AN USER INTERFACE FOR APPLICATION PROTOCOL DESIGN

Parisa Ghodous[1] and Denis Vandorpe

ABSTRACT: Over the last two decades, there have been numerous efforts directed towards developing the integrated and computerized product models. Recent advances to develop a mechanism capable to describe product data throughout the life cycle of product have led to STEP. Designing a STEP application product data model (Application Protocol) is so vast, iterative and complex that it seems necessary to develop a tool assisting this design.
In this paper, we describe an interface that allows the expert of an application to define his application's product data model in STEP. The system helps the expert to generate automatically his standard product data model. The prototype of this tool, using the hypertexts, internet and world wide web, allows the user to easily access to the integrated resources and the other Application Protocols of STEP in order to specialize the integrated resources. The possibility of verification of the uniqueness of different user's model and mapping the user's definitions to STEP definitions is provided for the user. As a case study, the metallic construction application is discussed.

1 INTRODUCTION

Today, industries and governments require an uniform product representation to support a variety of product life cycle activities and an effective, comprehensive and reliable communication mechanisms to integrate computer-aided (CAX) systems. TC184/SC4 of ISO is coordinating the efforts of different institutions and companies from several nations to arrive at one single standard for representation and exchange of product model data which came to be known by the acronym "STEP". The architecture of STEP information models is based on the three-layer structure of ANSI/SPARC: the internal or physical layer, the logical or conceptual layer and the external or application layer. The advantage of this structure is the possibility to define multiple views of application and implementation. The application layer provides relevant models called Application Protocols for various applications. Each Application Protocol contains the information for a specific application domain. Complete product information called integrated resources is given in the logical layer. The physical layer is concerned with sequential files, database and knowledge base. These information models are described by EXPRESS "The object flavored information model specification language" [Schenck and Wilson 1994]. EXPRESS supports a schama concept, that is a base for partitioning of each part or section of a part of the STEP standard. Within each schema, the primary focus is on entities which contain attributes for data and behavior.
As a part of project in which, we have tried to define a STEP product model for a crane industry, we have developed an user interface which can facilitate the definition of Application Protocols. In the following sections the methodology of design of this interface and the protoype developed with the use of NCSA mosaic, as well as the crane industry's product models defined with this prototype are described.

2 METHODOLOGY OF USER-INTERFACE DESIGN

Although user-interface design is still in part an art rather than a science, we can at least use the organized approach [Folley andVanDam 1988]. This approach is composed of the following key elements:

[1] LIGIA, University of Lyon1, Bât 710, 43, Blvd du 11 Nov. 1918, 69622 Villeurbanne cedex, France
TEL: (+33)72448000 ext 4274
FAX: (+33)72431312
Email:ghodous@ligia.univ-lyon1.fr
Html: //www710.univ-lyon1.fr/%7ghodous

- requirement analysis,
- conceptual design,
- functional design,
- sequencing and binding designs.

The details of this methodology for designing the Application Protocol development interface are described in the following sections.

3 REQUIREMENTS ANALYSIS

The first step in designing an interface is to decide what the interface is meant to accomplish. Understanding user requirements can be accomplished in part by studying how the problem under consideration is currently solved. Another successful approach is for the designer to learn how to perform the tasks in question. The objective is to understand what perspective users currently do, and, more important, why they do it. User characteristics must also be identified.

The process of Application Protocol development as shown in Figure 1, is a sequence of various tasks implying different techniques. The first task is the user requirement analysis. This analysis includes consulting the experts of the application and acquiring their knowledge, finding the entities and attributes in the scope of the application, finding the relationships between entities and obtain a consensus, common, precise, easily modifiable set of description. In case of large application the knowledge of application is classified in different sub-domains. Therefore the acquisition of this knowledge is based on the recognition of the groups of objects collected in coherent sub-domains[Grabowski, Rude and Hein 1994]. These groups are called Unit of Functionalities (UoFs). An UOF represents a specific concept or functionality from the end-users' point of view.

