The Distributed Information Service in the VEGA project

An approach towards the harmonisation of STEP, SGML and EDIFACT information standards for the support of integrated and distributed construction project information systems.

Philippe Debras, Alain Zarli, Virginie Amar, Patrice Poyet
Centre Scientifique et Technique du B,timent
290, route des Lucioles
B.P. 209
06904 Sophia Antipolis cedex

Keywords: *Distributed Information System, STEP, SGML, EDIFACT.*

Abstract

Over the past two decades, the complexity of engineering projects has increased. In such a context, having access to the right information at the right time is often crucial. In the area of Information Management, numerous efforts have already been made under the auspices of various major institutes towards the elaboration of Information Standards. Whether it be STEP for Product Data representation and exchange, SGML for documentation structuring and support or EDIFACT for Electronic Data Interchange, these standardisation efforts all address complementary aspects related to the representation and communication of engineering project information. However, both at the conceptual and technological levels, end-users nowadays face a lack of integration among these various existing or emerging Information Standards.

Building upon the on-going specification and implementation of a CORBA based middleware layer allowing the distribution of any STEP SDAI models, the **Distributed Information Service** of the VEGA European Esprit Research project addresses the integration and distribution of construction project information systems at two different levels. First, the conceptual level is concerned with the elaboration of STEP EXPRESS product data models supporting SGML documentary models and documents but also EDIFACT messages models and messages. The implementation level is then concerned with the development of dedicated converters supporting the back and forth translation of SGML documents and EDIFACT messages towards the STEP format, finally providing a remote access to any kind of information for all the actors involved in a construction project.

1. Introduction

Over the past two decades, the complexity of engineering projects has increased markedly along with their scale, strengthening constraints (quality, duration, cost, regulations), number of actors involved, volume of information required and produced. In such a context, having access to the right information at the right time is often crucial. In the Architecture Construction and Engineering field, large projects require the involvement of many body entities (client, architect, design engineers, specialists from different technical disciplines, technical controllers, construction companies) sitting at various locations, with different views and needs on the project. On the information side, numerous documents of diverse nature are involved in the construction process. Some of them such as building codes, examples of technical solutions, computation rules, define the legal context of a project. Others like master plan drawings, technical specification documents or bills of quantities are generated by the engineering activities and often have a contractual importance. Drawings are the straightforward media to convey most of the information needed by the construction companies. They include a lot of information that can hardly be put into words. Textual documents are complementary to drawings. They traditionally support the engineering aspects of the project description. In line with all this mass of data, decisions with the greater economic consequences are made in the many stages of the project life cycle.

Thus industrial processes are characterised nowadays by an intensive use of Information Technologies which, up to know, hardly met end-users expectations whatever the industrial sector might be. Despite the obvious benefits they provide, there remain however many frustrating problems. Some of them are associated with the use of non-standards or proprietary systems. Partly for competitive reasons, partly for functionality reasons, the various formats used by most of the manufactured systems are often incompatible. Information, even if already existing on a numeric storage unit, often needs to be retyped from a format to another. Another point is the fact that electronically stored information may not be presented in an understandable way and may be difficult to retrieve. The incapacity to support information distribution and communication on a wide basis is a recurrent problem too.

All these problems can be summarised under the following set of questions: how to organise information so that retrieval is easy while remaining independent of any application ? What use of the existing Information Standards can be done ? How to design an Information Technologies platform that will support efficiently the Information Management Infrastructure ? How to implement this Infrastructure and fill it in once implemented ? How to communicate information from the Information Management Infrastructure towards the various users working places ? These are some of the obvious keys of tomorrow industrial competitiveness and success.

In the area of Information Management, numerous efforts have already been made under the auspices of various major institutes towards the elaboration of Information Standards. Whether it be STEP [SGML] for Product Data representation and exchange, SGML [SGML] for documentation structuring and support or EDIFACT [EDI] for Electronic Data Interchange, these standardisation efforts all address complementary aspects related to representation and communication of engineering project information. However, both at the conceptual and technological levels, end-users nowadays face a lack of integration among these various existing or emerging Information Standards.

The need for a global information infrastructure that will federate these islands of standardisation becomes crucial. Such an infrastructure will provide a common basis through which information related to the elaboration of products will be integrated in an homogeneous way whatever its nature might be (product data model instance, marked up document, electronic message).

At the technological level, as a result of the third workpackage of the VEGA project [VEGA95], any kind of STEP EXPRESS product data model will be distributed thanks to a CORBA based middleware layer (COAST) [Koet97][Amar97], and advantage of this situation can be taken for the distribution of other types of standard based data provided that they be regarded as product data model as well. This is quite certainly a wise strategy for the integration and distribution of SGML based documents and for the wider exchange of EDI messages in a product model data based environment, leading to what we can call a **Distributed Information Service** (DIS).

After an overall presentation of the VEGA project, a general description of a DIS is offered. The VEGA DIS using SGML and EDI standards is then described. Finally, some conclusions are brought to this on-going research effort.

2. The VEGA project

2.1 Vision and objective

As industry is becoming increasingly integrated at many levels, between departments of a single company as well as between companies, there is a deep need for a technology of distributed systems supporting groups of workers geographically spread over the world but working together in order to achieve the same goal, based on standardisation efforts for the interchange of information between heterogeneous systems. The VEGA (*Virtual Enterprises using Groupware tools and distributed Architectures*) project [VEGA95] wants to address these issues and aims at establishing an information infrastructure which will support the technical activities and the business operations of the upcoming Virtual Enterprises.

