An X.500-based Product Catalogue

Electronic Data Interchange is the means by which companies exchange trade-related documents between application processes in electronically processable form. The initial development of this technology has been hampered by the lack of accepted international standards. One of these standards is X.500 for directory services. X.500 can be used as a powerful tool in allowing electronic data interchange to take place between companies without the necessity for prior agreement. The article describes a mechanism by which the directory can be used to store product information that may be used by a purchaser in ordering electronically.

The Networks & Telecommunications research group at Trinity College consists of a small number of full and part-time postgraduate students working on projects related to a broad spectrum of network services. One of the special areas of interest involves harnessing the power of open standards, such as OSI and EDIFACT, to solve the problems inherent in global open Electronic Data Interchange (EDI) [2].

** by Dr. Donal O'Mahony*
Trinity College Dublin

Technical Infrastructure

Many of the projects make use of OSI application level protocols and, to this end, the group has developed expertise in using the ISO Development Environment (ISODE) and associated software. The group operates an X.400 MTA based on the PP software within the COSINE MHS project and holds the root of the global X.500 directory tree for Ireland using QUIPU DSA software. We are also actively piloting an OSI security toolkit called OSISEC as part of the EC PASSWORD project. These provide the software building blocks for experimental work in new and enhanced EDI services.

Global Directory

Early work within the group in interfacing an existing application package to an EDI back-end highlighted a problem that occurs wherever documents related to trade in goods are processed electronically. Before the transaction can take place, each party must load information relating to the traded item into stock files that are held locally. The fact that different attributes may be used at each end means that EDI back-ends must engage in a translation process from one set of codes to another as well as undertaking frequent manual updates to ensure that the two instances of information are kept up to date. While these problems are surmountable when companies engage in a significant number of transactions with each other, they contribute to the well-known "first-order" problem in EDI, i.e. the amount of work that must be engaged in between two trading parties before the first purchase order can be dispatched electronically. One solution that was explored by Neil Weldon was to make use of the X.500 global directory to provide a shared context in which two parties who had not traded before could overcome the "first order" problem.

X.500 Standard

The CCITT X.500 recommendations describe how to build a global distributed database which can be used to hold information that is useful in establishing communications between people and application processes. This database is tree-structured with a sub-tree for all countries in the world, each of which can contain information relating to organisations, people, application processes, databases, etc. within that country. Since the database will become very large (there are currently in excess of 1 million entries), it can be distributed over a worldwide network of Directory System Agents (DSAs) each of which holds a small portion of the total information.Information is held in *objects* which have a number of mandatory and optional attributes. To facilitate the storage of diverse forms of information, a hierarchy of object *classes* has been defined and *inheritance* can be used to define new classes that contain all attributes of their parent object classes. Some examples of object classes include: *person, organization,* and *country.*The directory is interrogated by employing a Directory User Agent (DUA), which contacts its nearest DSA and interrogates it using a standardised access protocol (DAP). If the DSA is not holding the information locally, it can obtain it from a neighbouring DSA that is holding either the master copy of the information or a slave/cache copy. A key feature of the system is that DUAs operating in different parts of the world will see the same directory contents and the fact that the database is distributed is hidden from the user.

