IMPROVING THE INTELLIGIBILITY OF STEP MODELS FOR THE CONSTRUCTION INDUSTRY

Michael Liu[1], Philip Lovett[2], Nick Godwin[2], Eric Fletcher[1]
1. University of Sunderland
2. Coventry University

ABSTRACT: The overall objective of STEP (ISO-10303) is to become the world-wide standard for the representation and exchange of product data. This is to be achieved by the provision of a mechanism capable of describing product data throughout its life-cycle, completely independently of any individual implementation methods. STEP - the Standard for the Exchange of Product Model Data - is the outcome of a large international effort to develop product and process model data standards, which will enable data exchange between diverse computer systems and industrial applications for architecture, engineering and construction.

In 1993 the Application Protocol Planning Project for Building and Construction (APPP-BC) identified nested families of models required to represent information relating to the construction industry. Consequently the integrated resource Part 106 Building Construction Core Model, application protocols AP225 Structural Building Elements Using Explicit Shape Representation, AP228 Building Service HVAC, and AP230 Building Structural Frame: Steelwork were developed and included in the Standard.

STEP encompasses all aspects of a product and its life-cycle. However, the complexity of STEP models is a significant barrier to their wider use. Therefore user-friendly tools which can help users to understand STEP models and their data structures are highly desirable. These will enable STEP-related applications to be developed and the potential for integrated systems using STEP to be achieved in a wide range of industries, including construction.

The research presented in this paper involved a thorough investigation of STEP data structures and software tools that can be used to improve the intelligibility of STEP data. The initial investigations provided a basis for describing a range of potential software utilities and produced a broad characterisation of users of STEP. The more detailed work then focused upon application developers and Application Protocol (AP) model developers. The need for visualisation tools to improve the usability of the Standard was recognised, and prototypes produced, contributing to the fulfilment of the needs identified.

KEYWORDS: STEP, Application Protocol, EXPRESS, Visualisation, ISO 10303

INTRODUCTION - THE NEED FOR STEP

STEP (STandard for the Exchange of Product Model Data) is an ISO programme involving researchers and industrialists in many countries including the US, UK, Germany, France, Italy and Japan in the development of an engineering product data exchange standard, which is documented as *ISO 10303 Industrial Automation Systems – Product Data Representation and Exchange* [ISO 1994]. The history of STEP can be traced back to the development of specifications such as IGES (Initial Graphical Exchange Specification) in the US [NIST

Construction Informatics Digital Library http://itc.scix.net/
paper w78-2000-584.content

1980], VDA-FS (Verband der Automobilindustrie-Flachen-Schenitstelle) in Germany [DIN 1986], and SET (Standard D'Exchange et de Transfert) in France [AFN 1985]. In 1984, deficiencies in the existing generation of product data standards and specifications had been identified and were well known to industry, hence the birth of STEP.

Information about the products and manufacturing processes in which raw materials, components, and sub-assemblies, etc. are turned into products is referred to as product data, especially when it is created or used by CAD (Computer-Aided Design), CAE (Computer-Aided Engineering) and CAM (Computer-Aided Manufacturing) systems. It is often necessary to migrate product data to new generations of software and hardware, perhaps several times, since it has a relatively longer life-time than the applications used to process it, and the systems software. So it should be independent of any proprietary format. In addition, many engineering enterprises need to exchange data concerning their products in a computer-readable form. The exchange of data not only enables internal and external communication (both within the organisation and with clients, contractors, sub-contractors, suppliers, and partners etc.) but also makes the engineering data generated by one application readable by other application programs. The need to share information across many heterogeneous databases that have been designed independently and often use different data models and DBMSs (Database Management Systems) has grown significantly. Contributing factors include organisation mergers, collaborative agreements, integration across areas, and supplier-client integration. Meanwhile, high performance computer networks allow companies to collaborate electronically. The design, manufacture and support of many products involve the international partnership of companies. Consequently a product data integration standard to enable the inter-working of different computer systems has become a vital requirement.

While STEP has the potential of becoming a key information technology in a wide range of industrial sectors, the sheer bulk of its documentation and the complexity of its data structure can pose problems for users at every level. This paper introduces some of the essential concepts upon which STEP is based, including the framework of parts, and the relationships between them, with particular emphasis on attempts to utilise the standard within the construction industry. Problems relating to the comprehension of complex systems and situations are identified, and software tools which can be used to improve the intelligibility of STEP data models are discussed. In particular, software solutions enabling the visualisation of aspects of models are described in some detail.

STEP – THE STRUCTURE AND CONTENT

The objectives of STEP include the creation of a single international standard to cover all aspects of CAD/CAM data exchange, and the implementation of this standard within industry, superseding all others [Owen 1993]. In order to achieve this it is necessary to standardise a mechanism for describing product data throughout its life cycle, which is independent of any particular system, and to separate the description of product data from its implementation. Such a standard is suitable for neutral file exchange and long-term archiving, as well as providing the basis for shared product databases. Engineering data is complex, and STEP, as the standard for exchange of product data, must provide comprehensive coverage of the product development life-cycle. Therefore STEP is divided into many parts and each part covers a specific area, such as file formats, programming interfaces, testing procedures, and specific industrial information models etc. Figure 1 shows the high-level structure of the STEP standard.