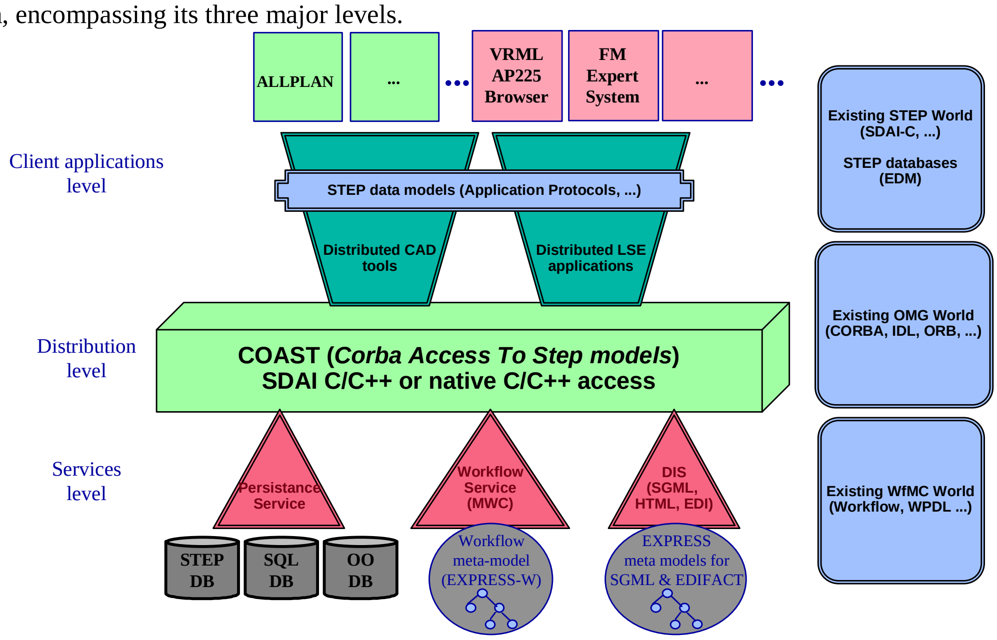
Such an objective has to be achieved through the development of distributed architectures, in good compliance with on-going standardisation activities like STEP in the area of PDT (Product Data Technology) and international industrial groupware specifications coming for example from the Object Management Group like CORBA [OMG95a]. VEGA is supposed on the one hand to explore the distribution and concurrent access to STEP databases, and on the other hand to design an extension of the current Standard Data Access Interface (SDAI) to product model data which will support as a natural mechanism the distribution of product data by means of an adapted object broker technology. Building upon existing results, VEGA will show that what is required as the backbone of the information infrastructure is a redesigned architecture to publish product data not in a client/server mode but in a new all distributed product data approach. Thus, VEGA will establish this generic backbone for a new communication era based on the distribution and remote access to STEP product model data.

The emphasis on creating an infrastructure, based on product data standards on the one hand and on industrial specifications on distributed environment and also concurrent engineering through workflow mechanisms on the other hand, should satisfy the broad expectation shared by the end-users and expressed as a consensus about the need to support distributed working. Such a specific information infrastructure will allow LSE applications to communicate easily and efficiently, thus speeding the work of designing and manufacturing a complex product and consequently reducing the time for bringing it to the market place. VEGA will focus on very interesting business cases in the Large Scale Engineering industry, which is a typical example of a fragmented industry where the business relationships last only for the duration of the projects. The main expected benefits for the virtual enterprise are the following:

- Integration of heterogeneous and distributed environments of various companies,
- Incorporation of legacy engineering/manufacturing information to preserve existing investments,
- Promote new forms of working by LSE business re-engineering (parallel working, workflow),
- Find new routes to pragmatic linkage of relevant emerging standards, and encourage the use of new standardised methodologies and tools.

2.2 General presentation of the VEGA platform

As previously mentioned, the developments in VEGA have to be done with a strong connection with current concepts and advances in the field of object-oriented technology and the elaboration of international standards by OMG such as CORBA [MoZa96][OHE96][Sieg96] and the IDL (Interface Description Language) language [OMG95a]. Such a distribution technology, allowing transparent access to information in the context of heterogeneous networks and environments, will be linked to STEP conceptual representation in order to constitute the central component of the VEGA platform (named COAST: *COrba Access to STEP models*) [Koet97]. The figure below proposes a global view on what will be the VEGA platform, encompassing its three major levels.



The first level is the one of client (end-users) applications, which can be potentially linked to the COAST backbone, as CAD/CAM applications, STEP (AP-based) applications, AI-based applications, PDMS (Product Data Management Systems) applications, and so on. These applications are based on STEP conceptual modelling (as Application Protocols like AP214, AP225 or AP228 for example), and are plugged to the COAST either through IDL interfaces (*Interface Description Language* - a system independent language defined by the OMG for specifying interfaces to objects) or using the native COAST API.

The second level is the one of distribution specification and concurrent access to data, and relies on the COAST. This level is the most fundamental as it can be considered as the spinal chord of the whole platform. Besides the exchange of data models through physical files (SPF - *STEP Physical File*) [ISO94a] and the concurrent access to a centralised database using a standardised API (SDAI) [ISO95], the COAST introduces to a third new means to access STEP information (i.e. described in EXPRESS [ISO94b]) well adapted to fragmented and remote information environments. The COAST is a service built upon CORBA, which conforms to the OMG's OMA (*Object Management Architecture*), and can be viewed semantically as a superset of the SDAI as it extend its functionality at two levels:

- It offers support for transparent distribution of information, so that the end user never needs to know on which server connected to the network stands the information he wants to access,
- It offers a true concurrent access to information, mainly through transactional behaviours (the management of transactions is done at the level of the COAST middleware itself).

Thus the COAST includes interfaces, protocols and services for the exchange of data between networked information systems, and is a fully object-oriented access method hiding to the end-user all the issues related to distribution, information search, heterogeneity and storage mechanisms. The COAST will rely on an ORB and will have to include various architectural components of CORBA 2.0, along with the use of some OMG Object Services [OMG95b]. It will also provide a native COAST API detailing a set of functions potentially accessible through some specific programming language binding by any application plugged to the COAST.

