Managing Different Interfaces in Electronic Commerce

Christian Huemer
Institute of Applied Computer Science and Information Systems,
University of Vienna, Austria, e-mail: ch@ifs.univie.ac.at

Abstract

During the last years the field of electronic commerce has seen a rapid growth. Various methods have been developed to allow the electronic transfer of business-related information. Those methods support either human to human, human to machine, machine to human or machine to machine interaction. Unfortunately, the methods used in those four categories are completely different and there does not exist a interconnection between them. Therefore, we describe how to manage the different interfaces used in electronic commerce for both the sending and receiving end of the business transaction.

1. Introduction

Efficient communication processes are one of the most important factors of success for an organization. This subject is focused by the methods of electronic commerce. In this paper we present a concept for an organization which is able to offer all kinds of electronic commerce to its business partners and wants to integrate its own information system into the business processes in the most efficient way. Suppose an organization which wants to collect orders for its product by using the methods offered by electronic commerce. It has few main business partners with a high transaction volume who themselves use an information system to place their orders and will thus prefer EDI to conduct the organization. On the other hand a lot of sales are made with smaller business partners and end consumers who do not use an information system. But those might have Internet access and consequently will find it convenient to order by using WWW forms or sending e-mails.

Accordingly, the organization must support inbound and outbound EDI transactions, receiving structured information from an electronic form (WWW form) resulting in a response to a browser and receiving unstructured information in an e-mail resulting in an computer generated response e-mail. Providing the opportunity of processing a business transaction via the above mentioned methods, will require different interfaces to the organization‘s information system. Unfortunately, the technologies currently used to support these methods are not completely compatible to each other. Nevertheless, an integration of these methods is an absolute necessity. Therefore, we present a framework which describes where to unify the information flows of the different methods.

2. Technological definition of E-Commerce

From a technological viewpoint electronic commerce can be defined as ‘two or more organizations interoperating to process a business transaction. The information describing the detail of that transaction is communicated across the interface between the parties by electronic means’ [1]. According to the ways in which data can be passed across the interface between the parties, we distinguish between four main categories of electronic commerce:

Human to human interaction: A human creates the information at the sending end and another human interprets and processes it at the receiving end (e.g. electronic mail). There is no interconnection to the information systems on both sides. This means that the information must be taken from and entered into the information systems by additional processes.

Human to machine interaction: At the sending end a human interacts with an application to compose the information in a structured, machine-interpretable form. These information is transferred to the receiving end which processes the transaction with no further human intervention (e.g. electronic forms). The sender still has to take and enter the information from/into his information system by separate processes, whereas the receiver is released from that burden.

Machine to human interaction: The information is automatically generated from the data in the sender’s information system, but a human has to interpret and process it (e.g. computer generated e-mail). The receiver has to enter the information into his information system by an additional process.

Machine to machine interaction: The information is

automatically generated from the data in the sender's information system. At the receiving end, the information is automatically processed and entered into the receiver's information system. This technique is known as electronic data interchange (EDI). It guarantees for a full interconnection to the information systems at both sides of the business transaction.

2.1. EDI

EDI is defined as the electronic transfer from computer to computer of commercial or administrative transactions using an agreed standard to structure the transaction or message data [2,3]. In EDI relationships data should be generated automatically at the sending end, transferred in a standard format to the receiving end where it should be processed automatically. EDI is independent of the transport medium, hardware and software used by the involved business partners. EDI ensures optimal benefits in an environment where all business partners use information systems.

It is obvious that simply transmitting data values of data elements included in the business transaction would not make any sense without some kind of identification. Therefore, a data format which defines how data are to be represented must be defined to ensure that no human intervention for its interpretation is required. To avoid a proliferation of bilateral exchange formats, a common interchange format between all the business partners is a critical factor in the evolution and acceptance of EDI. Then each of the trading partners sharing the common interchange format only must be able to translate from and into the agreed format to communicate with the others. Consequently, a standard describes the structured data format which presents the data content of a document.

