

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Table of Contents

Applies To:	4
Summary	4
EDI: Not Rocket Science	5
Heavy Lifting	5
White knuckle fliers	6
Taking care of business: Galactic Games.....	6
A dizzying height	6
Closing the loop	8
Getting the goods	8
A minor digression about standards	8
A web of activities.....	9
Making it.....	10
<i>La piece de resistance</i>	11
Let's get technical	11
An agreeable relationship.....	11
The match maker.....	13
SAP's traditional approach to EDI.....	13
Outward bound.....	16
Moving on out	16
Bringing it home	19
EDI and XI.....	20
The Source ... Templates and Maps.....	21
Browsing the libraries	21
What's the big deal about qualifiers?	22
Addressing the envelope.....	22
Outbound Enveloping.....	23
Inbound Deenveloping.....	24
Communications and Security.....	25

Some thoughts about EDI In an XI Environment

Wrapping Up	26
Author Bio	28

Figures

Figure 1: A bird's eye view of Galactic Games.	7
Figure 2: Transactional overview of finished goods purchasing processing.	9
Figure 3: EDI requirements and setup overview.	12
Figure 4: A bird's eye view of the EDI subsystem process flow.....	14
Figure 5: View of an X12 944 EDI transaction.	18
Figure 6: Enveloping: .Schematic overview of on-going processes of extraction and collection.....	24
Figure 7: Denveloping: .Splitting out the unique transactions.....	25

Some thoughts about EDI In an XI Environment

Applies To:

SAP Exchange Infrastructure v 3.0, SP12, Integration Repository and Directory

Summary

In spite of predictions of its death in recent years, EDI remains the 800 pound gorilla of e-Commerce and Business Process Integration between Trading Partners. While not designed as an EDI subsystem, XI can support EDI with some strategic customization and a little help from some third party tools.

This article explores some of the business and technical challenges and requirements of EDI and presents ideas about how XI can be tweaked to form the backbone of a technical architecture to enable EDI in your organization.

By: Emmanuel Hadzipetros

Company: NBC Universal

Date: November 14, 2005

Some thoughts about EDI In an XI Environment

EDI: Not Rocket Science

How do you use XI to handle EDI transactions is one of the more common EDI-related questions posed in the SDN XI forum. Indeed, I've asked a variant of it myself.

These questions generally raise a flurry of opinions that more often than not lay bare other struggles to understand the complexities of EDI, or Electronic Data Interchange, and XI's role in addressing them.

The problem, however, is that often these questions are being addressed from a purely technical perspective, as if EDI could be reduced to technical issues of interface design, development and configuration.

This is putting the cart before the horse. From a technical point of view, EDI is not rocket science. But EDI is more about business and process, reinforced by relationships, than it is about technology.

In fact, EDI is a prime example of process – the whole web of activities and relationships behind the running of any business – enabled by technology; it's about more effectively doing business with your trading partners, reducing paperwork and manual processes, and generally improving the bottom line all around.

EDI is expressed at its most basic level through the electronic exchange, over a proprietary network or across the Internet, of standard business documents such as Purchase Orders, Delivery Documents and Invoices in a universal format implemented through agreements to handle the particular requirements of your business relationship with your trading partners.

Heavy Lifting

EDI is the oldest form of e-Commerce around, dating back to the 1960s in the United States. Its imminent demise has been widely predicted by the professional techno-rati who tout the brave new world of on-demand information, web services and business process integration across corporate boundaries empowered by XML, the internet and all the other open standards and technologies rocking the industry today.

It is true: these are wonderful times and revolutionary changes in the way that data and business processes are shared are slowly being realized. But, to paraphrase Mark Twain's famous quip about his own demise: reports of EDI's death are greatly exaggerated.

What is in fact happening is that EDI vendors and consumers are discovering that XML, the internet and all the other emerging standards and technologies are highly compatible with EDI. They actually simplify the development of EDI standards across a broad range of industries, reducing its total cost of ownership and making it more accessible to a wider array of trading partners without comprising security.