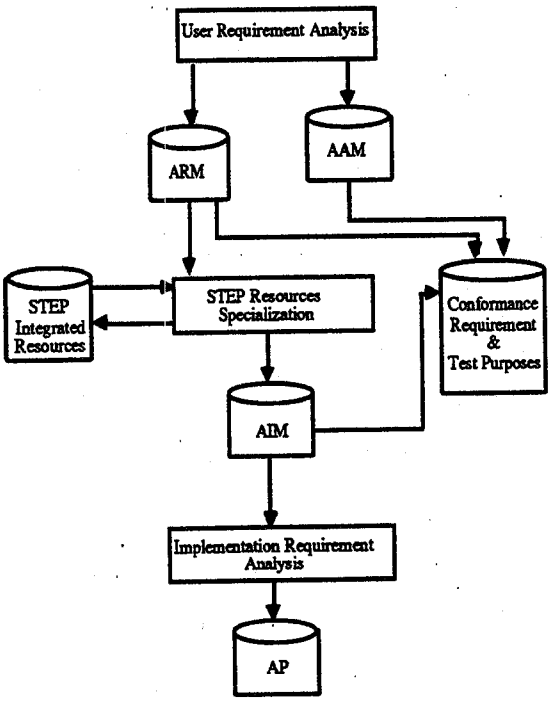


FIGURE 1. THE PROCESS OF APPLICATION PROTOCOL DESIGN

The result of user requirement analysis is documented in two different models called AAM (Application Activity Model) and ARM (Application Reference Model). The activity model AAM is a model that describes the application activities, processes and data flows, and set agreement on them. This model can be represented in IDEF0. The

data model ARM is a model of the information needed by the application written in generally accepted terminology of the application. This model can be represented in NIAM, IDEF1X, EXPRESS-G or EXPRESS.
The specialization of STEP integrated resources is provided for the ARM constructs and the result of this specialization is AIM (Application Interpreted Model). The AIM is an information model written in the EXPRESS language that uses the STEP resource models to convey the concepts and information requirements of the ARM. The specialization includes the development of mapping table between the application objects of ARM and the integrated resources, and creation of AIM EXPRESS schema. In case of the definition of the UoFs in ARM model, the mapping of these UoFs to integrated resources creates the AICs (Application Interpreted Construct). An AIC is a logical grouping of concepts that is shared by two or more AIMs. The concept of AIC permits the interoperability between the Application Protocols and shortens the time of AP development.
The process of mapping between the ARM objects and Integrated Resources, is not yet standardized in STEP. To develop a mapping table a detailed knowledge of all that is defined in the integrated resources is necessary. The correspondances between the ARM object and the Integrated Resources should be found and the cases where no Integrated Resources match exactly an ARM object as well as the cases where using the Integrated Resources creates unexpected information requirements should be solved. The implementation requirements' analysis includes the definition of the requirements on the implementation not depicted in the AIM, for example defining the schema's name to be coded in the header of the neutral file[Huah 1994]. The conformance requirements and test purposes, are derived from the user requirement analysis, the AAM, the ARM and the AIM.
The development and validation of a STEP AP is a complex, iterative process and in practice in each step, it is necessary to have a feedback to earlier step to improve the result of every step.
The AP development implies experts of the application, modelling experts and STEP experts. It is often possible to find several ways to deal with the same information requirement. The amount of editorial work is quite important and must not be neglected. The key point for a software developer is the development of the mapping table.

4 CASE STUDY

As a case study, the definition of crane's model for the design process of a specific industry has been considered [Ghodous 1994a]. The design process of the crane in the selected industry, is based on the traditional engineering drawing of parts and products. The crane consists of subassemblies and/or parts. In turn, a subassembly in a similar fashion, is composed of subassemblies and/or parts. The parts include the constituents of the tower mast, jib, crane runway, trolley pulley, trolley, hook, hoisting rope, jib tie, counterjib, counterjib ballast, operator's cab, counterweight, platform and engine. The tools used for the construction of a crane cover various manufacturing technologies as shot blasting, oxy-cutting, hack sawing, drilling, boring, milling, sheraing, painting and welding.
In the first phase of AP design, the crane expert needs to define a unique, complete, correct, minimal and understandable ARM model. The problem is that there is not a standard crane terminology and usually the process of design is not formalized in the industries and there exists the ambiguities in the meaning of terms stated in natural language. The other consideration is that there is a large number of products, and the expert should not confuse the concept of a product with the concept of an occurrence of a product. He sould provide a more generic model for a product and define the occurrence of a product by means of production management data. The prototype of the system provides the expert, a list of APs and user models defined before and supplies a help dictionary for STEP terms with examples and encourages the user to choose the equivalent STEP terms for his application objects. However there exists the possibility