In the course of time various different exchange formats or so-called „standards" have been developed. Those standards range from proprietary to branch-restricted and/or national solutions. Knowing the limitation of standards not shared by the whole businesses in the world, the UN/ECE developed in accordance with ISO EDIFACT as international and branch-independent standard. Therefore, we will explain the EDI concept on basis of EDIFACT. EDIFACT - like all other EDI standards - covers the following topics [4]:

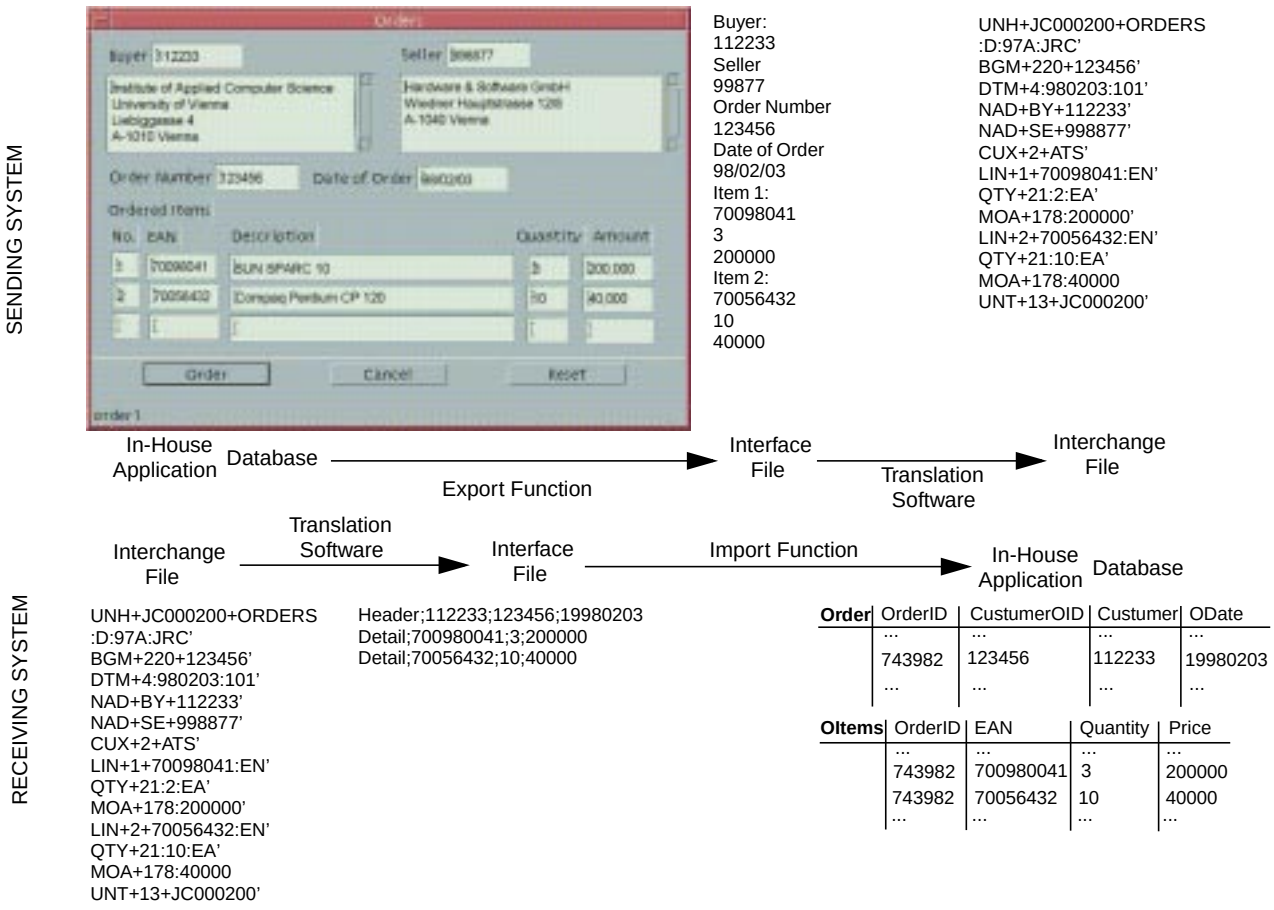

Figure 1. EDI-based information exchange

- A grammar or set of syntax rules which determine the acceptable characters of data and the order in which they may be used
- A vocabulary of accepted words, business definitions and data elements
- A letter, file or message design presenting the information in an ordered sequence