Just one small example: the rapidly growing move towards the transmission of business documents across the internet via AS2, fueled to a large degree by a mandate from Wal-Mart to its suppliers, eliminates the expensive and on-going fees for the use of proprietary networks managed by VANs (Value Added Networks).

And thank God for that: EDI does the heavy lifting of data and business process exchange for large (and, increasingly, smaller) organizations that need to buy and sell goods and services, including, to name just a few, the United States, Canada and other governments both national and regional; the United Nations; Boeing; GM; GE; Paramount Pictures; and Wal-Mart, with virtually all of their suppliers, constituting what is arguably the world's largest EDI ecosystem, with the possible exception of the US government.

It's easier to list industries than companies. Developed originally for the automotive and transportation industries, EDI is widely used by government, utility, health care, banking and finance, manufacturing, retail, distribution, and just about any other industry you might care to name.

Some thoughts about EDI In an XI Environment

It would be no exaggeration to state that globally, the annual dollar value of all documents and transactions carried by EDI is well in excess of \$1 Trillion.

EDI is still the 800 pound Gorilla of e-Commerce.

White knuckle fliers

EDI is usually a mission-critical 24x7 operation involving the exchange of massive quantities of data, mostly in batch processing mode, reflecting its heritage in a highly customized mainframe environment. As old as many of these system may be, they are extremely reliable. They have to be: downtime, or a glitch in the scheduled posting of say customer PO's, or a data error in a batch of thousands of invoices, can result in huge financial losses to one or both partners. There's very little room for error in EDI.

Management tends to be very conservative when it comes to their EDI system. They are always nervous whenever what works is tinkered with, or replaced by a new system. That's why implementing EDI can be a white knuckle experience: even if management knows that the legacy system needs to be replaced, the anxiety is palpable: the new system must work better than the old, without taking away anything that the old provided, especially reliability. And there are always lots of assiduously cultivated careers riding on it.

Taking care of business: Galactic Games

Consider the case of computer game publisher Galactic Games and their hot new title Intergalactic Conquest. How are they going to get it onto the shelves of their biggest customer Wal-Mart in a timely manner? How are they going to make money out of it while enabling their suppliers and Wal-Mart to turn their own profit?

This is the fundamental question in understanding the role of EDI. It's all about effectively running the business: buying and selling the goods and services required to develop, manufacture, sell, distribute and collect revenues on the new game; and controlling the costs so that each partner in the supply chain benefits. This determines the landscape and architecture of EDI.

Before we get to the technology, we need to understand at least something of the business model.