The third level is what can be called the " Server applications " level. It will offer a set of models and value-added services for specific purposes like workflow or documentary aspects. Thus, not only the COAST will allow access to distributed STEP-compliant technical manufacturing data, but it will lead to potential extensions to other sorts of data related to other standards as EDI, SGML, WfMC and others, provided the fact that these data could be represented as product model data as well. This level will especially characterise the synergy of distributed object computing with other existing or emerging technologies as document management systems, product management systems, workflow applications, and so on. This level also includes persistent long-term storage, as one of the main features of the virtual enterprise. Thus, it has to cover a wide range of DBMS (relational and object-oriented, as well as specific STEP databases which will be the focus in the VEGA project), to ensure persistent access to information objects independently of the possible distribution of the databases, and to offer a powerful high level language for query and updating of the information through the COAST.

A last point to mention here, which is one of the main ideas in VEGA, is that EXPRESS can be used as the basis for providing means of accessing EXPRESS-based information in a distributed environment. As EXPRESS is an information modelling language, workflow or documentary concepts, for instance can be designed in EXPRESS and then managed through the COAST. Moreover, and despite its name, the COAST does not manage only access to STEP models as defined in the SDAI: the only constraint is that information should be modelled by an EXPRESS schema.

3. Distributed Information Service: a general description

The current section structures the description of a Distributed Information Service through a User Perspective, an IT View and a Workbench. This abstraction effort leads to a generic approach to Distributed Information Services which can be reused to specify and implement other distributed services.

3.1. User Perspective

The User Perspective is certainly the right place to begin with when investigating the relevance of a Distributed Information Service. This User Perspective should state the existing working Context in relation with the foreseen service, the Objective the service is targeting and finally a user scenario evidencing the interest of the service. Starting point of the User Perspective, the Context depicts a working snapshot including the various actors involved at their respective location site and with their respective expectations. Such a description should evidence the limits of the current business processes with respect to at least one corner of the success triangle: quality, cost and time. Once the problem identified within the context presentation, the Objective clearly states the target the Distributed Information Service is willing to meet. Participating to a re-engineering of the concerned existing business process, the Distributed Information Service allows a renewed implementation of such a process. The User Scenario captures the description of this new implementation, enhancing the major changes for the various actors with respect to their working habits.

3.2. IT View

Complementary to the User Perspective, the IT View specifies and implements a dedicated software architecture that meets the needs of the User Scenario. Such an architecture relies on existing or emerging Information Technologies and Standards, on the representation of the data and operations that formalise the description of the Distributed Information Service, finally on necessary tools that support its implementation.

The Architecture stands for the global software structure of the Distributed Information Service in terms of software components and systems involved and interfaces allowing to interoperate with each other. The Architecture encompasses both the Specifications that will govern its implementation and the Implementation itself. Specifying the Architecture mainly means identifying its components and describing them through the role they play in the Architecture, the other components they are in relation with, the information they require, process and deliver. These Specifications stand in fact as a first overview as they are refined for Data Representation, Distributed Service and Supporting Tools within dedicated chapters. The Implementation of the Architecture relates to the gathering of the various software components that are put in play altogether thanks to devoted interfaces. Therefore this effort ends the implementation of the IT View as a whole.

Various Information Technologies and Standards stands for potential candidates to support the representation and communication of the Information involved in the Distributed Information Service. A brief introduction to these Standards pointing out relevant references is worth developing to assess this technological support.

Beyond the Architecture that allows it to be run, a Distributed Information Service is mainly defined through the description of the data it processes and the service it offers. Both data and service have to be specified and implemented within the course of the development of the DIS. Specifying the representation of the data involved by a DIS starts with the identification of its nature and continues with a description of its structure. A Conceptual Modelling effort is necessary to build from the former specifications a formal description of the Information. Dedicated modelling languages, mainly the STEP EXPRESS language, are natural candidates to capture such a description. Specifying the DIS relies in defining the service offered to the client in terms of elementary operations that should be activated to invoke the service as a whole. Formalising the DIS stands for capturing the aforementioned elementary operations through a dedicated language (e.g. IDL). Such a formalization encompasses the naming of the operations themselves but also the naming and typing of their parameters.

If the former operations define the client view of the DIS, there is also a need for Supporting Tools that will support on the Server side the implementation of these operations. Such Supporting Tools have to be described first in terms of functionality, structure, supporting Information Technologies and Standards, application programming interfaces. The Implementation of the Supporting Tools implies the software encoding of the necessary features that complies with the specifications and allow the tools to run.

3.3. Workbench

Elaborating a Workbench for a DIS starts by refining the User Scenario to a dedicated demonstration case, continues with the identification and integration of necessary Material and Software Systems and ends with a evaluation and demonstration of the running Supporting Tools and the DIS itself.

The User Scenario is usually described at a certain level of genericity where the various actors, data and systems involved are not explicitly identified. The Demonstration description therefore builds upon this scenario and derives a demonstration case at a satisfying level of details. Material involved in the demonstration case has first to be identified, then to be translated, if required, to a format that suits the DIS needs. Among the various criteria that should govern the choice of Material to support the evaluation and demonstration of the DIS, availability, relevance, available format stands for the most important ones. As identified data may not be available in a format that suits the DIS specifications, manual or automated conversion of this material has to be operated.

Bench marking provides both a user-oriented and technical evaluation of the DIS through the evaluation of the various Supporting Tools and the evaluation of the DIS itself. Supporting Tools are seen as interchangeable software components participating to the implementation of the DIS. Such components are worth being evaluated by themselves before the evaluation of the complete DIS is performed. The Bench marking of the DIS as a whole assesses its user-oriented and technical performance. A first part of this evaluation is usually run at the integration site itself. However, the on-user-site evaluation of the DIS is also crucial and should not be under-estimated.

4. A DIS using SGML and EDI standards

4.1. User perspective

The context of the DIS is investigated according to complementary aspects regarding the information handled within a LSE project information system. The various types of information are of great importance, but also the management of these types of information, the related security requirements and the communication media.

