 10303-12).

In the context of STEP, EXPRESS is designed for conceptual product data modelling, and supports the STEP goals of mapping to multiple implementation forms and programming languages. However, EXPRESS has also gained support in many other activities: other standardization work, such as POSC and EDIF, and many industrial, research, and academic projects. An annual EXPRESS Users' Group conference provides a forum for this broad spectrum of language users and developers to share ideas and experiences.

Even before the publication of EXPRESS as an International Standard in 1994, proposals were being developed for a second, extended edition of the language. A summary of these proposals is provided at the end of this appendix.

Before examining some of the capabilities of EXPRESS, it is worth stating some of the things that EXPRESS is *not*.

- EXPRESS is not a methodology. It is sufficiently flexible to be used in conjunction with many different methodologies, each of which may introduce constraints on the usage of the language.

- EXPRESS is not a *complete* information modelling language. Typically, an information model or data specification consists of natural language definitions and EXPRESS data definitions, together with EXPRESS-G graphical presentations.

- EXPRESS is not a programming language. It has no execution model (ie. even though EXPRESS can be compiled, it cannot be "run"). It is, however, mappable to the data structure capabilities of many programming languages.

EXPRESS constructs

This section provides a short survey, with examples, of the basic constructs and capabilities of EXPRESS.

Examples

Readers should note that the examples included here have been chosen to illustrate the features of EXPRESS as simply as possible. They are not extracts from "real" data models, do not follow any particular methodology or modelling style, and are not even intended to be "good" models. Indeed, they fail the criterion for completeness given above, in that they do not provide natural language definitions for each construct. The accompanying commentary and the use of appropriate names for types and entities should, however, allow the intent of each example to be clear. The syntax of all examples has all been validated using an EXPRESS compiler tool.[1]

Schema

EXPRESS supports the definition of modular data models, by partitioning the complete model into schemas.[2] Every model consists of one or more schemas, each defining a common scope for a collection of data definitions. Inter-schema interfacing allows different components of large data models to be developed separately; however, there is no concept of public and private specifications, such as is provided in many object-oriented programming and data definition languages.

Type

The basic "primitives" of EXPRESS are its base types such as REAL, INTEGER, STRING, and BOOLEAN. EXPRESS also supports the creation of "defined" types, such as enumerations (lists of values) or selections between diverse types.

Entity

The entity is the basic unit of data definition within EXPRESS. Generalization and specialization of entity types is supported, with multiple inheritance. EXPRESS is unusual by comparison to many other data modelling languages in that it does not make a distinction between attributes and relationships. In EXPRESS, an attribute is regarded as defining the role played by a base type or defined type in the definition of an entity; a relationship is similarly regarded as the role played by one entity type in the definition of another.

The following example illustrates the use of entities and attributes to define a simple data model.

```
SCHEMA example1;

ENTITY car;
  make      : STRING;
  car_model : STRING;
  year      : INTEGER;
  owner     : person;
END_ENTITY;

ENTITY person;
  first_name : STRING;
  last_name  : STRING;
END_ENTITY;
```

```
END_SCHEMA;
```

Note that the relationship between **car** and **person** uses a similar syntax to that of the attribute **make** or **car_model**. The latter is used as the attribute name since ‘model’ is one of the reserved words of the EXPRESS language.

The following example illustrates the use of defined types and enumeration types.

```
SCHEMA example2;

REFERENCE FROM example1 (person);

TYPE identifier = STRING;
END_TYPE;

TYPE licence_type = ENUMERATION OF
  (provisional,
   full);
END_TYPE;

ENTITY driving_licence;
  driver         : person;
  licence_number : identifier;
  validity       : licence_type;
END_ENTITY;

END_SCHEMA;
```

The inter-schema interfacing construct REFERENCE is used to access the definition of the **person** entity in example 1. A **driving_licence** is defined by a reference to the driver (person), the licence number, and the validity. The enumeration type states that the only possible values of validity are provisional or full.

Subtypes and supertypes

EXPRESS supports generalization/specialization relationships between entity types. If one entity type is defined to be a subtype of another, then it inherits all its properties, ie. its definition, attributes and constraints. Multiple inheritance is supported, where one entity type is defined as a subtype of two or more other entity types. The resulting generalization/specialization lattice for entities supports the definition of complex entity instances, combining the characteristics of several “parent” entity types. When an entity type has more than one defined subtype, the default relationship is that the subtypes may be instantiated independently or together. This may be constrained using the SUPERTYPE expression.

This example illustrates the inheritance of attributes from a “parent” entity to its “children”.

```
SCHEMA example3;

ENTITY vehicle
SUPERTYPE OF (ONEOF (car, truck));
  no_of_wheels : INTEGER;
END_ENTITY;

ENTITY car
SUBTYPE OF (vehicle);
  model_name : STRING;
END_ENTITY;

ENTITY truck
SUBTYPE OF (vehicle);
  load_limit : REAL;
END_ENTITY;

END_SCHEMA;
```

The **car** and **truck** entity types both inherit the attribute **no_of_wheels** from the parent **vehicle** entity type. The SUPERTYPE expression constrains the combinations of subtypes: in this model, a **vehicle** may be a **car** or a **truck**, but not both. Without the SUPERTYPE expression, this combination would be permitted by the model.

The following example shows how an entity type can inherit characteristics from more than one parent.

```
SCHEMA example4;

REFERENCE FROM example3 (truck);

TYPE propulsion_type = ENUMERATION OF
  (water_jet,
```

```
    screw);
END_TYPE;

ENTITY boat;
  propulsion : propulsion_type;
END_ENTITY;

ENTITY amphibian
SUBTYPE OF (truck, boat);
END_ENTITY;

END_SCHEMA;
```

An **amphibian** is defined as being both a boat and a truck, and inherits the characteristics of both. The attributes of **amphibian** are therefore **no_of_wheels** (inherited from **vehicle**), **load_limit** (inherited from **truck**), and **propulsion** (inherited from **boat**).