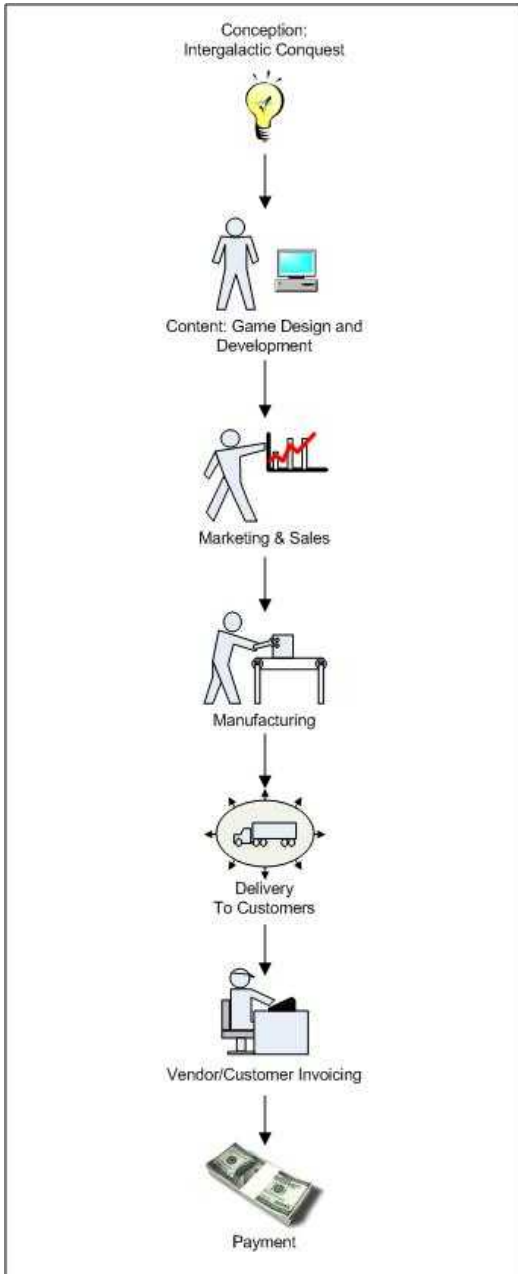
A dizzying height

The end-to-end business model is illustrated from the 100,000 foot level in Figure 1. The conception, the idea for the new game, comes first. But the game won't develop itself. Designers are contracted in LA who in turn sub-contract the coding to a service bureau in India.

It's not enough to develop a new game, however. Customer demand must be created and sales in hand before Galactic will invest the substantial sums required for manufacturing and distribution.

Some thoughts about EDI In an XI Environment

Figure 1: A bird's eye view of Galactic Games.



Some thoughts about EDI In an XI Environment

Marketing gets to work developing campaigns to raise awareness of the new game and the sales folk pick up the phone and get on planes to work their customer contacts. The customer is sold and Wal-Mart signs a contract for 100,000 units of Galactic Conquest.

Now manufacturing kicks in. The title is sold on CD-ROMs that are replicated and packaged by long-time vendor Disc Services International (DSI), who also manages the buying and selling of raw materials needed to manufacture the finished product, and takes care of distributing it to the customer. DSI also keeps track of inventory, both finished goods and raw materials, for all of Galactic's titles.

Closing the loop

Manufacturing is completed and DSI trucks the finished games to one of Wal-Mart's Distribution Centers. Now Wal-Mart's fabled distribution system kicks in and the hot new titles are farmed out to all 3,900 plus stores across the USA.

At this point, DSI invoices Galactic for their services and Galactic in turn invoices Wal-Mart. The moment of truth has finally arrived: Accounts Payable in both Galactic Games and Wal-Mart pay on their invoices in a fair and timely manner.

If everybody does their job right, and all supporting systems work as advertised, Galactic Games takes in more money from their customers than they paid out to design, develop, manufacture and distribute the game; as does each of the partners in the supply chain.

This is, of course, a simplified bird's eye view. The real world is far more complex. Buried beneath each element of this model is an intricate web of people, processes, business systems and technologies that work together, more often than not, to ensure the smooth, or not-so smooth, functioning of the business.

Getting the goods

We'll look a little more closely at one element of this processing chain: purchasing the replication and distribution services of Disc Services International. This will introduce us to the transactions between partners that feed this critical corner of the business.

Galactic Games is a long time SAP user. DSI is still on an AS400 system that was installed ten years ago and has been highly customized to meet the specific requirements of their business. This disparity of systems, however, has no impact on the ability of the two partners to do business through EDI.

A minor digression about standards

This is our first important point about implementing EDI. It is based on universally recognized message standards: ANSI X12 in the U.S. and Canada and UN/EDIFACT (for the U. N. Directories for Electronic Data Interchange for Administration, Commerce and Transport) for most of the rest of the world.

EDI standards provide the message structure and a commonly accepted semantics to represent various documents and data objects that can be exchanged within a business process. But these standards are flexible: the specifics of how each EDI transaction is used between two trading partners are worked out in negotiations that lead to the EDI contract.

And this is a second important point about EDI: relationships. How it is implemented between two trading partners is based on negotiation and long-term trading practices that often evolve organically over time between the various individuals involved in both organizations.

Some thoughts about EDI In an XI Environment

Many of these practices are formalized and structured in the EDI contract and reinforced by security measures required to ensure the integrity of the mission critical data exchanged between the two partners through an automated EDI process.