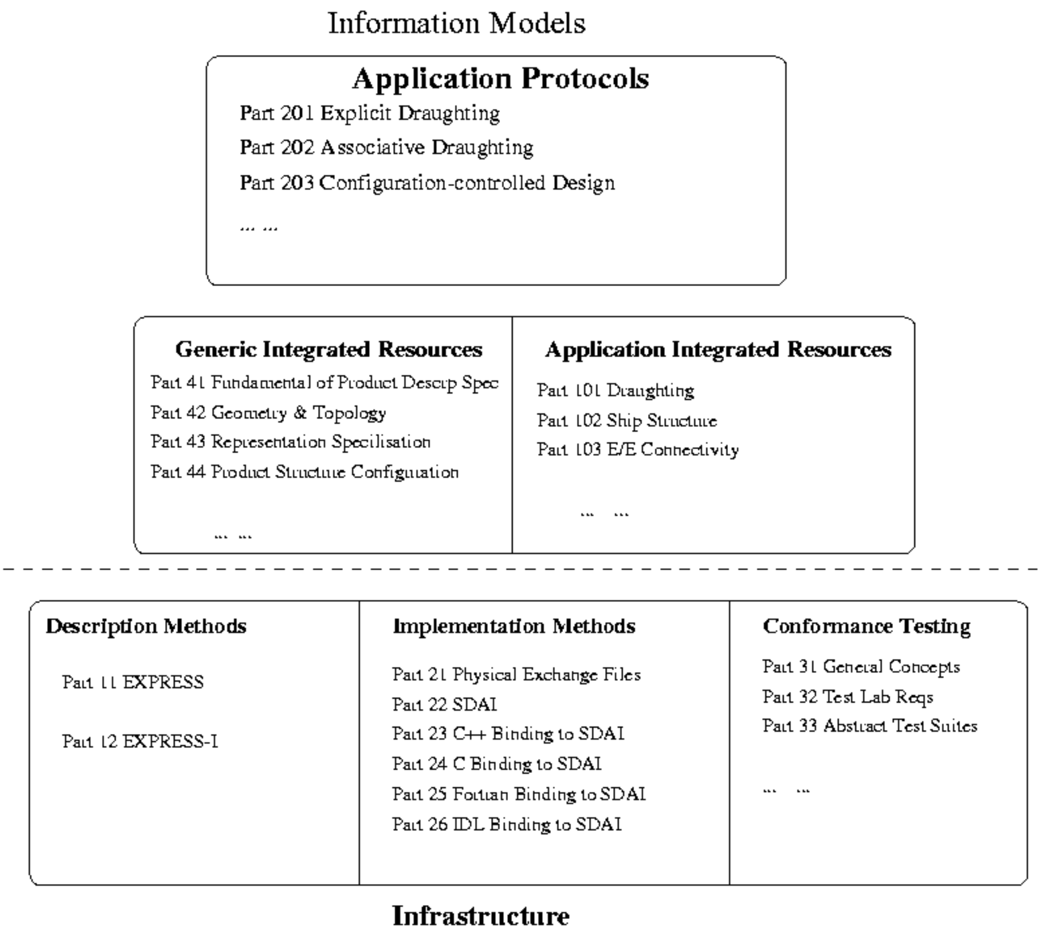

Figure 1. STEP Struture

As shown in figure 1, the infrastructure parts and the industry-specific parts have been separated. To date most of the infrastructure parts are completed, but some of the industry-specific parts are still under development. Also, industries can develop their own application protocols based upon the infrastructure.

DESCRIPTION METHODS – EXPRESS

Information can be exchanged only if both parties agree on the interpretation of data, and an information model is a representation of that agreement. A number of information modelling methods have been available since the late 70's and early 80's, including NIAM [Nijssen 1989], Entity-Relationship Diagrams [Chen 1976], and IDEF1X [WPAFB 1985]. But all of them rely heavily upon graphical representations. A lexical and formal form of information modelling can be easy to write and process. It can also be read by humans. EXPRESS as one of the core parts of STEP has been designed in this manner. The first prototype of EXPRESS was introduced to the STEP effort in 1986 and it went through many revisions along with the development of STEP. EXPRESS provides an unambiguous, computer-readable representation of product data and is also human-readable.

An EXPRESS information model is organised into schemata. A schema is used to define a topic of interest such as “topology”, in order to structure and partition the data. Hence a large information model can be divided into many schemata, which serve as a scoping mechanism. Within each schema there are the following categories of definitions:

- **Entities** define real world objects and their properties in the form of attributes. Attributes can have simple or structured values, and can be used to represent relationships between entities. Entities can be hierarchically classified as supertypes and subtypes. Subtypes can inherit attributes from supertypes.
- **Types** describe the values that attributes can take (i.e. their domains). There are seven pre-defined simple types in EXPRESS and new types can be constructed from these built-in types. An entity may itself be used as an attribute type.
- **Constraints** specify limitations on the values of attributes and entities, and are a powerful mechanism for describing behaviour in EXPRESS.
- **Rules** are used to describe constraints, and may be local – applying only to named entities, or global – applying to the entire information model.
- **Functions and procedures** are defined to assist in the algorithmic description of constraints.