To illustrate the information flows necessary to transfer relevant data from the sender‘s application to the recipient‘s application using EDIFACT we use a simplified example depicted in Figure 1: The Institute of Applied Computer Science orders computers from a local wholesaler by using its order program. The relevant data is inserted into the appropriate order dialog. By activating the ‘Order’-Button the order is stored in the institute’s application database. Furthermore, the relevant data are extracted and placed on an interface file. This interface file is in a proprietary, usually record based format. The interface file serves as input to the translation software. The function of the translation software is to convert the data in-house format into an EDIFACT message. For this purpose, the translation software’s programs simultaneously run through a mapping table, based on the appropriate message structure (order) and the file created at the interface file. Basic validation checks are made, codes converted partner/profiles ascertained and data formatted according to the EDI syntax. The formatted data are written to the interchange file which is transferred to the recipient via the communication interface.

At the receiving system the order of processing will be the following. Receiving of data can be direct of by calling down from a mailbox. The received interchange file in EDIFACT syntax is processed by the translation software. The translations software performs the inverse process to that one of the sender. Consequently, the data is deformatted according to the specified mapping tables. Validation is carried out according to the known parameters of the message and codes are converted to those used in-house. The required data are placed on an interface file for further use. Note that the proprietary file format of the interface file is usually different to that of the sender. Finally, the interface file is imported into the application’s database by probably performing further validation checks.

2.2. WWW commerce

The world wide web and its specification language HTML have seen a dramatic growth during the last years [5]. The HTML specification also includes electronic forms. Consequently, an electronic form is just a normal HTML file which works as a front-end for manual data entry. The back-end is usually a script (C, Perl, Java) which automatically processes the entered data and provides it to the in-house information system in an appropriate format. Since WWW electronic forms are fully under control of the requester and are always current, they function as an interface to the requester’s information system.

Many companies are putting software applications on the world wide web (WWW). These applications are usually written by using Java Applets or by using the Common Gateway Interface (CGI). Since we have chosen the CGI approach to illustrate integration of EDI and WWW, we will further concentrate on CGI. Nevertheless, it is important to note that the principal concept could be easily migrated to the techniques used in Java.

CGI defines the communication link between a WWW server and a WWW application. A CGI script is a program