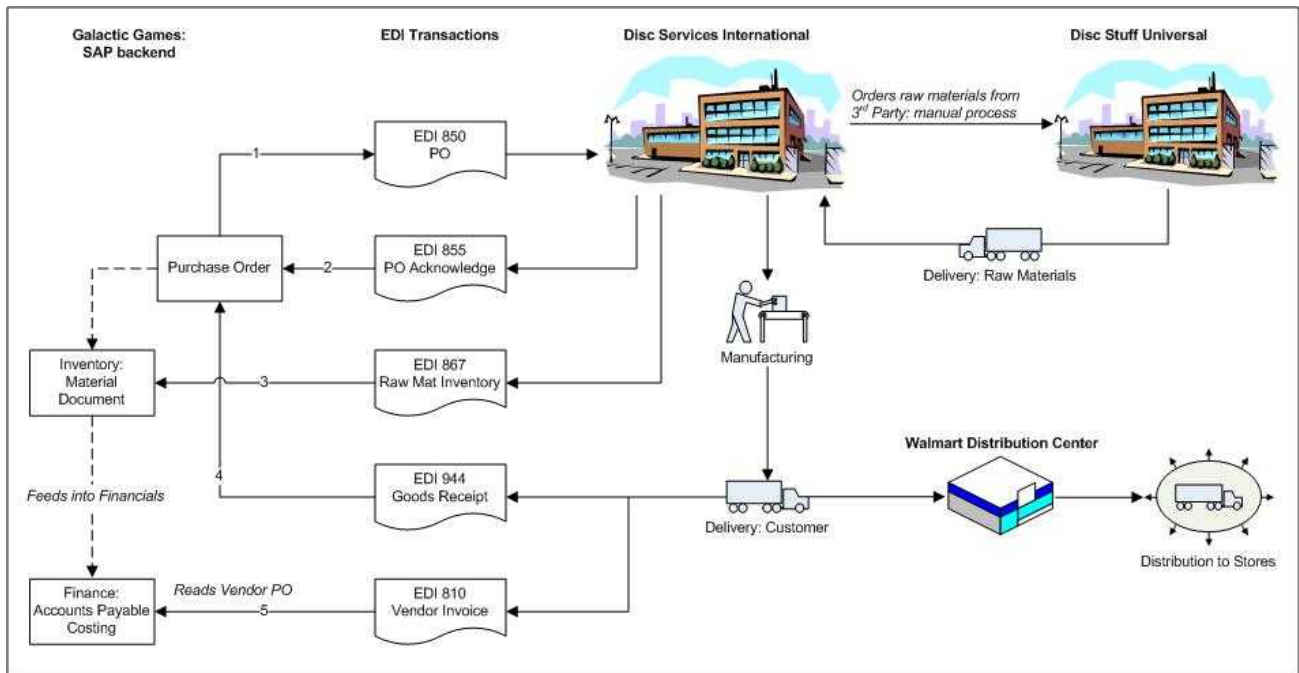
A web of activities

The process illustrated in Figure 2 is a simplified view of one important slice of the overall business. Behind the creation of the Purchase Order is a web of activities that includes such critical events as closing a sale to a customer, receiving the customer's Purchase Orders and posting them to SAP as Sales Orders, and forecasting requirements for ordering of finished good and raw materials, based on such factors as existing customer sales orders and on-hand inventory.

Behind each of these activities is another web of events and activities and relationships.

But we're only looking at Purchasing here, buying the goods that end up as finished product destined for the shelves of Wal-Mart stores across the country.

Figure 2: Transactional overview of finished goods purchasing processing.



Creation of the Vendor Purchase Order in SAP for 100,000 units of Intergalactic Conquest is the starting point. The PO also includes a listing of all raw materials required for manufacturing the finished games as well as instructions for such activities as sticker placement on the jewel case.

Some thoughts about EDI In an XI Environment

When the PO is saved, an ORDERS05 IDoc is generated and kicked out of SAP into the EDI subsystem where it is mapped and translated to an EDI X12 850 transaction and routed via an AS2 server and HTTP/S across the internet to DSI's AS2 server.

The receiving AS2 server routes the message to DSI's EDI subsystem where the X12 850 is translated to a fixed file format and posted to their backend AS400 business processing system as a work order.