Aggregations

EXPRESS provides a number of constructs that define aggregations or collections. The four variations on aggregations are as follows:

- SET: an unordered collection of elements, all of which are different;
- ARRAY: an ordered, indexed collection of elements, which may or may not be different;
- LIST: an ordered collection of elements, which may or may not be different;
- BAG: an unordered collection of elements, which may or may not be different.

The following example illustrates the use of BAG and LIST aggregations.

```
SCHEMA example5;

ENTITY part;
  id : INTEGER;
END_ENTITY;
```

```
ENTITY kit;
  parts : BAG [1:?] OF part;
END_ENTITY;

ENTITY instruction;
  description : STRING;
END_ENTITY;

ENTITY assembly_of_kit;
  steps : LIST [1:?] OF instruction;
END_ENTITY;

END_SCHEMA;
```

This example represents a plastic kit of an aeroplane. The kit itself is represented as a BAG of parts. There is no order to the parts, but the same type of part may occur more than once. The instructions to make up the kit have to followed in the correct, sequential order, so a LIST is used.

Select types

The SELECT construct is used to identify places where different types can play the same role. There need not be anything in common between the types.

```
SCHEMA example6;

ENTITY overhead_projector;
  power_rating : REAL;
END_ENTITY;

ENTITY flipchart;
  page_size : STRING;
END_ENTITY;

TYPE visual_aid = SELECT
  (overhead_projector,
   flipchart);
END_TYPE;

ENTITY presentation;
  medium : visual_aid;
END_ENTITY;
```

```
END_SCHEMA;
```

In this model, an overhead projector and a flip chart have no characteristics in common. However, both can be used to help in giving a presentation. In this example, the SELECT type **visual_aid** denotes that either a **overhead_projector** or a **flipchart** can play the role of **medium** for a **presentation**.

Derived attributes

In many cases, there are characteristics or properties of things that can be calculated from other characteristics. For example, a person's age can be calculated if the current date, and the date of birth, are known. EXPRESS provides a capability to define a attribute as being derived from the values of one or more other attributes.

The following example shows the use of a derived attribute for the area of a circle. Note that these are *not* the STEP definitions of point or circle!

```
SCHEMA example7;

ENTITY point;
  x : REAL;
  y : REAL;
END_ENTITY;

ENTITY circle;
  centre : point;
  radius : REAL;
DERIVE
  area   : REAL := PI * radius * radius;
END_ENTITY;

END_SCHEMA;
```

The **area** of the circle can be calculated from the values of the explicit attributes. It is specified using the "built-in" constant PI (π). The algorithm that evaluates the value of a derived attribute may be specified in a separately defined function.

Optional attributes

Not every attribute of an entity is required. EXPRESS supports this requirement by allowing attributes to be identified as optional. If an attribute is so identified, this means that:

- the value of the attribute need not be specified;
- the presence or absence of the value does not affect the meaning of the entity.

This example illustrates an *incorrect* use of an optional attribute.

```
SCHEMA example8;

ENTITY point;
  x : REAL;
  y : REAL;
  z : OPTIONAL REAL;
END_ENTITY;

END_SCHEMA;
```

This is incorrect since the presence or absence of a z co-ordinate determines whether the point is treated as two dimensional or three dimensional. OPTIONAL should always be read as “don’t care” to avoid this problem[3].

This example illustrates a correct use of an optional attribute.

```
SCHEMA example9;

REFERENCE FROM example2 (identifier);

ENTITY person;
  id         : identifier;
  first_name : STRING;
  last_name  : STRING;
   nick_name : OPTIONAL STRING;
END_ENTITY;

END_SCHEMA;
```

Here, the use of an optional attribute is correct as the knowledge of someone’s nick-name is not necessary to know who the person is.

Unique attributes

It is common to find circumstances in which one attribute, or a combination of attributes, are required to be unique across some collection of data. EXPRESS supports this requirement through the specification of uniqueness constraints. These can apply to a single attribute or to a combination of attributes.

```
SCHEMA example10;

ENTITY part;
  name        : STRING;
  description : STRING;
  part_number : INTEGER;
UNIQUE
  UR1 : part_number;
END_ENTITY;

ENTITY employee;
  name           : STRING;
  payroll_number : INTEGER;
  department_id  : INTEGER;
UNIQUE
  UR1 : payroll_number, department_id;
END_ENTITY;

END_SCHEMA;
```

In this example, each **part** must have a unique **part_number**. Note that EXPRESS does not define the "scope" of uniqueness; ie. the **part_number** may be unique within a given file or database, or within a company, or even within the entire universe! It is the responsibility of the developer of a data model to specify the scope within which the uniqueness applies.

In the second case, an employee is identified by a **payroll_number** and a **department_id**. This means that two employees in the same department may not have the payroll number 1234; however, two employees in *different* departments may have this number.

The label 'UR1' assigned to the UNIQUE constraint is provided to allow error reporting in an implementation.

Inverse attributes

The representation of relationships in EXPRESS appears to be one-directional. In fact, this is not the case: *all* relationships are essentially bi-directional. EXPRESS always makes one "half" of a relationship explicit.

```
SCHEMA example11;

ENTITY part_in_assembly;
  part_number : INTEGER;
  part_of     : assembly;
END_ENTITY;

ENTITY assembly;
  assembly_number : INTEGER;
END_ENTITY;

END_SCHEMA;
```

Here, a relationship is defined between a **part_in_assembly** and an **assembly**. There is, however, *always* an implicit reverse relationship, which is that an **assembly** is related to a SET of zero, one or many instances of **part_in_assembly**. This reverse relationship can be made explicit through the EXPRESS INVERSE attribute capability. This does *not* create a new relationship, it just makes the relationship visible by giving it a name, and therefore allows it to be constrained.