The information that is produced and exchanged within the frame of a LSE project is diverse and encompasses among others product data information, administrative and commercial information and documentation. Product data information captures thanks to a conceptual modelling approach the information intrinsequely associated with products all along their life cycle. Documentation usually stands for particular presentations of product data information. Such presentations are generated from a given viewpoint and with respect to a specific purpose. They encapsulate product data information within dedicated physical frames, whether product data is taken as is or processed (i.e. agglomerated). Bills of Quantities, Technical Specifications documents are examples of documentation information. Information usually exchanged thanks to EDI messages is more often administrative or commercial information associated with products. Orders, deliveries statements and invoices are typical examples of such information. Here again, the difference rather stands in the formatting presentation and supporting medium than in the data itself and the frontier between the two is not clearly defined.

The information standards that support the various aforementioned types of information rely upon conceptual and technical approaches which are not so far compatible which each other. Of course, it is always possible in an EDI message to refer to a given product by using an identifying property of the product. However, there is not, as of day, any fine-grain integration of technical, documenting and commercial data at the STEP, SGML and EDIFACT Information Standards level, despite the efforts and current on-going works of joint working groups in relation with the associated standardisation bodies.

In the context of administrative and commercial data interchange, the necessary confidentiality associated with the data which transits through world-wide spread physical networks is crucial. EDI addresses this security aspect thanks to the interchange contract binding the partners with respect to security and juridical barriers. Building upon the specific structure of messages, authentication procedures and encryption mechanisms, the interchange contract allows, when a problem occurs, to clearly establish involved liabilities. Moreover, existing software tools dedicated to the administration of such issues can ease the identification of the problem.

In comparison with other media for computerised exchange, the Internet is on the way to reduce transmission costs and allow a large scale communication. In particular, the Internet integrates fundamental technologies (i.e. MIME for *Multipurpose Internet Mail Extensions*), and presents a same common interface to data to be potentially shared or exchanged by all the applications on a networked environment. However, the Internet is characterised by a general lack of security. For instance, no clear concept of public and private data is supported. As for performances, transmissions and access periods are not as of today fully satisfactory, even if this situation should evolve with awaited information highways.

In the LSE and Building Construction industries, large projects appear to be more and more delocalized. They increasingly involve more actors sitting at various sites and requiring information on the project whether it relates to product data, documents or administrative information. Keeping all this information accessible, when required and if authorised, for each of the partner all along the project life cycle becomes a crucial challenge. Therefore from the various aspects explicited in the former chapter, the VEGA DIS essentially focuses on the management and distribution of the different types of information (product data, documents, messages) associated with LSE projects in the LSE and B&C industry. The undertaken strategy towards such an information management (storage and access) relies on the objectification of the SGML and EDIFACT standards at the document and message model and instance levels in order to reach a sufficient level of genericity, the distribution of the information being operated thanks to the COAST middleware layer which will furthermore provide security layers and communication media.

Assuming the availability of the VEGA DIS, the user scenario presentation is broken down through 7 main steps which are listed thereunder:

1. the elaboration of project documents and messages in SGML and EDIFACT formats ;
2. the delivery of these documents and messages to their respective addressees ;
3. the conversion of the documents and messages towards a STEP representation ;
4. the storage of these STEP based documents and messages within the distributed project information system ;
5. the retrieval of these documents and messages from the distributed project information system ;
6. the conversion of the documents and messages back to their respective SGML or EDIFACT formats ;
7. the visualisation or edition of these documents and messages on the actor site.

Step 1 addresses the elaboration of documents and messages as a very common activity within engineering projects. Such documents could be Bill of Quantities (BOQ) or Technical Specification Documents (TSD). On the message side, one can mention trading or commercial messages that are exchanged between the project companies and their external suppliers for instance. To play within the VEGA DIS scenario, documents have to conform to the SGML format and messages to the EDIFACT format, either they are elaborated thanks to appropriate editors or they are converted thanks to dedicated converters.

Coming to the delivery of the documents and messages, one should distinguish here between the delivery itself (step 2) and the storage of the documents and messages within the distributed project information system (steps 3 & 4). The delivery is an explicit transmission of the documents and messages towards a given list of recipients. EDI transactions implement the delivery of trading messages for instance. As for the storage, the document or message is made persistent and retrievable within and from the project information system, but the access initiative is one the recipient side. The two approaches can of course be combined: a document being stored and a message being sent notifying to the various partners the existence and location of the new document in the project information system.

Storing a document or message within the distributed project information system requires first to convert the SGML document or the EDIFACT message towards a corresponding STEP format (Step 3). The result of such a conversion is a STEP model that captures the document or message content.

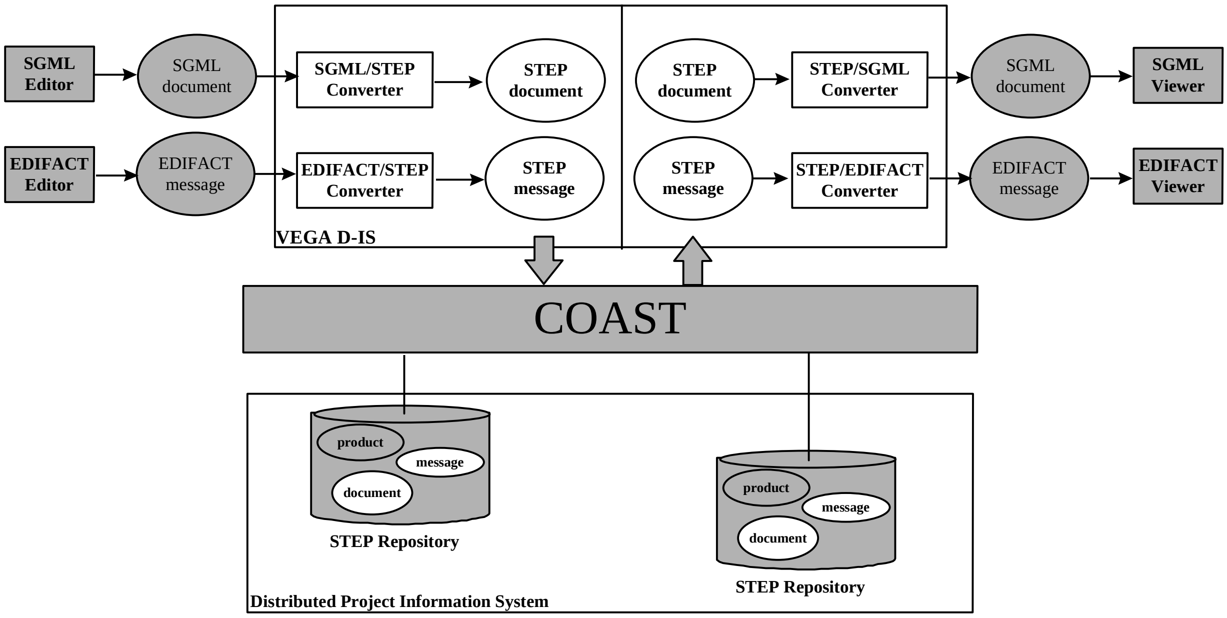
The STEP based document or message is then stored thanks to the COAST API within the distributed project information system (step 4) from which it can be retrieved later on (step 5). As the document or message is retrieved as a STEP model, it has to be converted back towards its initial format (step 6) before it can be edited or visualised within a dedicated editor or viewer.