EXPRESS is a specification and requirements language, and although procedural code is used within functions and procedures, it is purely descriptive [Schenck 1994]. In other words, it is not a programming language. The descriptive power of EXPRESS is considerable, but this is to some extent at the expense of human readability - models can be very detailed and complicated, and hard to understand.

IMPLEMENTATION METHODS

There is, within a STEP information model, no assumption about how the information will be stored in a computer. This contrasts greatly with other methods, in particular those based on the relational paradigm. However, in order to exchange information, it is necessary to agree on a common format for the representation of values corresponding to an EXPRESS model (instances). STEP Physical Files, described in Part 21 of the standard, are termed an implementation method, and are used for this purpose. Other implementation methods include the SDAI (Standard Data Access Interface - Part 22), which is used to describe operations that can be performed on instances conforming to an EXPRESS specification.

STEP INFORMATION MODELS – IR, AP, AND AIC

STEP information models are further classified into three groups: Integrated Resources (IR), Application Protocols (AP) and Application Integrated Constructs (AIC).

IRs are a number of conceptual information models which have been developed to reflect and support the common requirements of different product data application areas. While the data models defined in the IRs actually form a single conceptual model for product data, it is intended to be combined, refined and modified within an AP to meet a particular industrial need. IRs can be further categorised into two sub-classes: Generic Integrated Resources define data models totally independent of applications and describe very general characteristics of products across all industries, and Application Integrated Resources expand the generic resources to include the needs of specific areas of applications.

The standard models in the IRs are context-independent, and therefore may be used in a wide variety of applications. But their direct implementation would face this obstacle: each developer would implement only those entities relating to their own systems. Hence communication between systems using different components selected from the IRs would be problematic. The solution devised by STEP is the AP, which defines and fulfils the special requirements of a particular application of product data relevant to a specific industrial need, based on the IRs. APs provide not only an EXPRESS model and a list of objects, but also the context in which they are to be used and a mapping, which indicates the particular task they perform in the application. APs form the largest and the most important class of parts within STEP, and supply the link between STEP and the outside world.

The concept of an AIC has been introduced recently to STEP. It is a construct for describing the interoperable segments of definitions shared by multiple APs. AICs are sets of refined definitions, which have to be used as a single unit without any additional refinements.

BUILDING CONSTRUCTION EFFORTS WITHIN STEP

The initial interest and involvement in STEP from within the construction industry can be traced back to 80's. In 1993 an Application Protocol Planning Project for Building and Construction (APPP-BC) was initiated and nested families of construction models were identified, in order to represent information from building construction industries [Froese 1996]. To date, an application integrated resource – Part 106: The Building Construction Core Model (BCCM) has been completed. It is a model intended to serve as a unifying reference for building construction APs. It consists of three main parts: a core model that provides central building construction concepts common to all areas of the industry, common references used within building construction, and common data. Three application protocol projects are under development. They are:

- AP225 – Structural Building Elements Using Explicit Shape Representation. AP225 is for the representation of buildings as assemblies of elements (beams, columns, etc.) along with the explicit 3D geometry of each element and some additional information such as material properties, building element classification or element versions.
- AP228 – Building Services HVAC. AP228 covers heating, ventilation, and air conditioning building services.
- AP230 – Building Structural Framework: Steelwork. AP230 focuses on construction steelwork frame design, analysis and detailing, and fabrication.

APPLICATION PROTOCOLS

APs form the largest and core part of the STEP standard and may be the only part that is of interest to an industrial end-user. APs are formal documents covering a set of activities in a product life-cycle. They are large information models constructed from an application-specific interpretation of the context-independent entities present in the IRs. An AP consists of the following parts:

- Scope: This is the first part of each AP and describes the set of activities in a product life-cycle. It is documented mainly in English, or using IDEF0 diagrams [NIST 1993] to form the Application Activity Model (AAM).
- Application Reference Model (ARM): This describes the comprehensive requirements for the application domain in terms of the product information and context. The requirements

are defined in terms of basic application objects organised into related sets called Units of Functionality (UoF). Also provided is a list of application assertions that specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects.

- Application Interpreted Model (AIM): This contains a conceptual schema which is the result of the interpretation of the IRs consistent with the ARM. It includes a mapping table that maps ARM elements (objects, attributes, and assertions) to one or more AIM elements.
- Annexes include graphical presentations of the AAM, ARM, and AIM.

UNDERSTANDING AP MAPPING TABLES

In practice, a user of STEP will read the scope statement of an AP to see whether it is useful for his/her own activities. An implementor will read its self-contained EXPRESS schema with a view to writing software. But how easily would users or implementors understand the data structure of an AP? By nature STEP is a complex standard. APs are constructed upon the context-independent entities presented in the IRs, which are already large models. APs can only be larger. Taking AP214 - *Core Data for Automotive Mechanical Design Processes* - as an example, it has in total 1799 pages in written form including 14 annexes. The electronic form of the AIM EXPRESS schema contains 13385 lines. The ARM lists 365 application objects grouped into 35 UoFs, and 837 application assertions.