This modification to the previous example makes the relationship between **assembly** and **part_in_assembly** explicit, and constrains it such that there are always two or more **part_in_assembly** instances for each **assembly**.

```
SCHEMA example12;

ENTITY part_in_assembly;
  part_number : INTEGER;
  part_of     : assembly;
END_ENTITY;

ENTITY assembly;
  assembly_number : INTEGER;
INVERSE
  parts : SET [2:?] OF part_in_assembly FOR part_of;
END_ENTITY;

END_SCHEMA;
```

Local constraints

The final capability of EXPRESS to be considered within this brief survey is that of the definition of "local" constraints. We have already seen two special types of local constraint: UNIQUE and INVERSE. A more general capability allows the specification of constraints that apply to every instance of a type. These are referred to as "local" constraints as the constraint is specified within the definition of the type to which the constraint applies.

```
SCHEMA example13;

REFERENCE FROM example3 (vehicle);

ENTITY another_car
SUBTYPE OF (vehicle);
  car_model: STRING;
WHERE
  WR1: no_of_wheels >= 3;
END_ENTITY;

END_SCHEMA;
```

In this example, the entity **another_car** constrains the value of the attribute **no_of_wheels** (inherited from **vehicle**) to be greater than or equal to three. This is a very simple type of local constraint: an examination of "real" EXPRESS models (particularly those contained within STEP Application Protocols) will quickly show that many other, more complex constraints are possible. All, however, follow the basic pattern of a WHERE clause consisting of one or more local rules, each of which is a LOGICAL expression that evaluates to TRUE, FALSE or UNKNOWN.

Other features of EXPRESS

This overview has presented on the "basics" of the EXPRESS language. There are many other features and constructs that contribute to the power and flexibility of EXPRESS, including:

- algorithmic units: FUNCTION, PROCEDURE, RULE;
- standard constants, functions, and procedures;
- local and global constraints;

- schema interfacing: USE, REFERENCE.

The complexity of the EXPRESS language may be construed from the fact the EXPRESS language reference manual (ISO 10303-11:1994) is more that 200 pages long, and that the language specification includes 122 different keywords and 318 syntax productions.

EXPRESS-G

As stated in the overview at the start of this appendix, there is a graphical notation for a subset of EXPRESS, called EXPRESS-G. EXPRESS-G is intended for human communication: it was originally designed for documentation, ie. to present the results of a data modelling activity in a form more easily understood and assimilated by reviewers. However, EXPRESS-G has also proved to have significant utility in model *development*, and is now supported by a number of software tools, many of which support conversions between the lexical (text) and graphical forms of the language.

EXPRESS-G is defined in an annex of the EXPRESS language reference manual. Although it does not support all the features of the language, it does allow for cross-referencing between schemas and for multi-page diagrams for a single schema. This is fortunate, since EXPRESS-G is a generally “verbose” language, and requires more diagrams for a given model than other graphical languages such as IDEF1X, E-R, or NIAM.

Figure 34 below illustrates the basic elements of the EXPRESS-G language.

EXPRESS-G constructs

EXPRESS-G supports the following subset of the EXPRESS language.

- SCHEMA level diagrams;
  - schemas and inter-schema links only.

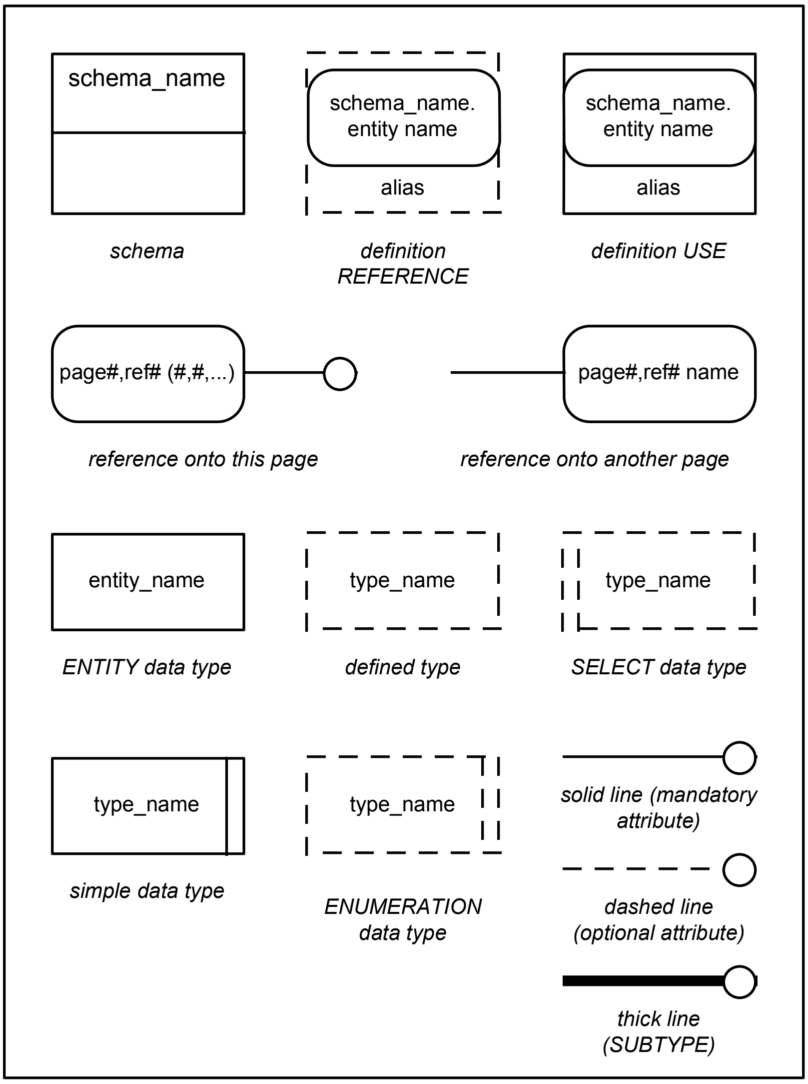

Figure 34: EXPRESS-G – graphical syntax

- ENTITY level diagrams;
  - references to definitions in other schemas;
  - multi-page references.
- ENTITY definitions.