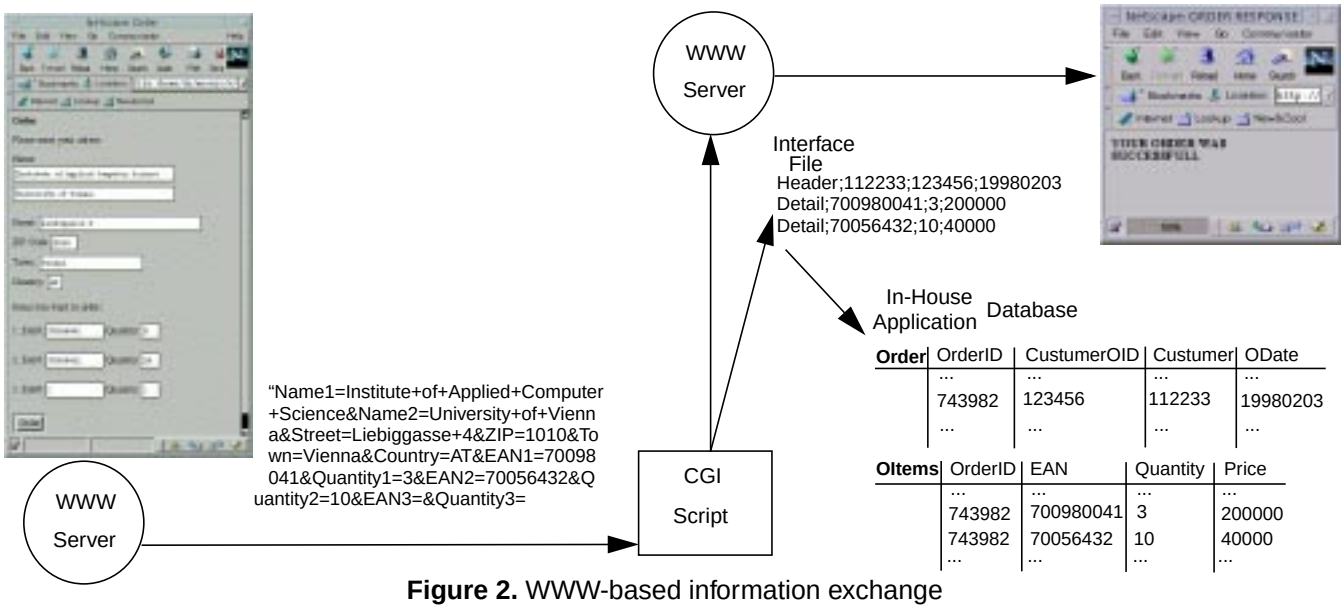

Figure 2. WWW-based information exchange

run by the WWW server in response to a request from a WWW browser. Usually this request results from a user submitting a HTML form. When a request for a CGI script is received by the WWW server, it runs the script as a child process [6].

The form's data is sent to the script when the user presses the 'Submit' button. Forms can use different methods to send their data to a script: GET and POST. More powerful is the POST method, where the data is sent to a CGI script through standard input (stdin). The raw data from a request is of the form *key1=value1&key2=value2*. The keys are the names of form elements, and the values are what the user entered.

The reactive script takes the input from the user and acts on the provided data. In business oriented systems the script provides the interface to the requester's information system. Therefore, the script either manipulates the data to insert it directly to the information system's database or it produces an interface file, which is in a suitable format for import into the database. After having performed its main function the script sends the server a reply that the server forwards to the client.

To illustrate the CGI POST approach we take the same example as in the previous section. In this case the wholesaler provides a WWW form to its customers. This form and the underlying HTML code are presented in Figure 2. The responsible agent at the institute completes the WWW form provided by the wholesaler. By pressing the 'Order' button the form's data is sent to the wholesaler's WWW server. The server starts the order script and passes the order data as depicted in Figure 2 in standard input to the script. The script takes the order data and calculates values required by the information system. Furthermore, it undertakes necessary data manipulations and rearrangements to produce the appropriate interface file, which is suitable for database import to the order-entry program of the wholesaler. Additionally, the script produces an order response statement which is passed through the sever to the client.

2.3. Generator-based information exchange

In this subsection we concentrate on information exchange where the receiving partner is not able to provide an machine interface to its information system. Therefore, the sender should be able to transfer the information in a style which could be easily interpreted by a human. In traditional systems this style corresponds to a letter. If the sending organization is not willing to have this letter typed by a secretary, it must provide an interface to its information system. This means that the organization provides a textual framework for each kind of business transaction, which consists partially of fixed text and variables which are filled by the actual data of the information system. The insertion of the relevant data from the application is performed by a generator function. This generator function can either be directly implemented into the application (database generator) or it is a script working on exported data placed in an interface file. The complete text created by the generator can either be sent to the printer, sent as a fax, or - following the idea of e-commerce - be placed into an e-mail to be sent to the recipient. In Figure 3 we present the generator based approach for our order example, using the intermediate interface file.

3. Integration of different interfaces

Having analyzed the methods used in electronic commerce, we presents our concept to manage all the different interfaces at both ends of the business transaction, which is depicted in Figure 4.