of ambiguity and misinterpretation of STEP terms. In case of using the user's own definition, the system provides the facility to define his own dictionary.
The data models (UoFs) for the crane industry, during the design process may be classified as follows:
- product structure and management of product
- geometry of parts and tools
- presentation of product shape and the different views
- materials
- surface conditions
- tolerances
- draughting techniques
- kinematics
- Finite Elements
- mass property

The product model, the CSG model and kinematic model of the crane (using EXPRESS-G notation) are shown in Figure 2, 3 and 4. These models are the instances of the class UoF of the metamodel presented in the following section. EXPRESS-G is a graphical subset of the EXPRESS language, supporting the notion of schema and at a lower level of abstraction, supporting entity, type, relationship and cardinality concepts. The EXPRESS-G basic notation used in the figures includes: entities (rectangles); supertype/subtype relationships (thick solid lines); required attributes (normal lines); relationship for optional attributes (dashed lines). Additionaly, the direction of an attribute is symbolized by an open circle, where the circle represents the 'many' side of a 'one to many' relationship. In the product model of the crane shown in Figure 2, each product definition can have many different versions of the product that represent alternative designs or modifications of previous designs. Products can also be designated as belonging to specific product categories (product configuration). The product relationship can be used to define assembly relationships where the relating product represents the assembly and the related product represents an element of the assembly. Each product can be a part or a tool.

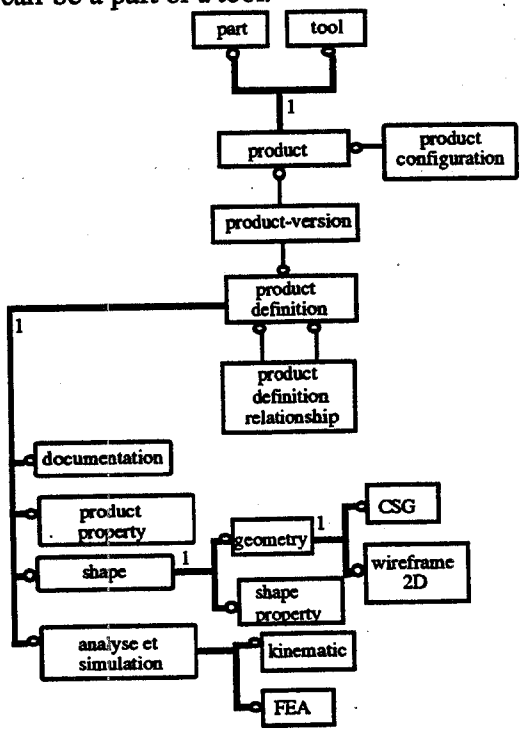


FIGURE 2. A PART OF THE PRODUCT MODEL OF THE CRANE DESCRIBED IN EXPRESS-G (GRAPHICAL NOTATION OF EXPRESS)

The geometry model of parts and tools is based on solid modelling (CSG) and wireframe 2D. The CSG model of the crane and an instance of the CSG model for a primary part are shown in figure 3.