4.2. IT view

4.2.1. Architecture

The current section aims at identifying the various components of an architecture that implements the VEGA DIS user scenario as depicted in the previous chapter. An overview of the offered architecture is given through the following picture.

Such an architecture allows the management of SGML documents and EDIFACT messages in the context of a STEP based environment distributed thanks to the COAST middleware. In this architecture, components that appear with a grey background colour participate to the implementation of the scenario without being part of the VEGA DIS development itself. Among these components, one can mention SGML and EDIFACT editors that allow the elaboration of SGML documents and EDIFACT messages, the COAST middleware, the associated STEP repositories, and finally SGML and EDIFACT viewers. As for the software components that are really part of the VEGA DIS implementation, one can mention converters that translates documents and messages between SGML/EDIFACT and STEP representation back and forth.



4.2.2. Information Technologies and Standards

Information Technologies and Standards that enter the scope of the current DIS encompass mainly STEP, SGML, EDI, CORBA and the WEB technologies. These standardisation efforts and technologies form the basis of software architectures that will meet most requirements in terms of Representation, Distribution and Communication of Information. Building upon them will allow the development of portable and interoperable software components that are to meet end-user expectations.

STEP

As one of the reasons of the rising success of Product Data Technologies (PDT) lies in the development of STEP (and especially the EXPRESS language), this standard has been chosen in the VEGA project as a basis for PDT to develop the information infrastructure of the Virtual Enterprise. STEP, the STandard for the Exchange of Product Data, also referred to as ISO 10303, has been under development by the ISO Technical Committee 184 (Industrial Automation Systems and Integration), Sub-Committee 4 (Industrial Data and Global Manufacturing Languages), and is a major stage towards efficiency to integrate all the possible actors into the product development process, and to provide satisfying means for data representation and exchange.

CALS and SGML

Computer-aided Acquisition and Logistics Support (CALS) is an initiative launched by the US Department of Defence which faced high overheads with the usual paper-intensive administration of weapons systems. Thus to be able to co-ordinate between suppliers and government the information that is generated during the lifetime of weapons systems (e.g. design, manufacturing, support and maintenance), the US Department of Defence imposed a number a standards targeting electronic representation, storage and distribution of technical product documentation. The ultimate objectives were an increased productivity and cheaper update of technical manuals, and an on-line access to maintenance process information towards substantial savings in the weapons-systems life cycle costs. Nowadays CALS may serve as a global model for Information Management Systems that aims at supporting complex technical products.

One of the standards imposed within the CALS framework is SGML (Standard Generalised Mark-up Language - ISO 8879) issued by the International Organisation for Standardisation (ISO) in December 1986. SGML first purpose lies in document representation: the standard formalises how documents may be split into on the one hand a part containing text, practically a SGML document instance, and on the other hand a part describing its structure referred to as a Document Type Definition (DTD). Specific programs called SGML parsers analyse and check that the mark-up in the document satisfies the rules defined by the DTD. Due to this structured form, SGML documents can be analysed by programs. For instance, retrieval facilities willing to access the title of a document or of one of its chapters which can hardly be applied on unstructured text may then be used. Furthermore SGML is not associated to any particular word-processing system. This asses portability of SGML conforming documents on many systems.

EDI and EDIFACT

The Electronic Data Interchange (EDI) was born in the seventies from the will to favour the growth of international trade by making its process easier, and to improve communications between companies. It was useful as well to fill the growing gap between the exchange rapidity of physical goods flow (taking faster and faster means of transport) and the slowness of documentation carriage, conveying the necessary information for the right circulation of these goods: this information should be at disposal rapidly and on reliable way in order to avoid any locking or delay and so over cost in the carriage of goods. The EDI is also an element making easy the moving path of goods. In fact several professional sectors, before the setting of universal exchange norm, had already defined their own " proprietary " EDI: SWIFT and its inter bank messages, American truck drivers, etc.

A first step has been the transposition and the processing under EDI messages of trading document (orders, bills, ...). But after several years of preliminary developments and experimentation, it's a UNO branch, the European Economical Commission - Trade division - working group 4 - expert group number 1 - who succeeded to bring closer the numerous existing standards inside a single norm: EDIFACT-UNO. This organism (UN/ECE/Trade/WP.4/GE.1), pointing out to make easier the internal trade process by using electronic exchange data, proposed to ISO to normalise its work results, as the generic feature of the EDI would make it obviously usable on the international matter as well as national and in a lot of sectors. Currently, the EDI is actually more used inside countries than for international trade.

The transfer of data is not something new, but the EDI is a new method through its concepts and characteristics. Before the emergence of an electronic data exchange norm, the exchange was made by private agreement between both partners. The progress of computer science, office automation communication and telecommunication, and above all the adoption of the EDIFACT norm as an international language used all over the world, have favoured the EDI development. This common language allows other partners to come in and decrease the interface problems related to enterprise business software. By realising an automated transfer from beginning to end, the exchanged data are directly usable by software.