- TYPE definitions.
- Attributes.
- Relationships and cardinalities.
- "Shorthand" for the presence of constraints.

Figure 35 is an EXPRESS-G presentation of the following example.

```
SCHEMA example14;

TYPE car_model = ENUMERATION OF
  (Mondeo,
   Cavalier);
END_TYPE;

ENTITY vehicle
SUPERTYPE OF (ONEOF (car, truck));
  chassis_number : INTEGER;
UNIQUE
  UR1: chassis_number;
END_ENTITY;

ENTITY car
SUBTYPE OF (vehicle);
  car_type : car_model;
END_ENTITY;

ENTITY truck
SUBTYPE OF (vehicle);
  load_limit   : REAL;
  CB_call_sign : OPTIONAL STRING;
END_ENTITY;

END_SCHEMA;
```

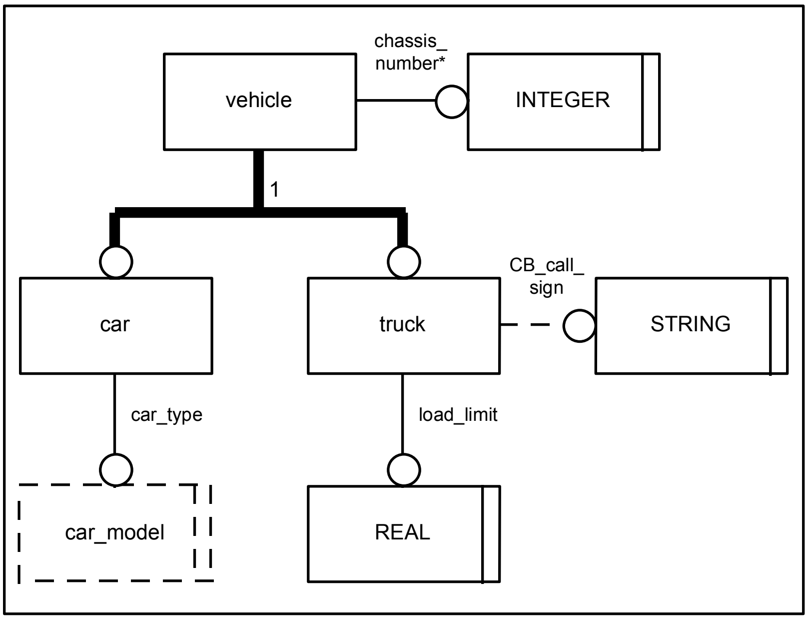

Figure 35: EXPRESS-G – example

The future of EXPRESS

As noted at the start of this appendix, work on a second edition of EXPRESS was initiated before the publication of the first edition as an ISO standard. A series of workshops were held starting in 1992, culminating in the approval of a project within ISO TC184/SC4 to develop extensions to the language. The results of this project are expected to be available for use by 1996-97.

The work on the second edition of EXPRESS covers two areas. The first area is that of revisions within the scope of the published edition of the standard. These revisions result from practical experience of the use of EXPRESS in STEP, POSC, EDIF, and other projects. The major areas of revision are anticipated to be:

- improved schema interfacing;
- definition and use of "templates";
- a rationalized type structure based on a formal "meta-model" of the language.

The second area of work covers extensions to the functionality of EXPRESS. There are many proposals for such extensions, covering domains such as:

- modelling of processes;
- modelling of events, activities, and transactions;
- more complete object-orientation, including the definition of "methods";
- mapping between different models;
- an execution model, allowing EXPRESS to be used as a "fourth generation" language for applications development.

Proposals, in the form of extensions to the syntax and semantics in each of these areas, have been developed in research and development projects. Most are identified by appending a single letter to the name of the language: EXPRESS-C, EXPRESS-M, EXPRESS-P, etc. The existence of these diverse proposals raises the issue whether the future of EXPRESS is as a single, even larger language, or as a "family" of related languages, each addressing a different set of requirements.

A further influence in the continued development of EXPRESS is the language activities of standards such as KIF (Knowledge Interchange Format), CDIF (CASE Data Interchange Format), CSMF (Conceptual Schema Modelling Facility), and SQL-3 (Structured Query Language). Many of these other activities are undertaken within ISO/IEC JTC1 "Information technology". As EXPRESS is developed further beyond the scope of conceptual product data modelling, it is possible that the continuation of the work will be undertaken within JTC1 rather than in SC4.

---

[1] The tool used was the EXPRESS/P21 Analyzer developed by the National Institute of Standards and Technology, Version V2.11.4-beta.

[2] It is normal in discussion of EXPRESS and other data modelling languages to use "schemas" as the plural of "schema", rather than "schemata". The EXPRESS language reference manual itself (ISO 10303-11) is a notable exception to this rule!

[3] More adventurous readers may wish to consider a more correct model for points. One possible answer is the **cartesian_point** entity defined in Part 42 of STEP.