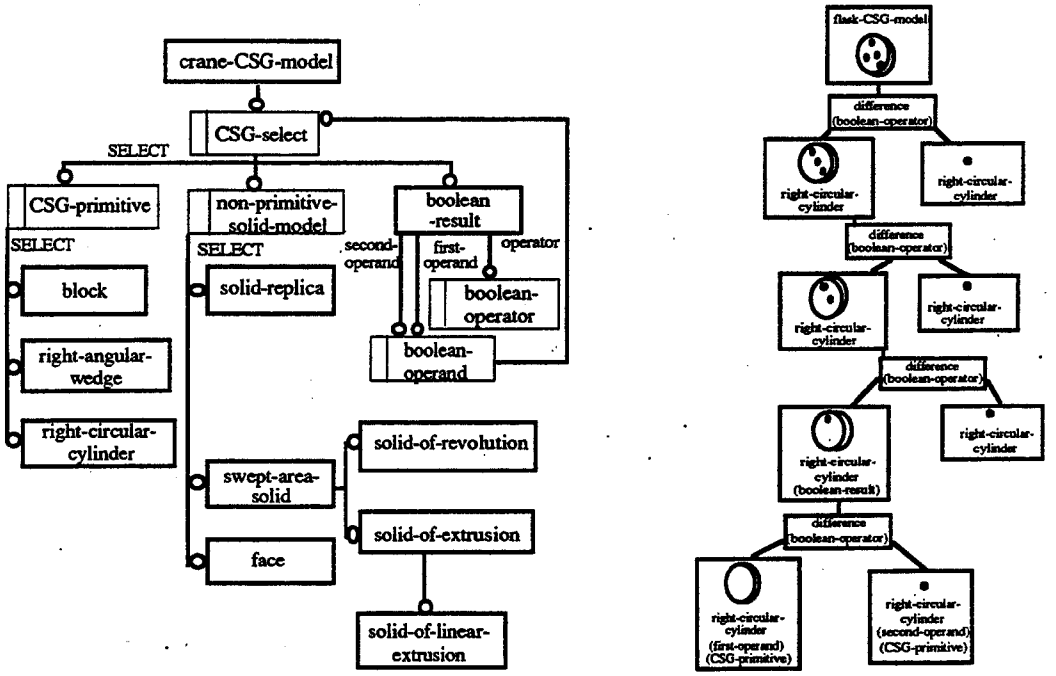


FIGURE 3. THE CRANE CSG MODEL DESCRIBED IN EXPRESS-G AND THE INSTANCE OF THIS MODEL FOR A PRIMARY PART

The kinematic model of the jib and mounting and dismounting of the crane is the most complex part of the design process. The kinematic model of the crane is shown in Figure 4.