The pre-established and normalised messages follow the EDIFACT norm (*Electronic Data Interchange For Administration, Commerce and Transport*). It is a syntax norm used as a basis for common international language for computerised data exchange for administration, Commerce and Transport. EDIFACT was adopted by United Nations and took over by ISO (ISO 9735) in 1987.

CORBA

The Object Management Group (OMG) works since 1989 on the creation of standards targeting the interoperability and portability of distributed object-oriented applications. One of its major achievement is the Common Object Broker Architecture (CORBA) specification, an interface description language (IDL) and a complete set of services (e.g. Naming, Events) that must be provided by compliant ORBs.

The Interface Definition Language (IDL) allows a neutral description of the operations that a client might invoke from a server. Such interface definitions fully specify each operation through the description of each parameters, mainly their type and their typology (in, out, in-out). Client applications are to be developed on the basis of this interface only, i.e. without further knowledge of their implementation on the server side. The IDL syntax is quite similar to C++ (e.g. data members, exception) although new keywords (e.g. one-way) are introduced to support distribution concepts. Clients are not written in IDL but in languages for which mappings have been defined (e.g. C, C++, Ada and Smalltalk).

WEB technologies

The Internet has to be nowadays more and more acknowledged as the common medium to support communication facilities within the development of client-server applications that deserved the larger audience. With that context, the WEB technology builds upon the HyperText Transfer Protocol (http) to provide the users with high-level graphical applications independent of the underlying client platform. Furthermore the JAVA language simplifies the development of WEB applications through the power and the flexibility of a real implementation language. The Intranet perspective en favours the use of the WEB technologies on Local Area Network on a (extended or Virtual) company scale whether it be real or virtual.

4.2.3. Data representation

Capturing Documents and Messages within a STEP based Information System requires the elaboration of conceptual models that support their representation. Such a conceptual effort ends with EXPRESS schemas that formalise the conceptual models according to the EXPRESS syntax. The conceptual models that are necessary for the support of the current DIS address on the one hand Document Models and Document Instances, on the other hand Message Models and Message Instances. The conceptual modelling effort offered in this article is limited to Document Models and Message Models, the Document Instances and Message Instances being still under development.

As for Document Models, the objective is to capture the semantics of the SGML DTD description language throughout the identification and description of its main concepts. A presentation of this language therefore stands for the specifications related to the objectification of Document Models. In the same way, the presentation of the EDIFACT Message Model description will serve as specifications of the Message Model objectification.

SGML Model

The SGML language provides means for encoding the structure of documents thanks to the DTDs capability. In a DTD, each documentary concept is represented as an *element* described by a model of contents and an optional list of *attributes*. A model of contents defines authorised assemblies of elements thanks to operators:

- ELT? means 0 or 1 occurrence of element ELT ;
- ELT* means 0, 1 or n occurrences of element ELT ;
- ELT+ means 1 or n occurrences of element ELT ;
- ELT1,ELT2 means a sequence of elements ELT1 and ELT2 ;
- ELT1 | ELT2 means exclusive or between ELT1 and ELT2 .

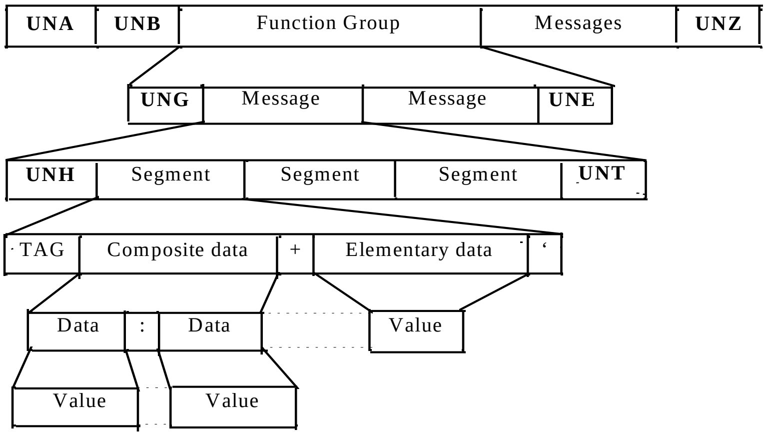
Here is a small part of a DTD that formalises a document structure for Technical Specifications documents.

```
<!--------------------- Technical-Specifications DTD ---------------->
<!ELEMENT technical-specifications - - (general-information, corpus) >
<!ATTLIST technical-specifications id ID #REQUIRED >
<!ELEMENT general-information - - (title, subject, date, version?) >
<!ELEMENT (title | title1 | title2 | title3) - - (title-number, title-
label) >
<!ELEMENT (title-number | title-label) - - #PCDATA >
<!ELEMENT corpus - - (division1+) >
<!ELEMENT division1 - - (title1, paragraph*, division2+) >
<!ELEMENT division2 - - (title2, paragraph*, division3) >
<!ELEMENT division3 - - (title3, paragraph*) >
<!ATTLIST (division1 | division2 | division3) id ID #REQUIRED >
...
```

EDIFACT Message Model

Each unit which is exchanged in an EDIFACT connection is called an *interchange*. Several interchanges may take place during a connection. Each interchange consists of one or more messages. A *message* which has the EDIFACT syntax is an ordered sequence of *segment*. At this stage of the presentation of the EDIFACT specification, any EDI business transaction has potentially its own structure which is of course conveyed with the message itself through the description of the segments. However, in order to simplify the use of the standard, building upon the EDIFACT neutral and architecture independent syntax rules, various EDI message prototypes are been developed at the international level, which federate the many users of the EDI technology. Such EDI message prototypes address the various kinds of business transactions, invoice, purchase order or request for quote for instance. Therefore, the EDI message which is effectively exchange throughout a business transaction appears as an occurrence of such a EDI message prototype, a transmission format of which is agreed by all partners over the interchange contract. Such an occurrence will convey for instance the identifying reference of the sender, a code of a given product, a quantity requested and a price per unit.