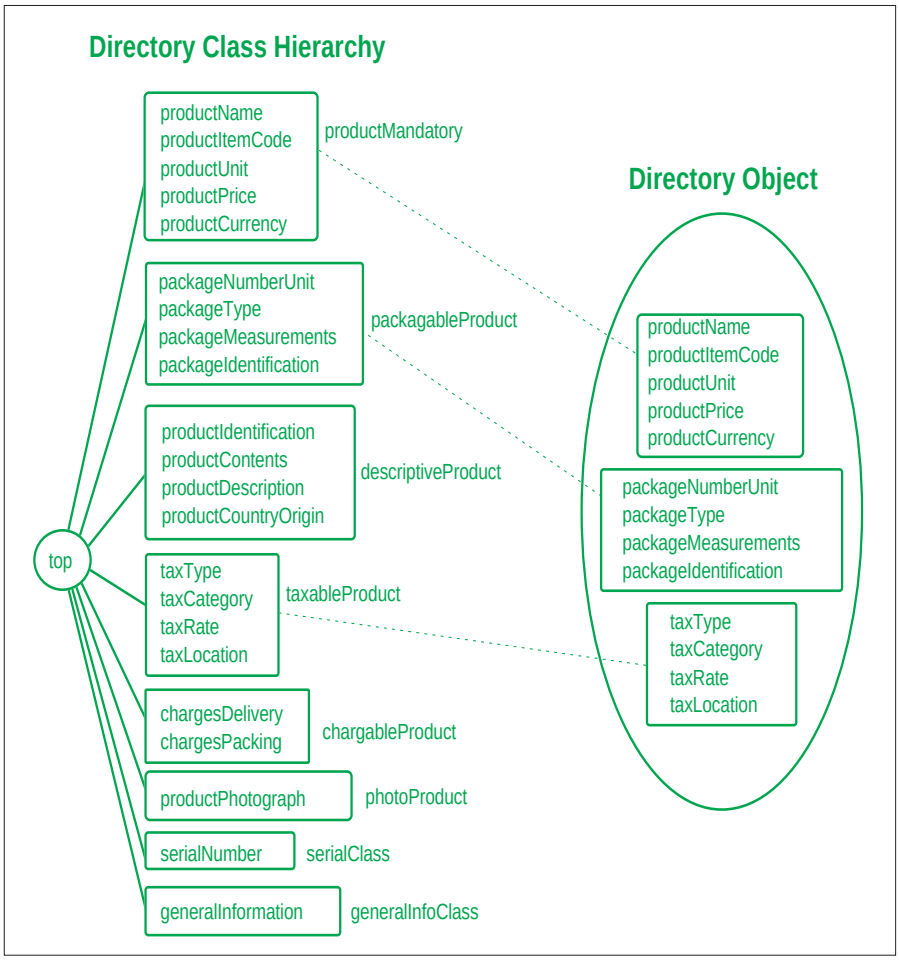

Figure 1: Inheriting from many EDI object classes

X.435 Standard

The X.435 standard describing how EDI messages can be sent over an X.400

mail network defines a number of new object classes for the directory, including *EDIUser* which allows an organisation to locate a prospective trading partner within the directory and procure information on how to contact that partner about what types of EDI documents could be received from him in addition to other information.

X.500-based Product Catalogue

At Trinity College, we have extended this notion to allow a company's entire range of products to be included in the directory. This was done by designing a range of new object classes to hold product-related information. The most important class is called *productMandatory* and any object that is an instance of this class will contain basic information such as price and itemcode. Depending on requirements, other classes can be "mixed-in" to provide further information. The diagram shows how the multiple inheritance features of the directory can be used to produce an object containing tax and packaging information in addition to the normal *productMandatory* attributes. Special-purpose DUAs can be designed either to focus on the basic information or, where appropriate hardware exists, use more exotic attributes such as the product photograph or an audio/video clip describing the product features. An additional benefit to holding the product in the directory is that it acquires a globally unique *Distinguished Name* which can help in eliminating the need for many different codes and associated translation mechanisms.

By using the distributed database facilities of X.500, a company can keep control of the *MASTER* copy of the product catalogue, and any changes made to this will (within a short time period) be reflected in the responses to queries made by DUAs all over the world. By using the SLAVE copy and CACHING facilities of the directory, the data can be moved as close to the customer base as is required to assure a reasonable response time to queries. Indeed, one can envisage a case where changes to the database could be propagated by the physical distribution of CD-ROM to the point of sale.

Prototype

To test the feasibility of this idea, we built a prototype EDI X.500 User Agent. This allowed a user to browse through the global directory to find a supplier company. In a subtree beneath the companies entry, objects are stored, each containing details on an individual product. The user can browse through the catalogue, view photos of the product, check prices, and, finally, click on a product to be purchased.