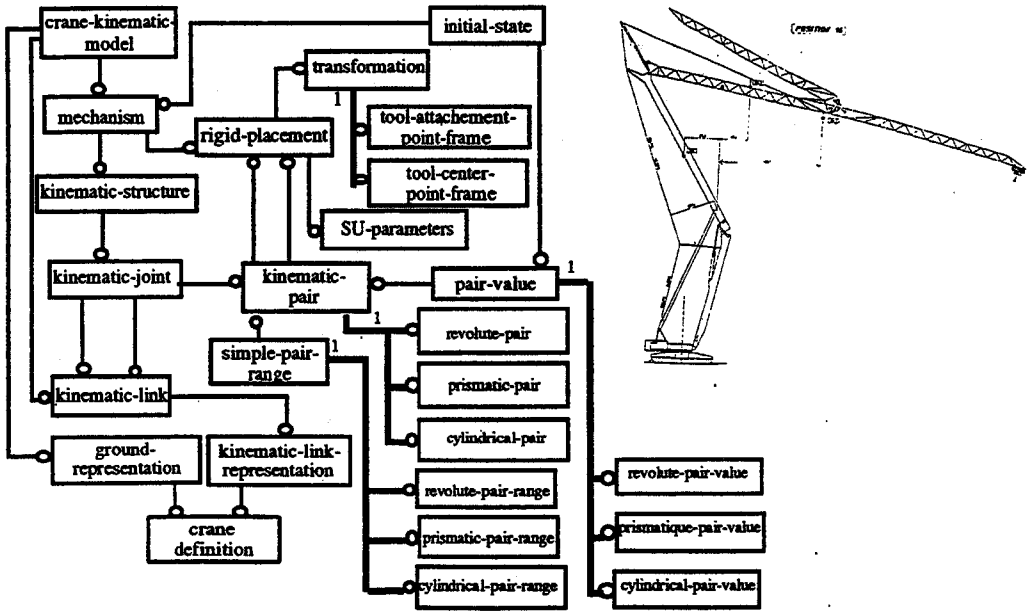


FIGURE 4 THE CRANE KINEMATIC MODEL DESCRIBED IN EXPRESS-G

The specialization of Integrated Resources is the most critical phase of AP design and the expert needs to have a detailed knowledge of the large entities' pool of STEP. STEP, is in fact, so wide that it is possible to interprete it as we want. The prototype allows the user to access and select the Integrated Resources, to introduce the constraints, to remove a choice, to omit a subtype or supertype of resource entities, to define a new subtype of an integrated resource with specific application attributes and to drop the optional attributes. The ARM entities from the EXPRESS schema are mapped into AIM entities progressing from the root entities downwards. If any ARM entity does not have an equivalent STEP resource entity, the attributes of the ARM entity should be mapped instead. This process is continued until a complete mapping of the ARM model is achieved. Thus all parts of the ARM model are eventually mapped into STEP resource entities or base type entities. In each step of specialization the system provides the STEP resource entities that are proper for mapping and provides their help dictionary and the interface for displaying and defining the necessary mapping rules. It is clear that the specialization of domains that can be formalized mathematically (geometry for example) is much simpler compared to other domains such as product management.

5 CONCEPTUAL DESIGN

When the requirements have been worked out, a top-down design is next completed by developing the conceptual design. Several alternative conceptual designs are developed and evaluated on the basis of how well they will allow users to carry out the tasks identified in the requirements definition. The conceptual design typically defines objects, properties of objects, relationships between objects, and operations on objects.
In the Application Protocol design interface, different kinds of objects are defined to represent a user product model, STEP models and interface system. The major classes are User-Model, Concept, STEP-Model, Integrated-Resources, AIC, Schema, Entity, Global-Constraint, Domain-Constraint, Uniqueness-Constraint, Function, Procedure and Constant. The classes used to represent the interface system are: Application, Icon, Window, Frame, View, Document, Menu, Command and Message. The complete list of the objects and their instance variables as well as the operations is given in [Ghodous 94b]. The semantic network of the user product model classes is shown in figure 5.

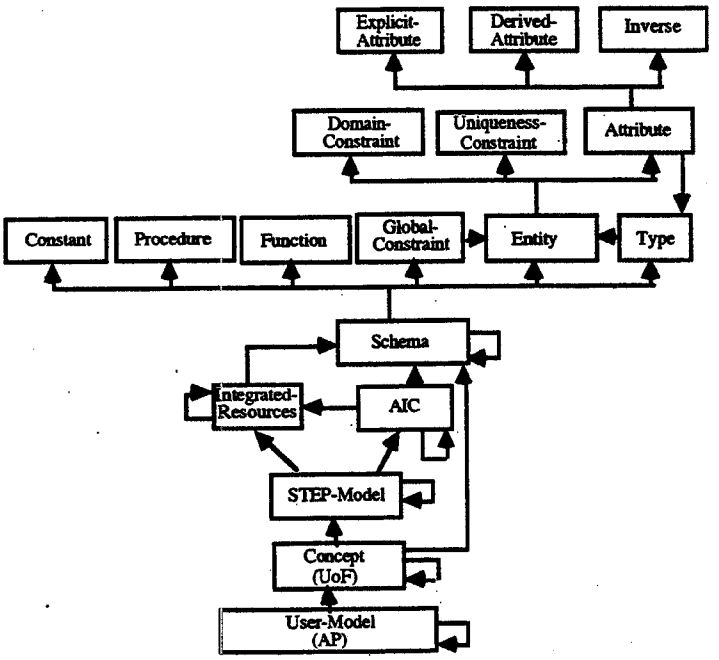


FIGURE 5. THE SEMANTIC NETWORK OF THE USER PRODUCT MODEL CLASSES

The definition of some classes is given in the following:

Class User-Model

Class User-Model is the abstraction of common characteristics of the user's application model from the user's viewpoint. A User-Model may be considered in terms of STEP as an AP. Each User-Model may be used by the other User-Models. The instance variables of this class are name, concepts, developer-name, date, version, related user models, ... The operations defined on this class are: display the User-Model, select the User-Model, delete the User-Model, create a User-Model, display, create, delete and modify the relation between user models, test the unification of the User-Models and save the User-Model.

Class Concept

Class Concept is the abstraction of common characteristics of a functional view of the User-Model. Class concept can be considered in terms of STEP as a Unit-of-Functionality. Each concept can be used by the other concepts. The instance variables of this class are name, relation, schema, STEP-Model, related concepts. The operation defined for this class are: display the Concept, select the Concept, create a Concept, delete a Concept, create, modify, delete and display the relation between concepts and save the Concept.