Appendix C: Further Information about STEP

This appendix lists all parts of the STEP (ISO 10303) and P-LIB (ISO 13584) in development at the time of publication. It also describes the structure of the ISO committees responsible for the development of STEP.

Standards in development

The following lists all the standards approved for development or publication within SC4, as at June, 1995. Readers are advised to consult their national standards bodies, or make use of the STEP On-line Information Service (see Appendix D), to determine the current list and status. All STEP (ISO 10303) parts have the general title “Product data representation and exchange”.

ISO 10303-1	Overview and fundamental principles
ISO 10303-11	Description methods: The EXPRESS language reference manual
ISO 10303-12	Description methods: The EXPRESS-I language reference manual
ISO 10303-13	Description methods: Architecture and methodology reference manual
ISO 10303-21	Implementation methods: Clear text encoding of the exchange structure
ISO 10303-22	Implementation methods: Standard data access interface
ISO 10303-23	Implementation methods: C++ language binding to SDAI
ISO 10303-24	Implementation methods: C language binding to SDAI

ISO 10303-25	Implementation methods: FORTRAN language binding to SDAI
ISO 10303-31	Conformance testing methodology and framework: general concepts
ISO 10303-32	Conformance testing methodology and framework: Requirements on testing laboratories and clients
ISO 10303-33	Conformance testing methodology and framework: Abstract test suites
ISO 10303-34	Conformance testing methodology and framework: Abstract test methods for Part 21 implementations
ISO 10303-35	Conformance testing methodology and framework: Abstract test methods for Part 22 implementations
ISO 10303-41	Integrated generic resources: Fundamentals of product description and support
ISO 10303-42	Integrated generic resources: Geometric and topological representation
ISO 10303-43	Integrated generic resources: Representation structures
ISO 10303-44	Integrated generic resources: Product structure configuration
ISO 10303-45	Integrated generic resources: Materials
ISO 10303-46	Integrated generic resources: Visual presentation
ISO 10303-47	Integrated generic resources: Shape tolerances
ISO 10303-48	Integrated generic resources: Form features
ISO 10303-49	Integrated generic resources: Process structure and properties
ISO 10303-101	Integrated application resources: Draughting

ISO 10303-103	Integrated application resources: Electrical and electronic connectivity
ISO 10303-104	Integrated application resources: Finite element analysis
ISO 10303-105	Integrated application resources: Kinematics
ISO 10303-106	Integrated application resources: Building construction core model
ISO 10303-201	Application protocol: Explicit draughting
ISO 10303-202	Application protocol: Associative draughting
ISO 10303-203	Application protocol: Configuration controlled design
ISO 10303-204	Application protocol: Mechanical design using boundary representation
ISO 10303-205	Application protocol: Mechanical design using surface representation
ISO 10303-207	Application protocol: Sheet metal die planning and design
ISO 10303-208	Application protocol: Life cycle product change process
ISO 10303-209	Application protocol: Design through analysis of composite and metallic structures
ISO 10303-210	Application protocol: Electronic printed circuit assembly, design and manufacture
ISO 10303-211	Application protocol: Electronics test diagnostics and re-manufacture
ISO 10303-212	Application protocol: Electrotechnical plants
ISO 10303-213	Application protocol: Numerical control process plans for machined parts

ISO 10303-214	Application protocol: Core data for automotive mechanical design
ISO 10303-215	Application protocol: Ship arrangement
ISO 10303-216	Application protocol: Ship moulded forms
ISO 10303-217	Application protocol: Ship piping
ISO 10303-218	Application protocol: Ship structures
ISO 10303-220	Application protocol: Printed circuit assembly manufacturing planning
ISO 10303-221	Application protocol: Functional data and their schematic representation for process plant
ISO 10303-222	Application protocol: Exchange of design engineering to manufacturing for composite structures
ISO 10303-223	Application protocol: Exchange of design and manufacturing information for cast parts
ISO 10303-224	Application protocol: Mechanical products definition for process planning using form features
ISO 10303-225	Application protocol: Structural building elements using explicit shape representation
ISO 10303-226	Application protocol: Ships mechanical systems
ISO 10303-227	Application protocol: Plant spatial configuration
ISO 10303-228	Application protocol: Building services: Heating, ventilation and air conditioning
ISO 10303-230	Application protocol: Building structural frame: Steel work
ISO 10303-301	Abstract test suite: Explicit draughting

ISO 10303-302	Abstract test suite: Associative draughting
ISO 10303-303	Abstract test suite: Configuration controlled design
ISO 10303-304	Abstract test suite: Mechanical design using boundary representation
ISO 10303-305	Abstract test suite: Mechanical design using surface representation
ISO 10303-307	Abstract test suite: Sheet metal die planning and design
ISO 10303-308	Abstract test suite: Life cycle product change process
ISO 10303-309	Abstract test suite: Design through analysis of composite and metallic structures
ISO 10303-310	Abstract test suite: Electronic printed circuit assembly, design and manufacture
ISO 10303-311	Abstract test suite: Electronics test diagnostics and remanufacture
ISO 10303-312	Abstract test suite: Electrotechnical plants
ISO 10303-313	Abstract test suite: Numerical control process plans for machined parts
ISO 10303-314	Abstract test suite: Core data for automotive mechanical design
ISO 10303-315	Abstract test suite: Ship arrangement
ISO 10303-316	Abstract test suite: Ship moulded forms
ISO 10303-317	Abstract test suite: Ship piping
ISO 10303-318	Abstract test suite: Ship structures