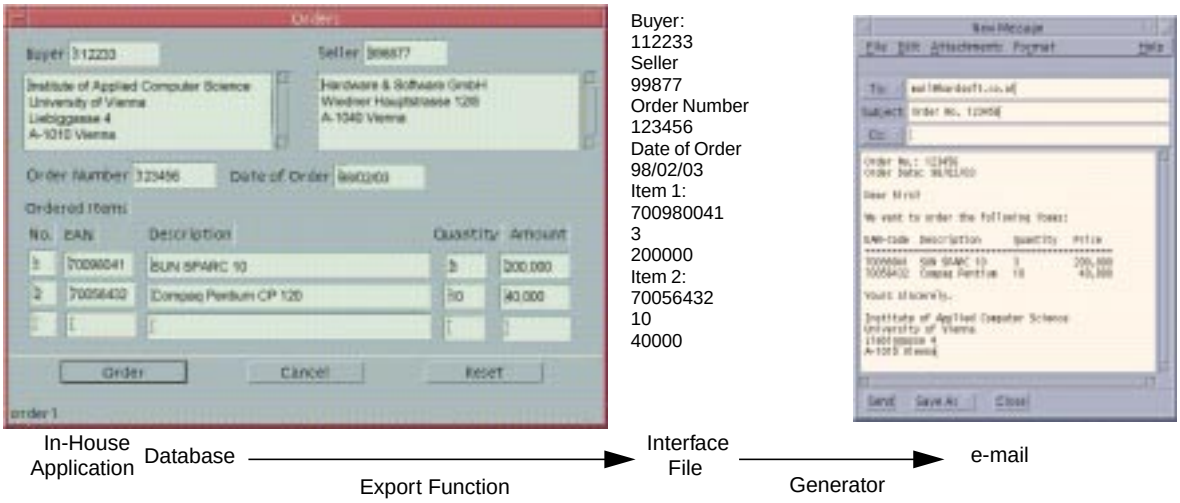

Figure 3. Generator-based information exchange

First, we concentrate on the sending end of the transaction. If there is no interconnection to the information systems on both ends, the sender must re-key the information stored in the application into an e-mail by a secretary. In case the recipient offers an WWW form for the underlying transaction, the sender has to re-key the information into the corresponding form. This might be convenient for the sender if there is no connection to its application. But in the opposite case the WWW approach is in conflict with its intention to automatically produce the transfer information. Unfortunately, there does not exist a method of automatically posting in-house data into a WWW form. Therefore, the business partners have to resolve the conflicting situation by taking an EDI approach into consideration or the more powerful organization will insist on its preferred method.

If the sender is able to take advantage of automatically generating the transfer data, it has two alternatives depending on whether the receiver supports a machine interface or not. In both cases the transfer data will be exported from the application‘s database into the same kind of interface file. Then a script can act on this interface file. This script searches for the receiver‘s identification in the interface file and determines according to a specification table whether the receiver supports EDI or not. According to this decision it will either start the translation software to produce an EDI message or automatically generate an e-mail.

On the receiving end the integration of electronic commerce into the organization‘s information system will be the following. If the sender provides the transaction data only as e-mail - no matter whether automatically produced or not - a human must re-key the included information into the recipients information system. To avoid this overhead, the receiver should force its business partners to use its machine interfaces represented by WWW forms and EDI.

A requester willing to provide both interfaces to its business partners should redirect the output of a CGI script into one of the intermediate products of the EDI approach. For this purpose two alternatives seem to be practicable. The first opportunity is to enter into the EDI approach before the translation software. This means that the CGI script has to transform the form’s data into an EDI message which is stored in a separate file. Then the translation software can process the file like every incoming EDI message. This kind of integration has two disadvantages: Firstly, script languages usually offer methods (of included libraries) to receive the form data in a very convenient way. As a consequence it is a great overhead to translate this data into a rather complex EDI format - which is only considered to be an intermediate format. Secondly, it is very likely that the response statement is created when the script has produced the EDI file. But this does not mean that data has passed through to the application. Otherwise the script has to be synchronized with the translation software.