Class STEP-Model

Class STEP-Model is the abstraction of common characteristics of the Concept (UoF) from the STEP viewpoint. A concept may be considered in terms of STEP, as a specialization of an Integrared-Resource or an AIC. The instance variables of this class are name, Integrated-Resources, AICs, relation. The operations defined on this class are display a STEP-Model, select a STEP-Model and display the relationship between the STEP-Models.

Class AIC

Class AIC represents the common charctersistics of the AIC's concept of STEP. The instance variables of this class are part-number, name, Integrated-Resources and relation. The operations defined on this class are: display an AIC, select the AICs, display the relation between the AICs.

Class Integrated-Resource

Class Integrated-Resource is the abstraction of common characteristics of STEP Integrated Resources model. The instance variables of this class are part-number, name, schemas and relation. The operations defined on this class are: display an Integrated-Resource, select an Integrated-Resource, display the relation between the Integrated-Resources.

Class Schema

Class Schema represents the schema construct in EXPRESS language. It is the abstraction of common characteristics of a universe of discourse and it is a basis for the partition of a model. The intercommunication between the schemas is possible. The instances of this class are: name, reference-from-relations, use-from-relations, defined-types, entities, functions, procedures, global-constraints and constant. The operations defined on this class are: display the schema, select the schemas, display the relation between schemas, create the schema for non STEP model.

Class Entity

Class Entity represents the common characteristic of a thing with distinct and real existence. This thing is identifiable and specific. The instance variables of this class are name, attributes, supertypes, subtypes, domain-constraints. The operations defined on this class are: display the entity, select the entities, create an entity (in condition if it is a subtype), display the relations with the other entities and remove a subtype or supertype of an entity.

Class Global-Constraint

Class Global-Constraint represents the common characteristics of a set of constraints that apply to every entity instance. The instance variables of this class are: name and rule-body. The operations defined on this class are: display the Global-Constraint, create a Global-Constraint, add the Global-Constraint, select the Global-Constraints, display the entities that the constraint is applied to them.

Class Domain-Constraint

Class Domain-Constraint represents the set of constraints that constrains the value of individual attributes or a combination of attributes for every entity instance. The instance variables of this class are inherited from class Constraint. The operation defined on this class except those inherited is display of the attributes that the constraint is applied to them.

Class Uniqueness-Constraint

This class represents the uniqueness constraint for individual explicit attributes or combinations of them. The instance variable defined on this class is the combination of attributes.

Class Attribute

This class represents a kind of property of an entity. It identifies how the entity is represented and distinguishes the instance of an entity from the other instance of the same entity. The instance variable of the attribute is name. The operations defined on this class are: display an attribute, create the attribute, select the attribute and modifythe attribute.

Class Explicit- attribute

This class represents the attribute that have a value of the indicated type. The instance variables of this class are: name, type, optional and uniqueness-constraint.

Class Derived-attribute

Class Derived-attribute represents the property of an entity that can be computed in some manner. The instance variables are name, type, value that is given as an expression.

Class Inverse

This class represents the relationship between the entity being declared and a named attribute of another entity. The instance variables are name, entity-name, atrribute-name. The attribute-name is the name of the attribute of the referenced entity.

Class Type

Class type represents the specifications of the domains of instance values in EXPRESS language. The types are classified as simple types, aggregation types, defined types, entity types, enumeration types, generic types or select types. The operations defined on this class are: create a type and display the list of types.

Class Function and Procedure

These classes represent the function and procedure construct in EXPRESS language. The operations defined on this class are: create a function or procedure, display the function or procedure, display the liste of functions or procedures, modify and delete the user defined function and procedure.

Class Constant

This class represents the constant construct in EXPRESS language. The operations defined on this class are: create a constant, display the list of the user-defined constants and standard constants, and modify and delete the user-defined constant.

6 FUNCTIONAL DESIGN

The functional design focuses on the commandes and what they do it. Attention must be paid to the information each command requires, to the effects of each command, to the new or modified information presented to the user when the command is involved, and to possible error conditions. The commands are the messages send to the objects which activate the appropriate methods. For example, the following commands exist for the user-model object: display, add, delete, modify, copy, save, reset, test of unification, display data dictionary. The functional design is the detailed elaboration of the meanings of these operations. It defines meanings, but not the sequence of actions or the devices which the actions are conducted [Ghodous 94b].