A particular problem concerns mapping tables. When mappings between application elements and the AIM elements are straightforward, users have no problem in understanding them. However, a mapping path such as that shown in the following table needs a little more effort to understand. To search a text or electronic document for the objects listed in the mapping path and their relationships in order to make the reference path clear can be quite troublesome and confusing. The EXPRESS-G diagram provided in the annex to an AP is meant to improve the intelligibility of the AIM schema. However, because of the size of the schema, the EXPRESS-G diagram is also large, and it too can be difficult to follow. By searching a large AIM model using a text editor, application developers can locate all the related AIM constructs and their connections represented in the reference path of the mapping table. It would save a great deal of time and trouble if this searching process could be automated. Moreover, it would make this part of the mapping table clearer and easier to understand if a local graphical representation of these constructs and their relationships could be produced. The intelligibility of the local mapping table and complicated EXPRESS entities is the aim of this research.

Application element	AIM element	Source	Rules	Reference path
shape_element to wireframe_ model_2d (as composition)	PATH			Shape_aspect shape_aspect.of_shape -> production_definition_shape <= property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation => shape_representation => geometrically_bounded_2d_wireframe_representation

Table 1. Sample Mapping Table – Mapping Path

By searching the complete electronic form of the AIM model using a text editor, a local graphical interpretation of this part of the mapping table has been produced and is shown as figure 2. This makes the reference path much easier to understand.

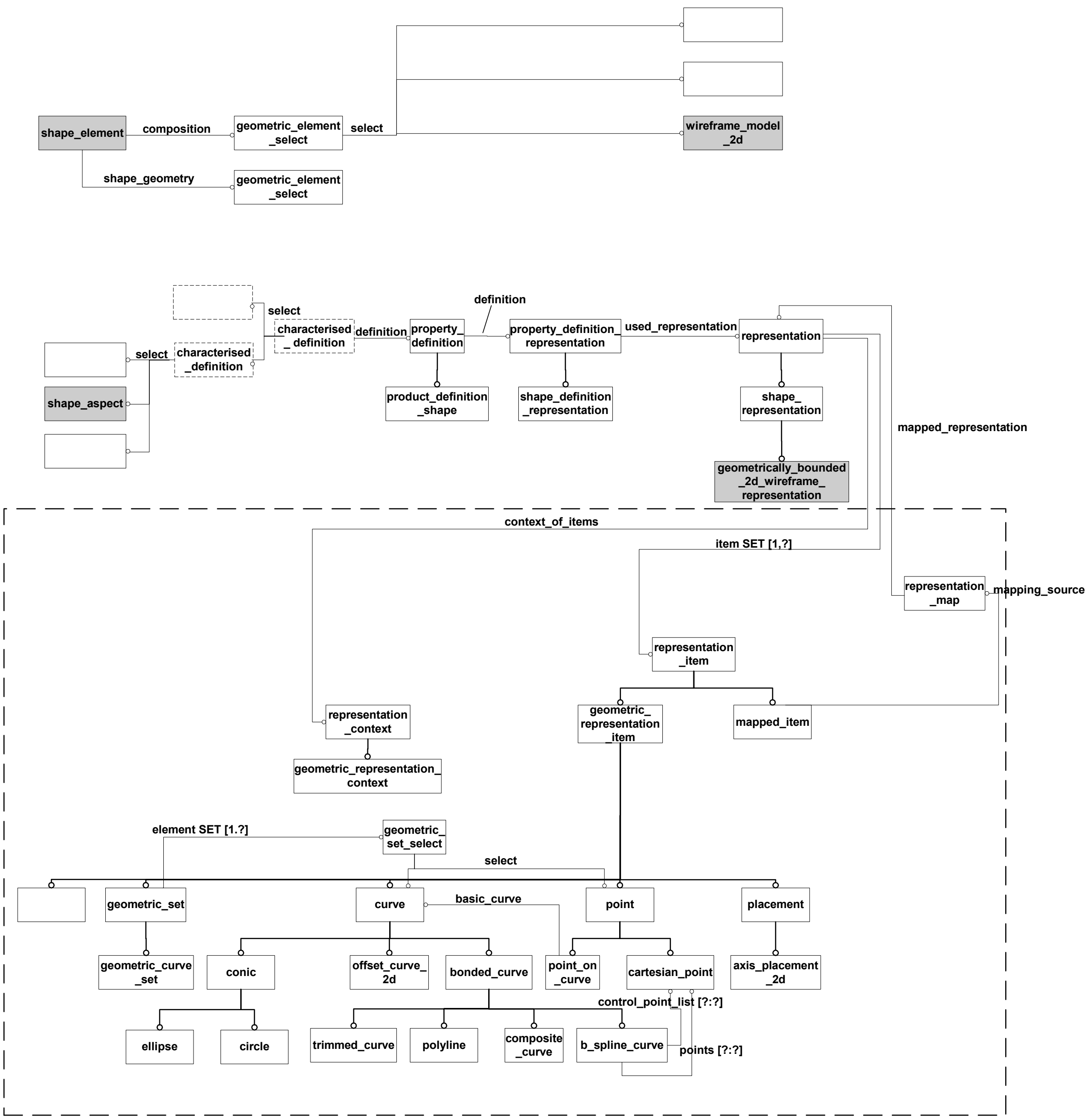

Figure 2. Local Graphical Interpretation for Part of the Mapping Table

UNDERSTANDING COMPLICATED WHERE RULES

The expressions within rules are often very complicated although EXPRESS provides operators and functions for their description. This is primarily because of the complex nature of the realities they so richly describe, usually in an engineering context. Furthermore, this expressive power is considered more important than human readability. It is not unusual for rules beginning with the word WHERE to contain complicated expressions of more than ten lines of code. The complexity of such WHERE rules is one of the greatest difficulties in the understanding of STEP standards.