ISO 10303-320	Abstract test suite: Printed circuit assembly manufacturing planning
ISO 10303-321	Abstract test suite: Functional data and their schematic representation for process plant
ISO 10303-322	Abstract test suite: Exchange of design engineering to manufacturing for composite structures
ISO 10303-323	Abstract test suite: Exchange of design and manufacturing information for cast parts
ISO 10303-324	Abstract test suite: Mechanical products definition for process planning using form features
ISO 10303-325	Abstract test suite: Structural building elements using explicit shape representation
ISO 10303-326	Abstract test suite: Ships mechanical systems
ISO 10303-327	Abstract test suite: Plant spatial configuration
ISO 10303-328	Abstract test suite: Building services: Heating, ventilation and air conditioning
ISO 10303-330	Abstract test suite: Building structural frame: Steel work
ISO 10303-501	Application interpreted construct: Edge-based wireframe
ISO 10303-502	Application interpreted construct: Shell-based wireframe
ISO 10303-503	Application interpreted construct: Geometrically bounded 2D wireframe
ISO 10303-504	Application interpreted construct: Draughting annotation
ISO 10303-505	Application interpreted construct: Drawing structure and administration

ISO 10303-506	Application interpreted construct: Draughting elements
ISO 10303-507	Application interpreted construct: Geometrically bounded surface
ISO 10303-508	Application interpreted construct: Non-manifold surface
ISO 10303-509	Application interpreted construct: Manifold surface
ISO 10303-510	Application interpreted construct: Geometrically bounded wireframe
ISO 10303-511	Application interpreted construct: Topologically bounded surface
ISO 10303-512	Application interpreted construct: Faceted boundary representation
ISO 10303-513	Application interpreted construct: Elementary boundary representation
ISO 10303-514	Application interpreted construct: Advanced boundary representation
ISO 10303-515	Application interpreted construct: Constructive solid geometry
ISO 10303-516	Application interpreted construct: Mechanical design context
ISO 10303-517	Application interpreted construct: Mechanical design geometric presentation
ISO 10303-518	Application interpreted construct: Mechanical design shaded presentation
ISO 10303-519	Application interpreted construct: Change

All P-LIB (ISO 13584) parts have the general title “Parts libraries”.

ISO 13584-1	General overview
ISO 13584-10	Conceptual model
ISO 13584-20	General resources
ISO 13584-24	Library supplier format
ISO 13584-26	Identification codes
ISO 13584-31	Programming interface
ISO 13584-42	Dictionary methodology
ISO 13584-101	Geometrical view exchange protocol by parametric program
ISO 13584-102	Geometrical view exchange protocol by ISO 10303 conforming specification

No standards have yet been proposed by ISO TC184/SC4/WG8 “MANDATE”.

The structure of the STEP Committees

As noted in Chapter 4, STEP is developed within ISO Technical Committee TC184 “Industrial Automation Systems and Integration”, Sub-Committee SC4 “Industrial Data”. Within SC4, a number of working groups undertake the actual technical work of standards development. The structure and composition of these working groups varies with time, depending on the demands of industry that are being fulfilled within the standardization process.

The committee structure in place during the development of the “initial release” of STEP (1990-1994) is described below.[1] The work of SC4 is organized into one *ad hoc* group, nine working groups and three advisory groups.

Parametrics	A work item on Parametrics was approved in 1994. Until the relationship of this work to that on STEP, P-LIB and MANDATE is determined, and the impact on the standards being developed assessed, the Parametrics group is managed separately from the various Working Groups of SC4.
WG2 Parts Library (P-LIB)	Responsible for developing the Parts Library standard ISO 13584.
WG3 Product Modelling	The largest of the working groups under SC4: WG3 is responsible for the development of STEP Application Protocols and Integrated Resources. WG3 is divided into a number of teams which cover the standardization needs of specific application areas or disciplines. These teams include those responsible for geometry, draughting, finite element analysis, building & construction, shipbuilding, automotive design, etc.
WG4 Qualification and Integration	WG4 is responsible for ensuring the technical consistency across the entire STEP standard. WG4 includes projects responsible for Resource Integration and AIM Development, these forming the core “data engineering” functions of STEP.
WG5 STEP Development Methods	WG5 is responsible for developing and documenting the architecture and methods used as the basis for the development of STEP; the development of the EXPRESS language is also undertaken within WG5.[2]
WG6: Conformance Testing Procedures	WG6 is responsible for the development of the Conformance testing methodology and framework class of parts within STEP, as well as assisting projects that are developing Application Protocols in the design and review of conformance requirements and of abstract test suites.

WG7: Implementation Specifications	WG7 develops the implementation methods of STEP: physical file and SDAI
WG8: Industrial Manufacturing Management Data (MANDATE)	The scope of WG8 is the development of methods and standard data supporting the exchange of non-product definition data within industrial manufacturing plants.
JWG9: Electrical and electronic applications	This is a joint working group between ISO TC184/SC4 and IEC TC93, and is responsible for the development of those parts of STEP that cover electrical and electronic applications
WG10: Technical Architecture	WG10 is responsible for the identification and resolution of technical issues related to architecture and methodology, particularly as they apply to the common requirements of the three areas of standardization within SC4
Project Management Advisory Group (PMAG)	The PMAG is responsible for the project management of all STEP development activities.
Strategic Planning Advisory Group (SPAG)	The SPAG is responsible for the co-ordination of SC4 activities.
Editing Committee	The Editing Committee is responsible for ensuring technical and editorial consistency across all parts of STEP. This responsibility includes the development and maintenance of guidelines for documentation, etc.

[1] As work on this book is completed, a proposed revision to the structure of the working groups of SC4 is being considered. This may result in the creation of new working groups, and the merger or elimination of existing groups. The purpose of this reorganization is to improve the efficiency of the standards development process and its project management. All the functions defined under the current working group structure will be preserved. The working group on Technical Architecture (WG10) was created in 1994 as an initial stage in the reorganization.

[2] Responsibility for work on architecture and methods was transferred from WG5 to WG10 in 1995.