The semantic part of EDIFACT corresponds to the use of the different components clarifying the function of the EDI message and consequently giving it its whole sense. The main types of components are the following. The **elementary data** identifies an elementary information, for instance the product code, the postal code, the price of a product, the date of a document, and so on. The **composite data** is a combination of simple data and are used to express a logical information. The **segment** groups together simple and/or composite data into logical entities corresponding to a specific function. The EDIFACT **message** is composed of a set of segments for a specific business function, as for instance an order or an invoice. The **function group**, composed by a set of messages of the same type. The **interchange**, or exchange, which forms the whole true communication and dialogue between two partners. An exchange contains one or several messages, one or several function groups. The exchange of a message composed of structured data automatically manageable by the computerised applications of the partners founds an optimised EDI communication between business partners. The following diagram summarises the organisation of an EDIFACT based interchange:



Conceptual modelling

Following up the former Specifications, the Conceptual Modelling effort undertaken here results in two EXPRESS schemas that support the representation of SGML Document Models (dtd_schema) and of EDIFACT Message Models (ediset_model schema) respectively.

(*

The purpose of this data model is to support the representation of documentary models thanks to an object representation of the SGML DTD language. Therefore the semantics of the underlying SGML DTD concepts is captured through EXPRESS entities.

*)

```
SCHEMA dtd_schema ;
```

(*

The entry point of the model is the *dtd* concept which refers to the documentary model as a whole. As such, a *dtd* is identified by a *label* and specifies a *starting-element*, a specific *element* which governs the root of a SGML document tree that conforms to the DTD. Furthermore, a DTD is made of a list (*dtd_items*) of *dtd_item*(s).

*)

```
ENTITY dtd
 label : label ;
 starting_element : element ;
 dtd_items : SET OF dtd_item ;
END_ENTITY ; -- dtd
```

(*

A *dtd_item* may be either a *comment* or a statement.

*)

```
ENTITY dtd_item
 ABSTRACT SUPERTYPE OF (comment, statement);
END_ENTITY ; -- dtd_item
```

…

(*

A *statement* is a generalisation of *entity_statement*, *element_statement* and *attribute_statement*. Each of these *statement* keeps track, through its *dtd* property, of the *dtd* it is associated with.

*)

```
ENTITY statement
 ABSTRACT SUPERTYPE OF (entity_statement, element_statement,
 attribute_statement);
 SUBTYPE OF (dtd_item) ;
INVERSE
 dtd : dtd FOR dtd_items ;
END_ENTITY ; -- dtd_item
```

…

(*

The *element_statement* specifies the structure of the contents of one or several *element*(s) through the *definition*, *inclusion* and *exclusion* attributes. Furthermore b*eginning-tag* and *ending-tag* are Boolean properties specifying whether the referred tags are compulsory or not. Finally an optional *comment* and a set of *attribute*(s) may be attached to a *element*.
*)

```
ENTITY element_statement
 SUBTYPE OF (statement) ;
 beginning_tag : beginning-tag ;
 ending_tag : ending-tag ;
 definition : expression ;
 inclusion : OPTIONAL expression ;
 exclusion : OPTIONAL expression ;
 comment : OPTIONAL comment ;
INVERSE
 elements : SET OF element FOR element_statement ;
END_ENTITY ; -- element_statement
```

(*
The *element* is the key concept of a DTD as it supports the representation of elementary items within the logical structure of the document. A *element* is identified by a *label*. As for *entity*, the *element* inherits from the *expression* concept. Finally a set of *attribute*(s) may be attached to a *element*.
*)

```
ENTITY element
 SUBTYPE OF (expression) ;
 label : label ;
 element_statement : element_statement ;
 attributes : SET OF attribute ;
END_ENTITY ; -- element
```

(*
A last entity is the *expression* one, which may be either a *sgml_entity*, a *element* or a *operator*. Various types of operators potentially participates to the definition of a *expression*. These operators are mainly spread between *unary_operator*(s) and *nary_operator*(s).
*)

```
ENTITY expression
-
END_ENTITY ; -- expression
-
END_SCHEMA ; -- dtd_schema
```

(*
The EXPRESS model hereafter exhibited is a simplified version corresponding to the current definition of the EDIFACT norm. However, a specialisation of the *segment* entity is proposed, so as to characterise some segments the specific role of which has to be considered in a different manner: those are the segments UNA, UNB, UNZ, UNG, UNE, UNH, and UNT, which are mandatory in every interchanges except the UNA segment which is optional.
*)

```
SCHEMA edifact_schema ;

TYPE ITEM = SELECT (SIMPLE_ITEM, COMPOSITE_ITEM);
END_TYPE ;
```

(*
This type allows to include within a segment, in a similar way, simple data and composite data. Note that an entity, abstract supertype of the entities *simple_item* and *composite_item*, could allow to “ factorize ” the common data, in a future version.
The three following entities are the basis for the EDIFACT messages.
*)

```
ENTITY SIMPLE_ITEM;
       val : string;
END_ENTITY ;

ENTITY COMPOSITE_ITEM ;
       items : list [1 : ?] of SIMPLE_ITEM;
END_ENTITY ;
```

(*
As it is the case for the function group and the message, a *COMPOSITE_ITEM* entity is composed of several *SIMPLE_ITEM* entities representing the smallest information unit, and only have one attribute being the value itself.

```
*)
ENTITY SEGMENT
       supertype of (oneof SEGMENT_UNA,
       SEGMENT_UNB,
       SEGMENT_UNG,
       SEGMENT_UNE,
       SEGMENT_UNH,
       SEGMENT_UNZ,
       SEGMENT_UNT);
tag : optional string ;
components : list [1 : ?] of ITEM;
END_ENTITY ;
(*
```

An *SEGMENT* entity is composed of a segment name, called TAG, then of a set of values, called ITEM which are either *COMPOSITE_ITEM* or *SIMPLE_ITEM* elements.

```
*)
TYPE INTERCHANGE_COMPONENT =     SELECT (MESSAGE ,FUNCTIONAL_GROUP) ;
END_TYPE ;
(*
```