In figure 2 the items bounded by the dotted line do not appear in the reference path in the mapping table, yet the area is highly significant. They are referred to in the additional constraints (expressed as WHERE rules) for the entity *geometrically_bounded_2d_wireframe_representation,* which is the mapping destination from *shape_aspect*. The following is the definition of the entity itself:

```
ENTITY geometrically_bounded_2d_wireframe_representation
  SUBTYPE OF (shape_representation);
  WHERE
   wr1 : SELF.context_of_items\geometric_representation_context.
         coordinate_space_dimension = 2;
   wr2 : SIZEOF(QUERY(item <* SELF.items | NOT (SIZEOF(TYPEOF(item) * [
         'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET',
         'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_2D','AUTOMOTIVE_DESIGN.MAPPED_ITEM'
         ]) = 1))) = 0;
   wr3 : SIZEOF(QUERY(item <* SELF.items | SIZEOF(TYPEOF(item) * [
         'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET',
         'AUTOMOTIVE_DESIGN.MAPPED_ITEM']) = 1)) >= 1;
   wr4 : SIZEOF(QUERY(mi <* QUERY(item <* SELF.items | (
         'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(item))) | NOT (
         'AUTOMOTIVE_DESIGN.' +
         'GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION' IN TYPEOF(mi\
         mapped_item.mapping_source.mapped_representation)))) = 0;
   wr5 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
         'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
         SIZEOF(QUERY(elem <* gcs\geometric_set.elements | NOT (SIZEOF(TYPEOF(
         elem) * ['AUTOMOTIVE_DESIGN.B_SPLINE_CURVE','AUTOMOTIVE_DESIGN.CIRCLE'
         ,'AUTOMOTIVE_DESIGN.COMPOSITE_CURVE','AUTOMOTIVE_DESIGN.ELLIPSE',
         'AUTOMOTIVE_DESIGN.OFFSET_CURVE_2D','AUTOMOTIVE_DESIGN.POINT',
         'AUTOMOTIVE_DESIGN.POLYLINE','AUTOMOTIVE_DESIGN.TRIMMED_CURVE']) = 1))
         ) = 0))) = 0;
   wr6 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
         'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
         SIZEOF(QUERY(crv <* QUERY(elem <* gcs\geometric_set.elements | (
         'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(elem))) | NOT (
         valid_basis_curve_in_2d_wireframe(crv,
         'AIC_GEOMETRICALLY_BOUNDED_2D_WIRERAME')))) = 0))) = 0;
   wr7 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
         'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
         SIZEOF(QUERY(pnt <* QUERY(elem <* gcs\geometric_set.elements | (
         'AUTOMOTIVE_DESIGN.POINT' IN TYPEOF(elem))) | NOT (SIZEOF(TYPEOF(pnt)
         * ['AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
         'AUTOMOTIVE_DESIGN.POINT_ON_CURVE']) = 1))) = 0))) = 0;
   wr8 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
         'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
         SIZEOF(QUERY(pl <* QUERY(elem <* gcs\geometric_set.elements | (
         'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(elem))) | NOT (SIZEOF(pl\
         polyline.points) > 2))) = 0))) = 0;
 END_ENTITY;
```

Entity *geometrically_bounded_2d_wireframe_representation* is a subtype of entity *shape_representation*, which is itself a subtype of entity *representation*. In EXPRESS, a subtype inherits all the attributes and constraints of its supertype(s). So the three attributes *name, item*, and *context_of_items* of the entity *representation* are inherited and the WHERE rules are in fact constraints on these inherited attributes.

With the help of the local graphical representation, the above complicated WHERE rules can be interpreted as follows:

- **wr1:** 'context_of_items' is an inherited attribute from the entity 'representation', which is a supertype of the entity 'shape_representation', which is a supertype of the entity ‘geometrically_bounded_2d_wireframe_representation’. The attribute points to the entity ‘representation_context’. Entity ‘geometric_representation_context’ is a subtype of ‘representation_context’ and has an attribute ‘coordinate_space_dimension’. wr1 requires the value of the attribute to be 2.