Appendix D: Sources of further information

This appendix identifies some of the possible sources of additional information about STEP.

National Standards Bodies

National Standards Bodies play two key roles in STEP: they act as a focal point for national activities contributing to the development of the standard, and they also publish and distribute the completed ISO standards. In some countries, the standards will be available exactly as published by ISO in Geneva. Other countries or regions adopt STEP as national standards: this may involve no more than adding a new cover page, or may require the translation of the standard into other languages.

Readers wishing to acquire copies of the STEP standards should refer in the first instance to their national standards bodies.

Projects and consortia

The following are points of contact for some of the STEP development and implementation projects mentioned in Chapter 9.

EMSA

European Marine STEP Association
c/o Det Norske Veritas Research A.S.
PO Box 300
N-1322 Høvik
Norway

EPISTLE

Shell International Petroleum Company Ltd.
ICT/47
Shell Centre
London
SE7 2NA
UK

PDES, Inc.

PDES, Inc.
5300 International Boulevard
North Charleston
SC 29418
USA

PI-STEP

Process Industries STEP Consortium
c/o ICI Engineering
Brunner House
Northwich
Cheshire
CW8 4DJ
UK

PlantSTEP

PlantSTEP
5090 Richmond Avenue #098
Houston
Texas 77056
USA

POSC

POSC
10777 Westheimer, Suite 275
Houston
Texas 77042
USA

POSC/CAESAR

The POSC/CAESAR Project
c/o Saga Petroleum a.s.
PO Box 490
1301 Sandvika
Norway

ProSTEP

ProSTEP Association
Julius-Reiber-Straße 15
D-64293 Darmstadt
Germany

SPI-NL

Projectbureau SPI-NL
c/o Akzo Nobel Engineering
Postbus 9300
6800 SB Arnhem
The Netherlands

STEP on the Internet

As befits a standard used for digital communication, the STEP development community makes extensive use of electronic mail and other facilities offered through Internet.

An e-mail “exploder” is used as one of the primary mechanisms for information exchange and issue discussion within the STEP community. To join this exploder, send a message to:

```
majordomo@cme.nist.gov
```

containing the command:

```
subscribe sc4
```

as the first line of the message (*not* the subject). For information on the operation of the SC4 e-mail exploder (and others in specialist areas of STEP), send a message containing the command:

```
help
```

to the same address.

The National Institute of Standards and Technology (NIST) also operates a STEP On-line Information Service (SOLIS). This can be accessed by several routes: kermit server, e-mail, anonymous FTP, or Internet gopher. To get information on the use of SOLIS by each of these methods, send an initial e-mail message to:

```
solis@cme.nist.gov
```

containing the command:

```
help
```

There are also several World Wide Web pages providing information related to STEP.

```
http://www.imw.tu-clausthal.de/imw/projects/step/stand.html
```

accesses the P-LIB archive at the University of Clausthal (Germany). This site includes a mirror of the NIST SOLIS archive, and is also available via anonymous FTP.

```
http://www.eeel.nist.gov/pipde/Intro.html
```

provides access to a library of information on projects related to STEP and other digital product data technologies, created and maintained by the US National Initiative for Product Data Exchange.

```
http://www.hike.te.chiba-u.ac.jp/ikeda/documentation/home.html
```

is the home page of a public archive maintained by Chiba University in Japan, containing documentation and supporting information on STEP.

Appendix E: Bibliography

CADDETC, *The Exchange Agreement: Guidelines for the Successful Exchange of CAD-CAM Data*, CADDETC, 1990.

CIMdata, *The Executive Guide to Product Data Management,* Department of Trade and Industry booklet, 1994.

Executive Guide to STEP for the Process Industries, Process Industries STEP Consortium, 1995

Hamel, Gary and C. K. Prahalad, *Competing For The Future*, Harvard Business School Press, 1994.

Hammer, Michael and James Champney, *Re-engineering the Corporation: A Manifesto for Business Revolution*, Nicholas Brealey Publishing Limited, 1993.

Owen, Jon, *STEP: An Introduction*, Information Geometers Ltd., 1993

Smith, Joan M., *An introduction to CALS: the strategy and the standards*, Technology Appraisals, 1990.

Wellington, Joan and Bradford Smith, *ISO TC184/SC4 Reference Manual,* National Institute of Standards and Technology, NISTIR 5665, June 1995.

Willis, Robert, *Implementing the Vision – A Case Study Embracing CALS Technologies and Philosophies,.* CALS Roadmap 2000 Case Study Report, January 1995.

Wilson, Peter and Douglas Schenck, *Information Modelling the EXPRESS way*, Oxford University Press, 1993.

Womack, James P., Daniel T. Jones and Daniel Roos, *The Machine That Changed The World: Based On The Massachusetts Institute of Technology 5-Million-Dollar 5-Year Study On The Future Of The Automobile*, Rawson Associates, 1990.

Appendix F: Glossary of terms and abbreviations

AEC	Architecture Engineering Construction. Within work on STEP, the term "AEC" covers building & civil engineering, process plant, shipbuilding and offshore industries.
AIC	Application Interpreted Construct. A collection of EXPRESS constructs that fulfil a common requirement of two or more Application Protocols
AIM	Application Interpreted Model. A model within a STEP Application Protocol that fulfils the requirements of the Application Protocol using constructs defined in the STEP Integrated Resources. Defined in EXPRESS.
AMT/4	BSI/DISC technical committee responsible for UK input to STEP.
ANSI X.12	US national standard for EDI messages
AP	Application Protocol. A STEP Part which specifies the data constructs that satisfy a specific set of industrial requirements.
ARM	Application Reference Model. A model within a STEP Application Protocol that defines requirements from the viewpoint of application experts. Usually defined using IDEF1X, NIAM or EXPRESS-G.
B-REP	Boundary Representation. A mathematical description of three dimensional geometry used in many CAD systems.
BSI	British Standards Institute: UK national standards body.
CAD	Computer Aided Design
CAD*I	ESPRIT I project: responsible for initial development of STEP geometry models.