7 SEQUENCING AND BINDING DESIGNS

The sequencing and binding designs, define together the form of the interface. These designs select an appropriate set of dialogue styles and apply these styles to the specific functionality. The seven user interface styles are WYSI-WYG (What You See Is What You Get), Direct Manipulation, Menu selection, Form fill-in, Command language, Natural language, Question-answer dialogue.
Since we have considered the user of the system, not to be extremely familiar with STEP vocabularies, we have selected a collection of menu-based interface and form fill in style that works especially well with novices. The system designer restricts the total set of considerations or choices when he designs the menus, thus the novice has fewer choices to consider and fewer possible mistakes to make, when he decides a choice. Additionally, an occasional user of the system can benefit from the structure imposed by menus. He does not need to remember each step or command used while working with the system. At each step the available commands appear and help to stimulate memory and to reduce workload. Menus are also a good choice if the person using the system is not familiar with the domain's vocabulary because menus restrict the choice set and generally reduce choices to phrases or in some cases, single words. A context sensitive dictionary for the STEP terms, for the user's application and for the system is built in the system with the use of hypertexts. The system provides the user, the facility to define his own dictionary.

8 PROTOTYPE

For the prototyping of this interface, we have used NSCA Mosaic under X11. NSCA Mosaic is a networked information discovery, retrieval and collaboration tool and World

Wide Web browser developed at the National Center for Supercomputing Applications. Mosaic provides a hyper text interface to the global internet. The architecture of prototype is shown in Figure 6. The prototype provides the expert of the application with the facility:

- to define his application model,
- to define his application dictionary or to access the built-in application dictionaries,
- to access the STEP dictionary and models,
- to create the corresponding STEP model of his application model and
- to verify the uniqueness of his application model with the other application models.

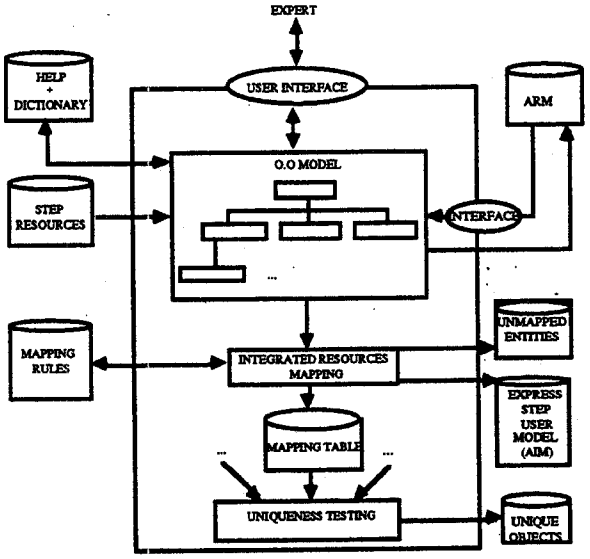
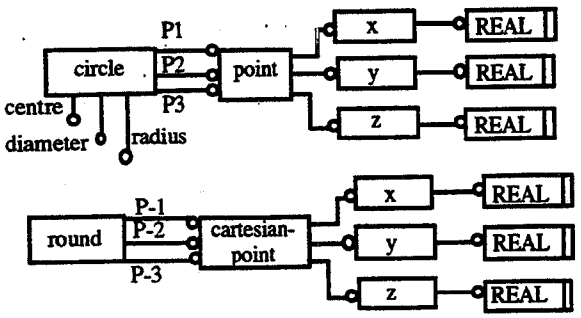


FIGURE 6. THE ARCHITECTURE OF PROTOTYPE

The integrated resources mapping module creates the EXPRESS STEP user model (AIM) and the mapping table. When a user wants to define a new entity or to give a new definition to an existing STEP entity, this module with the help of mapping rules defined in the system or defined by the user, tries to map this entity to one or different entities of STEP integrated resources. In the simple case that the entity has a one to one relationship with the entity of STEP, the system provides the possibility to take into account the user's own definition. For example if the user has to define an entity called round that is a circle defined with 3 points, the system proposes the entity circle with the basic definition used in STEP, and also shows him the different types of circle definition with the conversion rules to transform these different definitions into the basic definition. If the user wants to use his own definition, the system gives him the possibility to define his own definition with the necessary mapping rule (Figure 7) [Bailey and Mead 1993]. The unmapped entities will be reported.