- **wr2:** any ‘item’ is exactly one of ‘geometric_curve_set’, ‘axis2_placement_2d’, or ‘mapped_item’. (The attribute ‘items’ is a set of values for which ‘item’ is a generic example.)
- **wr3:** at least one of the ‘items’ is either a ‘geometric_curve_set’ or a ‘mapped_item’. (It also implies the ‘item’ is not both, but this could be deduced from wr2 anyway.)
- **wr4:** if ‘item’ is a ‘mapped_item’, then the ‘mapped_item.mapping_source.mapped_representation’ must be of type ‘geometrically_bounded_2d_wireframe_representation’ or a subtype.
- **wr5:** if ‘item’ is a ‘geometric_curve_set’, then each of the elements in the set must be one of the following: ‘b_spline_curve’, ‘circle’, …, ‘trimmed_curve’. wr5 allows seven different curves and a point.
- **wr6:** if ‘item’ is ‘geometric_curve_set’, then for each of its ‘elements’ that is a ‘curve’, the function ‘valid_basis_curve_in_2d_wireframe’ must evaluate to TRUE.
- **wr7:** if ‘item’ is ‘geometric_curve_set’, then each of its ‘elements’ that is a ‘point’ must be either a ‘point_on_curve’ or a ‘cartesian_point’.
- **wr8:** if ‘item’ is ‘geometric_curve_set’, then any of its ‘elements’ that is a ‘polyline’ is a polyline with more than two points.

A mechanism to improve the readability of rules in this way would significantly enhance the intelligibility of EXPRESS.

VISUALISATION OF LOCAL MAPPING TABLES AND COMPLICATED WHERE RULES

Software prototypes for the generation of local graphics for AP mapping tables and WHERE rules have been developed at Coventry University. The development tools were the freeware visualisation tool *daVinci* provided by the University of Bremen, Germany [UB 1999], and the NIST EXPRESS parser Fedex [NIST 1999].

The graph is a fundamental structure in computer science and it is well suited for representing sets of objects and the relationships between them. However the techniques to visualise such graphs are not common in today’s computer applications. High-quality graph layout is difficult to implement and there are very few reusable tools for graph visualisation. *daVinci* was developed in 1992 with the primary objective of providing a universal layout tool for directed graphs with a generic graph user interface which can be used on top of any application program. Graph data is sent by the client application to *daVinci*, which handles the layout algorithms and computer graphics necessary for visualisation.

A graph is a structure representing objects as nodes and the relationships between them as edges. The application objects in a mapping table and the components of an EXPRESS schema and their relationships are ideally suited for this form of representation. *daVinci* is an interactive tool for visualising directed graphs (i.e. those in which all edges have a direction - for each edge there is a parent and a child node). *daVinci*’s graph layout reflects these hierarchical relationships by arranging the nodes at horizontal levels so that all parent nodes are above their child nodes and all edges point downwards in a top-down layout, or to the right in a left-right layout. The direction from a parent to its child(ren) is represented by arrow(s). This is called hierarchical visualisation of a directed graph.

daVinci requires input in a textual format called term representation, so a graph can be created with any text editor. A graph editor tool is also provided. The research upon which this paper reports has included the development of software to parse an AP EXPRESS model and generate term representation. It is also necessary to input text files from the AP mapping table, or the entity with complex WHERE rules. The term representation is then input to *daVinci*, which produces the required graph.

The NIST EXPRESS Toolkit is a set of software tools for manipulating EXPRESS information models and Part 21 product models. It was developed within PDES (Product Data Exchange using STEP) activity and it is a research-oriented toolkit, available free via the Internet. One of the components of the toolkit is Fedex – an EXPRESS parser.

Fedex is a three-pass translator which performs syntactic and semantic analysis of EXPRESS schemata. It was implemented in ANSI Standard C using Yacc and Lex, developed on Sparc workstations. The first two passes are the standard parsing and symbol-table resolution passes of a traditional compiler. These produce a working form which consists of data structures reflecting the structure and contents of the EXPRESS source. In fact, the NIST Toolkit acts as a database for the schema information which was read from the input EXPRESS file and stored as an in-memory working form. The first two passes form the process of generating this database. Function calls can then be designed to retrieve and manipulate the data information in the working form as required [Clark 1990].

The third pass of the parser is a flexible output generation pass which can be tailored to various applications. It is flexible because Fedex leaves this pass to the developers. As the first two passes have already generated a well-structured in-memory database, the third pass is intended for users to traverse the data structure of the working form and produce output in a specified format, in this case, *daVinci* term representation.

Therefore, a third pass for Fedex was built into the two prototypes, making use of the working form that was generated by the first two. It traverses the data structure, identifies the required data (corresponding to a part of the mapping table or the objects related to an entity with complicated WHERE rules) and outputs it in the *daVinci* term representation format. The structure of the part of the mapping table, or the entity relationship can then be visualised using the *daVinci* system. Figure 3 is an IDEF0 diagram illustrating the specification of the software.