CADEX ESPRIT II project: responsible for initial development of ISO 10303-204 and ISO 10303-205, as well as for prototype implementations of the use of STEP for the exchange of geometry information.

CAE Computer Aided Engineering

CALS Continuous Acquisition and Life cycle Support: a US Department of Defense initiative aiming to achieve cost savings and quality improvement in the procurement and support of complex weapons systems.

CAM Computer Aided Manufacturing

CARP Computer-aided Rapid Prototyping: EUREKA project applying STEP models and methods to rapid prototyping technologies.

CAx General term for CAD, CAM, CAE, etc.

CD Committee Draft: proposed standard distributed internationally for ballot and approval. A STEP CD is balloted for a three month period amongst the countries participating in the work of ISO TC184/SC4.

CEN Commité Européen de Normalisation: European standards organization (equivalent of ISO).

CENELEC Commité Européen de Normalisation Electrotechnique: European standards organization (equivalent of IEC).

CIMSTEEL EUREKA project using STEP models and methods within the structural steel industry.

CTS2 (CTS-CAD/CAM): European Community project, within the Conformance Testing Services programme, responsible for initial definition of testing requirements within STEP.

DGIII Directorate General of the European Commission responsible for standardization and ESPRIT Projects.

DIS Draft International Standard. The stage in approval of a standard following approval a Committee Draft. A DIS is balloted for a six month period amongst all the member nations of ISO and IEC and, if approved, is published as a full International Standard.

DISC Industry-supported agency of BSI responsible for IT standards in the UK.

DP Draft Proposal: initial distribution of a proposed standard for international ballot and review (renamed Committee Draft in 1991)

EDI Electronic Data Interchange

EDIFACT International standard for EDI messages

EPDEN European Product Data Exchange Network. Liaison between centres of excellence and expertise in STEP and other product data standards. Members are CADDETC (UK), Association GOSET (France), IRPL (France), SINTEF (Norway), IVF (Sweden), TNO (the Netherlands) and ProSTEP (Germany).

ESPRIT European Strategic Programme for Research in Information Technology

EUREKA European Community research programme.

EXPRESS a data definition language (ISO 10303-11).

EXPRESS-G a graphical subset of the EXPRESS language.

GMAP Geometric Modelling Applications Project: US Air Force project – precursor to STEP/PDES activities.

IDEF0 ICAM Definition Language Zero. A graphical activity modelling language and methodology developed within the US Air Force "Integrated Computer Aided Manufacturing" project.

IDEF1X ICAM Definition Language One – Extended. A graphical data modelling language and methodology developed within the US Air Force "Integrated Computer Aided Manufacturing" project

IEC International Electrotechnical Commission

IGES Initial Graphics Exchange Specification: ANSI standard for exchange of data between CAD systems.

IMPPACT ESPRIT II project: developed early prototype product data sharing environment based on STEP.

IR Integrated Resource. A STEP Part that defines data constructs common to many application requirements.

ISDN Integrated Services Digital Network: high speed digital communications services provided using public telephone systems.

ISO International Organization for Standardization

IT Information Technology

MARITIME ESPRIT III project: successor to NEUTRABAS project – developing (in conjunction with NIDDESC) STEP Application Protocols for the shipbuilding industry.

NC Numerically Controlled: used to describe or denote automated machine tools and other industrial systems

NEUTRABAS ESPRIT II project: developed early prototype product data sharing environment for the shipbuilding industry, based on STEP.

NIAM Nijssen Information Analysis Method. A graphical data modelling language and methodology.

NIDDESC Navy Industry Digital Data Exchange Standards Committee. US activity, supported by US Navy and major shipbuilding yards: working with the MARITIME project to define STEP Application Protocols for the shipbuilding industry.

NIRO ESPRIT II project: contributed to STEP model for kinematics based on the requirements of robotics systems.

NIST National Institute for Standards and Technology (formerly National Bureau of Standards). Agency of the US Department of Commerce, actively involved in STEP development.

PDDI Product Data Definition Interface: US Air Force project – precursor to STEP/PDES activities.

PDES Product Data Exchange using STEP: US activity supporting the development of STEP.

PDES, Inc. Major US industry collaborative project, formed in 1988 to accelerate the development and implementation of STEP.

PDI Product Data Interchange: "paperless" exchange of engineering and other technical data within and between enterprises.

PDTAG Product Data Technology Advisory Group: expert group advising the European Commission with respect to STEP and other product data technology issues.

PI-STEP Process Industries STEP Consortium: UK project developing and demonstrating applicability of STEP in the process industries.

PISA ESPRIT III project: investigating extensions to STEP to support process data and implementations based on object-oriented technologies.

ProcessBase ESPRIT III project: developing and implementing STEP Application Protocols for the process industries.

PRODEX ESPRIT III project: successor to CADEX project – developing STEP Application Protocols and implementations for the automotive industry.

ProSTEP Major STEP development and implementation project supported by German industry and government

RAMP Rapid Acquisition of Manufactured Parts: US Navy project and facility combining Flexible Manufacturing Cell technology with STEP-based product data models.

SDAI Standard Data Access Interface (ISO 10303-22).

SQL Structured Query Language. A standard interface specification for database management systems.

STEP Standard for the Exchange of Product Model Data: informal name for ISO 10303 Product Data Representation and Exchange

X.400, X.500 Standards for networking and communications

Index

E

G

I

L

M

N

O

P

Q

S