Mapping Rule:

P-1.x := P1.x
P-2.x := P2.x
P-3.x := P3.x
P-1.y := P1.y
P-2.y := P2.y
P-3.y := P3.y
P-1.z := P1.z
P-2.z := P2.z
P-3.z := P3.z

FIGURE 7. EXAMPLE OF MAPPING RULE

Two different experts can define the same object in different manners, creating different ARM models for the same application. The integration and standardization of APs necessitate the uniqueness of the ARMs. The uniqueness testing module tests the uniqueness of different ARM models from the mapping tables. To test the uniqueness of several different ARM objects, their mapping path are compared. Objects with the same mapping path are equivalent objects. Therefore this module can improve the possibility to define a unique model.

The prototype permits the transfer of the ARM model defined by the expert to the O.O model of the system.

9 CONCLUSIONS

To assist the Application Protocol designers, an user interface based on object-oriented methodology has been developed. Object-oriented methodology permits the encapsulation, modularity and facilitates the conversion between the user models and EXPRESS models. The prototype of this interface with the use of XMOSAIC provides the facilites for information retrieval and communication to STEP models and the other user models. Our experience of using the prototype of this interface for the crane industry shows that this interface can capture the information related to the user's product model. Further research depends on the new works on Application Protocol methodology, the stability of the techniques used for design of AP and the stability and completeness of integrated resources. After this, it may be possible to generalize this modelling technique for other applications.

ACKNOWLEDGEMENTS

We thank the POTAIN company especially Mr Berleau for the study of the crane's design and manufacturing process.

REFERENCES

Bailey, I., and Mead, M., 1993, "EXPRESS-M: A Schema Mapping Language", In Proceeding of Third Annual EXPESS User's Group, Berlin, Germany.
Danner, W. F., Sanford, D. T., Yang, Y., 1991, STEP Resource Integration: Semantic & Syntactic Rules, ISO TC184/SC4/N80.
Fulton, J., A., 1992, "Technical Report on the Semantic Unification Meta-Model Volume1: Semantic Unification of Static Models", ISO TC184/SC4/WG3 N175.
Ghodous, P., 1994a, "Protocole d'Application de la conception de la grue POTAIN", Rapport interne LIGIA/LISPI, Université Lyon 1, France.
Ghodous, P., 1994b, "Un Générateur de modèle STEP", Rapport interne LIGIA/LISPI, Université Lyon 1, France.
Grabowski, H. , Rude, S. and Hein, K., 1994, "Core Data for Automative Mechanical Design Processes (AP 214)", In Proceeding of European Product Data Technology Days '94, Revue internationale de CFAO, Hermes. Vol 9- n° 3/1994, pp. 413-433, Paris, France.
Huah, P., 1994, "Using STEP Technology : the expertise gained through AP development", In Proceeding of European Product Data Technology Days '94, Revue internationale de CFAO, Hermes. Vol 9- n° 3/1994, pp. 391-400, Paris, France.
ISO CD 10303 Product Data Representation and Exchange, International Organization for Standardization, Subcommittee 4.
Kramer, T. R. , Palmer, M. E. and Bernard Feeney, A., 1992, "Issues and Recommendations for a STEP Application Protocol Framework", NIST.
Mckay, A., Bloor, M. S., and Owen, J., 1994, "Application Protocols : a Position Paper", In Proceeding of European Product Data Technology Days '94, Revue internationale de CFAO, Hermes. Vol 9- n° 3/1994, pp. 377-389, Paris, France.
Owen, J., 1993, "STEP an introduction", Information Geometers Ltd, UK.

Schenck, D. A., and Wilson, P. R., 1994, "Information modelling: The EXPRESS Way", Oxford University Press, New York.
Tsichritzis, D., and Klug A., eds. 1978, "The ANSI/SPARC DBMS framework: report of the study group on data base mangement systems", Information Systems, Volume 3, pp.173-191, published by Pergamon Press Ltd, Oxford, UK.
Wilson, P. R., 1992, "Processing Tools for EXPRESS", NIST server.