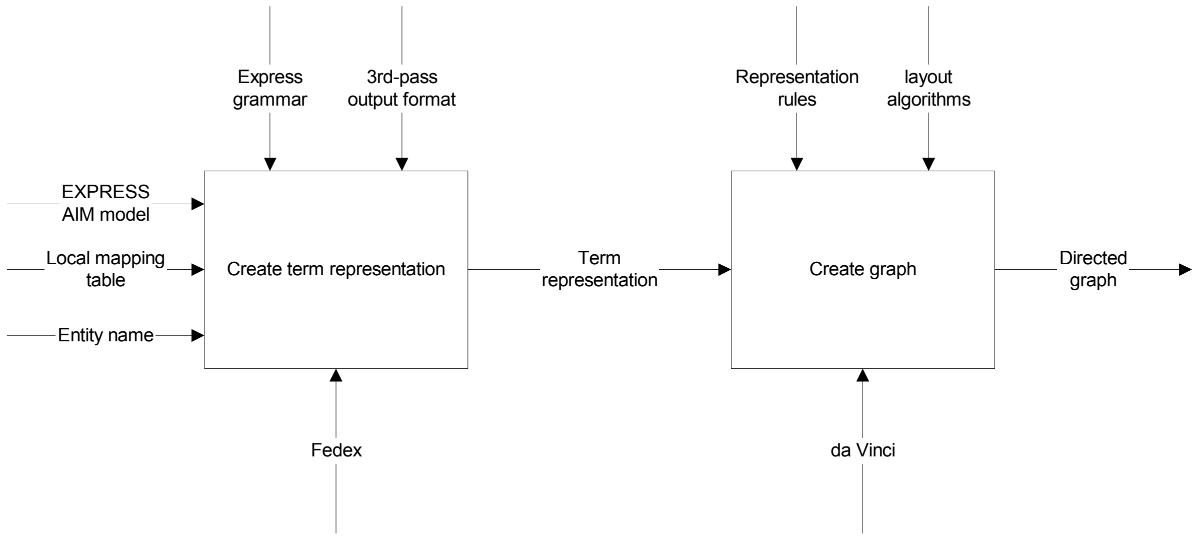

Figure 3. Prototype Software Specification

GRAPH LAYOUT

A *daVinci* graph is not the same as an EXPRESS-G diagram. It reflects hierarchical relationships by arranging nodes at horizontal levels so that all parent nodes are above their child nodes and all edges point downwards in a top-down layout. (Similarly, in a left-right layout all parent nodes are on the left and edges point rightwards.) This is not ideal for representing entities in an EXPRESS model. For example, when searching an AIM model both supertype and subtype may be found for an entity. EXPRESS-G shows an entity's supertype on a higher level, and a subtype on a lower level. *daVinci* shows both supertype and subtype as children of the entity. Thus the supertype/subtype hierarchy is not well presented in *daVinci*. In addition, if entity A is one of the attributes of entity B, the two entities will be presented by *daVinci* as parent and child whereas in EXPRESS-G they will be on the same level. These relationships are indicated in *daVinci* graph by putting an explanation on the edge which connects the two entities.

VISUALISATION OF STEP DATA USING VRML

The WWW (World Wide Web) today provides on-line textual and graphical information to anyone who can access the Internet and has the appropriate viewing software. This means that if companies using different CAD/CAE applications can send their 3D models across the Internet in a common format, anyone else with an appropriate viewer can look at the model without needing the original application. In this way, companies can provide their customers with their product design far more quickly. Similarly, amongst sub-contractors, engineering departments etc. within a company views of parts and assemblies can be made available in a timely manner through an intra-net application. Until fairly recently the graphical capability of the WWW has been limited to simple bitmap pictures and hyper-linked text, which are unsuitable for the representation of 3D product design. However the above scenario can now be realised using VRML (Virtual Reality Markup Language), which has been developed to provide the WWW with 3D modelling capability. VRML may be the above-mentioned common format for 3D CAD/CAE models created using different proprietary software. The idea of translating STEP Part 21 data into VRML in order to visualise the STEP data instances has been examined at Coventry University and an EXPRESS meta-model of VRML V1.0 was developed by South Bank University in 1996 [Bailey 1996].

STEP and IAI/IFC

The only ~~building and~~ construction AP which has~~ve~~ so far been ~~ratified and~~ standardised and ratified ~~up to now~~ is AP225. The launch of the International Alliance for Interoperability (IAI) in 1995 has slowed ~~somehow reduced~~ the STEP Building and Construction Application Protocol activities [IAI 2000]. The IAI was started by 12 companies involved in the AEC/FM industry ~~with the initiative~~ to provide models for use with their then emerging object-oriented CAD software. One of the reasons that IAI has prospered within the AEC/FM industry ~~is perhaps due to the fact~~ may be that it focuses on cross-disciplinary co-ordination. The us~~eage~~ of more than one AP at a time was not intended ~~originally~~ when the concept of APs was adopted in STEP [Metzger 1996]. However, for product data, areas to be modelled cannot always be covered by a single AP. Hence ~~the~~ AP interoperability remains a problem.

The puropose of the IAI is to specify how the “things” that could occur in a constructed facility should be represented electronically. These specifications, which represent a data structure supporting an electronic project model ~~useful in~~ for sharing data across applications, are called Industry Foundation Classes (IFC) [IAI 2000]. The IFC Object Model is the key deliverable of IAI and its specification view uses the international data definition language EXPRESS. Although the two prototypes presented in this paper were developed ~~particularly~~ for ~~the~~ STEP APs, they could ~~also~~ be applied to ~~work with~~ any EXPRESS models in order to improve their understandability ~~of the model~~. Therefore ~~it is hoped that~~ they could be useful tools for visualising~~ation of~~ the data structure of the IFC Object Models.