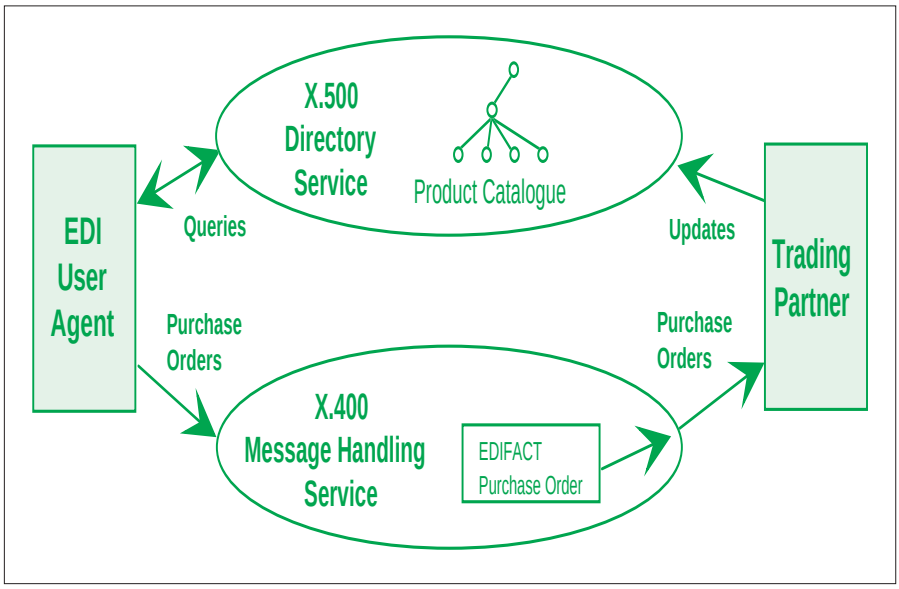

Figure 2: Using the X.500 EDI User Agent

The DUA then extracts sufficient information from the product attributes to fill all of the fields of the *LIN* segment in an EDIFACT purchase order. When the purchase order is complete, pressing a *send* button causes the DUA to extract an electronic mail address from the organisation entry and dispatch the document electronically.

The prototype EDI UA used normal UNIX mail as the document transfer mechanism, but a follow-up project has developed a simple X.400 user agent which can be adapted to full X.435 capability in the future. The issues involved in EDI user agent design are being pursued by Orla Greevy, and some early results of her research are given in [1].

Security and Payment Systems

Another area which the group is pursuing is that of security and electronic payment systems. It is hoped that the prototype EDI UA can be extended to allow standard security services such as message authentication, non-repudiation and confidentiality. Hitesh Tewari is working on a flexible electronic payment system based on the IBM *control vector* concept that will allow many different forms of payment (cash, cheque, draft) to be effected in a secure fashion. These payments could be included in EDI documents taking the Open EDI concept a stage further towards eliminating barriers to electronic trading.

Auditing and Control Workbench

In the short to medium-term future, the majority of EDI users will move to electronic trading by grafting an EDI back-end to existing applications software. One of the disadvantages of this scheme is that normal auditing and control procedures, which are paper-based, are often diminished in effectiveness. In October this year, the group will embark on a project in association with a local EDI software company to develop an EDI Auditing & Control Workbench. The goal of this research is to develop an intermediate system which can intercept EDI traffic flowing into and out of an organisation to allow auditing and control functions to be carried out effectively. Among the functions that this workbench will provide are: document visualisation, logging, reconciliation and authorisation as well as using knowledge-based systems to ensure that electronic transactions are being carried out according to company policy.

Conclusion

If truly open EDI is to be achieved, technological solutions must be provided that enable parties to trade electronically without the need for prior contact. These transactions must take place in a secure, auditable and controlled fashion. Future projects within the group will be driven by this goal.

References ■

[1] *Greevy, O.:* Current Trends in EDI and X.400, in: Bulletin Technique, Swiss PTT Telecom, No. 11/1992, pp 472-481.

[2] *O'Mahony, D.:* The Use of X.400 and X.500 in Electronic Data Interchange, in: Proceedings of the 17th International Conference on Information-Technologies and Programming, Sofia, July 1992, pp 36-44.

** Dr. Donal O'Mahony is a lecturer in computer science and co-ordinator of the networks and telecommunications research group at the Computer Science Department of Trinity College in Dublin.*