As it is the case with the *ITEM* type, this type allows to deal in a similar way with messages and groups of messages within an EDI interchange.

```
*)
ENTITY MESSAGE ;
UNH: SEGMENT_UNH;
       segments : list [1 : ?] of SEGMENT ;
UNT: SEGMENT_UNT;
END_ENTITY ;

ENTITY FUNCTIONAL_GROUP;
UNG: SEGMENT_UNG;
       messages : list[1 : ?] of MESSAGES;
UNE: SEGMENT_UNE;
END_ENTITY ;
(*
```

The *FUNCTIONAL_GROUP* entity groups a set of messages of the same type. The *MESSAGE* entity is composed of the two segments UNH and UNT, with the list of “ generic ” segments (*SEGMENT*).

```
*)
ENTITY INTERCHANGE ;
UNA: SEGMENT_UNA;
UNB: SEGMENT_UNB;
       components : list [1 : ?] of INTERCHANGE_COMPONENT;
UNZ: SEGMENT_UNZ;
END_ENTITY ;
(*
```

The entity *INTERCHANGE* is the one describing the concept of interchange itself, thus it is the root of the entities inheritance graph. This entity contains two or three segments: the UNA optional segment, the UNB and UNZ mandatory segments, and an ordered list of objects which are either *MESSAGE* or *FUNCTIONAL_GROUP* entities.

The rest of the schema is composed of the set of specialised segments which can be found in most of the EDIFACT messages.

```
*)
ENTITY SEGMENT_UNA
-
END_ENTITY ;
-
END_SCHEMA ; -- edifact_schema
```

5. Conclusion and future works

Building upon the on-going specification and implementation of the COAST, one of the objectives of the VEGA project is to address the integration and distribution of LSE information systems within the DIS. The DIS will take advantage of developments within VEGA to enlarge the support and distribution of information from product data to any kind of documentation or EDIFACT based messages, ensuring a remote access to this kind of information for all the actors involved in LSE projects through the connection of the DIS to the COAST.

Indeed, the VEGA DIS is supposed to mainly focus on the *management* and *distribution* of these different types of information (product data, documents, messages), provided that they be regarded as product model data defined by an explicit meta-model or schema satisfying the EXPRESS concepts and semantics. This means the remote access and exchange of previously elaborated and coherent SGML documents and EDI messages in the particular context of distributed systems environments. One must keep in mind that it is clearly out of scope of the DIS to deal with all issues related to *integration* of product data information and documentary or EDI message information, i.e. the automatic production of SGML documents (resp. EDIFACT messages) from the product data information. On the other side, the DIS maybe the spinal chord of more global SGML (resp. EDI) based frameworks, including the creation, the integration with product data information for a specific purpose, and the global management (delivery, storage, access, distribution, protection) of SGML documents (resp. EDI messages). To reach its objective, the DIS will rely on the distribution of any STEP models through specific CORBA mechanisms (COAST, furthermore providing security layers and communication media), and on standards for information as:

- SGML for any kind of structured documents ;
- EDIFACT for all EDI messages issues ;
- And also HTML/VRML for more dedicated WEB documents, granting easy navigation & access across world-wide spread company's information systems based on WWW hypermedia technologies. VRML (Virtual Reality Modelling Language) is a file format supposed to become a new WEB standard for describing interactive 3D objects and worlds to be then viewed by WEB users, in a similar way HTML is used to view text under WEB browser.

The DIS will be a major attempt showing the benefits of bringing together technologies like STEP, CORBA and the WEB to offer a distributed electronic data management service. Inside the universe of the future virtual enterprise, it will offer a powerful means to the various actors to keep their different views on the project and to exchange in the meantime in a transparent and seamless manner the related information thanks to a same generic and neutral administration of the various representations and formats of this information. Linking in such a way technical product information, documentation and electronic messages will improve information dissemination, configuration control and collaborative work for broad benefits in LSE real business cases.

6 References

[SGML] The SGML Web Page - http://www.sil.org/sgml/sgml.html

[EDI] The Electronic Data Interchange - http://www.ensmp.fr/~scherer/weeb/edi.html

[Fowl95] J. Fowler. *STEP for Data Management, Exchange and Sharing*. Technology Appraisals 1995.

[Koet97] M. Koethe, K . Schulz., V. Amar, A. Zarli. *COAST Architecture: The CORBA Access to STEP Information Storage Architecture and Specification*, Deliverable D301 in the ESPRIT-Project 20408 ìVEGAî - 1997.

[Amar97] V. Amar, M. Koethe, K. Schulz, A. Zarli. An open Step-based distributed infrstructure: the COAST platform to appear in the proceedings of the Concurrent Engineering in Construction (CEC) conference, July 3-4, 1997 London.

[VEGA95] VEGA - ESPRIT-Project 20408: “Virtual Enterprises using Groupware and distributed Architectures ”, Technical Annex - September 95.

[OMG95a] The Common Object Request Broker Architecture and Specification (CORBA) Revision 2.0. July 95. *URL http://www.omg.org*.

[MoZa96] T. J. Mowbray & R. Zahavi. *The Essential CORBA - System Integration Using Distributed Objects*. John Wiley and Sons.

[OHE96] R. Orfali, D. Harkey and J. Edward. *The Essential Distributed Objects Survival Guide*. John Wiley and Sons Editor.

[Sieg96] J. Siegel & Al. "CORBA Fundamentals and Programming," John Wiley and Sons, April 96.

[ISO94a] .Industrial automation systems and integration - Product data representation and exchange Part 21. *Implementation methods: Clear text encoding of the exchange structure*. 1994.

[ISO95] Industrial automation systems and integration - Product data representation and exchange Part 22. *Standard Data Access Interface*. 1995.

[ISO94b] Industrial automation systems and integration - Product data representation and exchange Part 11. *Description methods: the EXPRESS language reference manual*. 1994

[OMG95b] Object Management Group (OMG). *CORBA Services: Common Objects Services Specification*, Revised Edition, 95-3-31 ed, Marsh 1995.