CONCLUSIONS

The extent to which industrial users and software developers will be able to adopt and exploit the potential of STEP to increase productivity and effectiveness is not yet clear. This uncertainty is partly due to the complexity of the standard. An attempt has been made to improve the intelligibility of AP models by providing localised visualisation for complicated AP mapping tables and entity constraints. A tool which may prove extremely valuable for this purpose is VRML, and an initial attempt at visualising STEP instances using VRML was made in the project. These software prototypes to generate local *daVinci* graphs, and VRML visualisation tools for AP models could be part of a useful toolkit to improve the understanding of AP models. Figure 4 outlines the structure of a possible distributed implementation of this toolkit, built upon the CORBA (Common Object Request Broker Architecture) infrastructure. A further development would be an interface definition as a wrapper for the functional components.

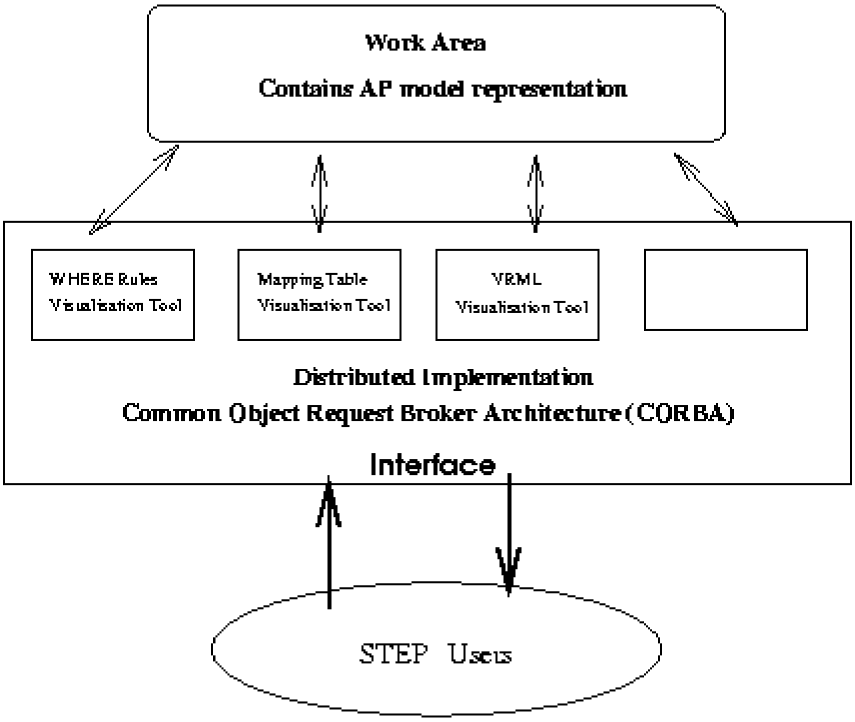

Figure 4. Model to Aid the Intelligibility of AP Data Structures

REFERENCES

Association Francaise de Normalisation (1985), *Automatisation industrielle. Representation externe des donnees de definition de produits. Specification du standard d'echange et de transferts (SET),* Version 85-08, Z68-300, (AFNOR) 85181, Paris.

Bailey I, David C, Marath R (1996), "An EXPRESS Model of VRML"*, EUG 96 Conference notes*, Toronto, Canada.

Chen P (1976), *The Entity Relationship Model – Towards a Unified View of Data*, ACM Transactions on Database Systems, Vol. 1, No. 1.

Clark S (1990), *An Introduction to the NIST PDES Toolkit,* Technical Report NISTIR4336, NIST, Gaithersburg, Maryland.

Deutsches Institute für Normung (1986), *VDA Flaechenschnittstelle (VDAFS),* Version 1.0, DIN 66301, DIN, Beuth Verlag, Berlin.

Froese T (1996), "STEP Standards and the Construction Industry"*, CSCE 1996 Conference (Volume I)*, CSCE, Montreal, Canada.

IAI (2000), http://iaiweb.lbl.gov/, *Welcome to the International IAI Web Site.*

ISO (1994), *Industrial Automation Systems and Integration – Product Data Representation and Exchange*
- *Part 1: Overview and Fundamental Principles,*
- *Part 11: Description Methods: The EXPRESS Language Reference Manual,*
- *Part 21: Clear Text Encoding of the Exchange Structure*

- *Part 214: Application Protocol: Core Data for Automotive Mechanical Design Processes*

Metzger F.J (1996), *On the Usefulness of Mapping Tables in APs,* ISO TC184/SC4/WG10 N60, NIST.

Nijssen G, Halpin T (1989), *Conceptual Schema and Relational Database Design: A Fact Oriented Approach,* Prentice Hall, New York.

NIST (1980), *Initial Graphics Exchange Specification,* Version 1.0, Maryland.

NIST (1993), *Integration Definition For Function Modelling (IDEF0).*

NIST (1999), http://www.mel.nist.gov/msidstaff/sauder/SCL.htm, *The NIST STEP Class Library*.

Owen J (1993), *STEP – An Introduction,* Information Geometers.

Schenck D, Wilson P (1994), *Information Modelling the EXPRESS Way,* Oxford University Press.

Universität Bremen (1999), http://www.informatik.uni-bremen.de/~davinci, *The Graph Visualisation System daVinci*.

Wright-Patterson AFB (1985), *Integrated Information Support System (IISS), Vol V: Common Data Model Subsystem, Part 4: Information Modelling Manual – IDEF1X,* Report Number AFWAL-TR-86-4006, Volume V, AFWAL/MLTC.