Therefore, we prefer the WWW approach to enter the step-by-step EDI approach after the mapping in the translation software. Consequently, the CGI script should generate the same interface file as the translation software. In this integration approach, the full power of CGI scripts is exhausted because there is no need to generate an

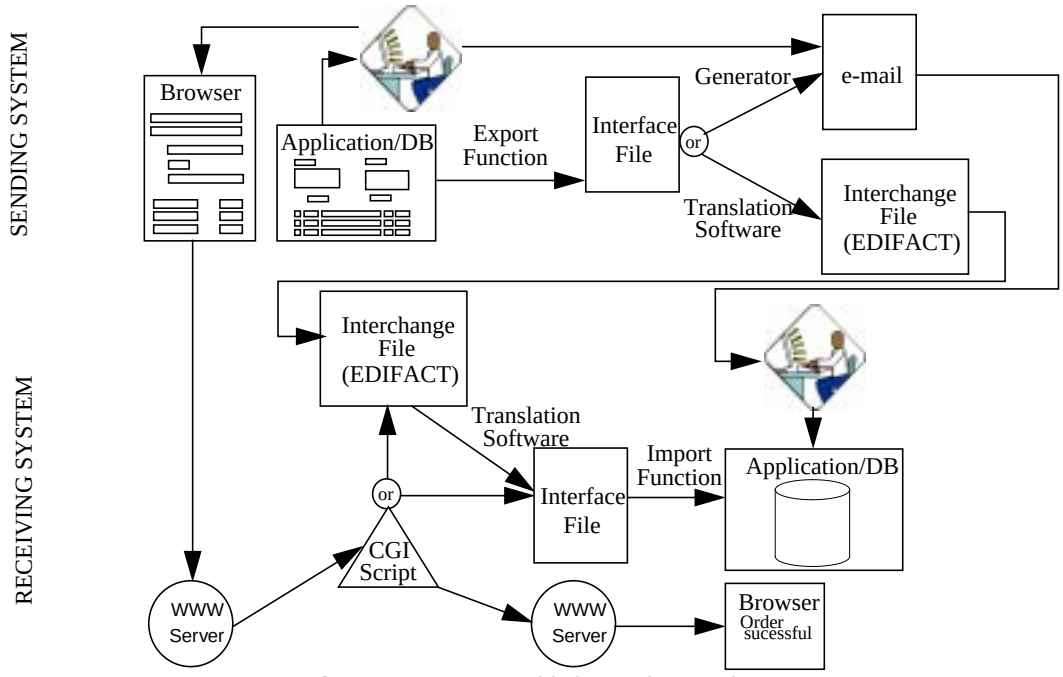

Figure 4. Integrated information exchange

intermediate EDI file. This also means that one step of the EDI approach can be omitted. Furthermore, the response statement can be immediately sent after the interface file has been generated.

References

[1] Steel K.; Matching Functionality of Interoperating Applications: Another Approach to EDI Standardisation; ISO/IEC JTC1/SC30 IT11/7/94-108; September 1994

[2] Emmelhainz M.A.; Electronic Data Interchange: A Total Management Guide; Van Nostrand Reinhold; New York; 1990

[3] Hill N.C., Ferguson D.M.; Electronic Data Interchange: A Definition and Perspective; EDI Forum: The Journal of Electric Data Interchange, Vol. 1, Issue 1; 1989; pp 5 - 12

[4] Berge J.; The EDIFACT Standards; NCC Blackwell Limited; Oxford; 1991

[5] Berners-Lee T., Cailliau R., Luotonen A., Nielsen H.F., Secret A., The world-wide web, Communications of the ACM, Vol 37, Issue 8; pp 76 -82

[6] Asbury S., Mathews J, Sol S.; CGI How-to; Waite Group Press, 1996