DSI first checks their raw material inventory. To manufacture the entire order, they need blank discs, jewel cases, labeling and sticker material. If they don't have enough to cover the entire order, taking into account all the other jobs they currently have on the go, they need to order from their supplier, Disc Stuff Universal, with whom they do not have an EDI relationship.

So they pick up the phone and fax an order and they get an expected delivery date allowing them to set a target date for the completion of manufacturing. Hopefully this will correspond to the expected delivery date of Galactic's Purchase Order.

DSI acknowledges the Galactic PO with an extract from their Work Order, including expected completion date. This follows the same EDI route in reverse back to Galactic Games. The extract maps to an X12 855 transaction and is transmitted to the Galactic AS2 Server and the EDI subsystem where it is translated to an ORDERS05 IDoc which in turns updates the PO with the Acknowledgement information.

Meanwhile, Disc Stuff Universal ships raw materials to DSI, which receives and posts the goods to its inventory management system. And because DSI is managing inventory for Galactic, and because the raw materials are part of the costs of manufacturing the finished games, these costs must be captured in Galactic's inventory and finance systems.

So DSI kicks out details of this inventory to Galactic in an EDI 867 (material quantities and locations: pricing has already been set by prior agreements and are present in pricing conditions in Galactic SAP system).

The 867 flows thru DSI's EDI architecture back to Galactic where it is mapped to an MBGMCR02 IDoc and sent into SAP to post inventory movements that generate material documents and update the financials, which allows Galactic management to track the cost of raw materials in the manufacturing process.

Making it

DSI has everything it needs to kick off manufacturing. The games are stamped onto CD-ROMS and packaged in jewel cases. Artwork is placed, promotional stickers are added and the packages are wrapped in cellophane. Additional stickers and security tags are added to the outside and the finished games packed into boxes and moved into DSI's distribution center for shipment to the Customer.

The games are finally loaded onto pallets, put onto a truck and shipped to a Wal-Mart Distribution Center. Once the games are shipped, DSI's responsibility ends: they generate a Warehouse Receipt from their Delivery System and send Galactic Games an EDI 944 transaction.

The 944 is translated into an MBGMCR02 IDoc in Galactic's EDI subsystem and sent into SAP, where it posts a Goods Receipt against the original Purchase Order. Inventory is updated by the GR, including back-flushing (consumption) of raw materials and the PO is completed, its history updated with the material documents that have balanced inventory with the goods consumed by the Order and with the shipped product itself.

Balance between consumption and production has been achieved and the financial system is updated.

Some thoughts about EDI In an XI Environment

La piece de resistance ...

Now is the moment of truth: the issuing of the invoices. DSI's financial system kicks out an invoice that is translated to an X12 810 and transmitted to Galactic's EDI subsystem. The 810 is translated to an INVOIC02 IDoc and sent to SAP where it posts a Vendor invoice, pulling pricing information from the original PO. The price that the Vendor lists on his 810 will not post to the invoice in the Galactic system. It should, however, be the same as the price sent to DSI by Galactic when the PO was originally transmitted.

If all checks pass, the invoice is created, financials updated and Accounts Payable, in a timely and courteous manner, cuts a check.

At the same time, although it's not illustrated in our current process, Galactic issues its own invoice to Wal-Mart. While everything is connected, that's part of another process.

Let's get technical

This is the big picture, admittedly from the proverbial dizzying height, but it provides the foundation to begin considering the technical solution. What does this picture tell us? What are the pieces that we need?

First the obvious: we need to be able to issue and post electronic documents: Purchase Orders, Acknowledgements, Inventory Movements, Goods Receipts, Invoices and so on. This means a backend system. For Galactic, it's SAP while DSI enjoys a highly customized JD Edwards system.

Regardless of what backend system is used, the business documents it issues are mapped to standard EDI X12 transactions that carry document data from the issuing system to the receiving system.

And while the X12 transactions follow universally recognized standards, businesses are like the people who run them: no two operate in the same manner. The X12 transactions need to reflect all the idiosyncrasies of the business process behind the business document at both the sending and receiving ends.

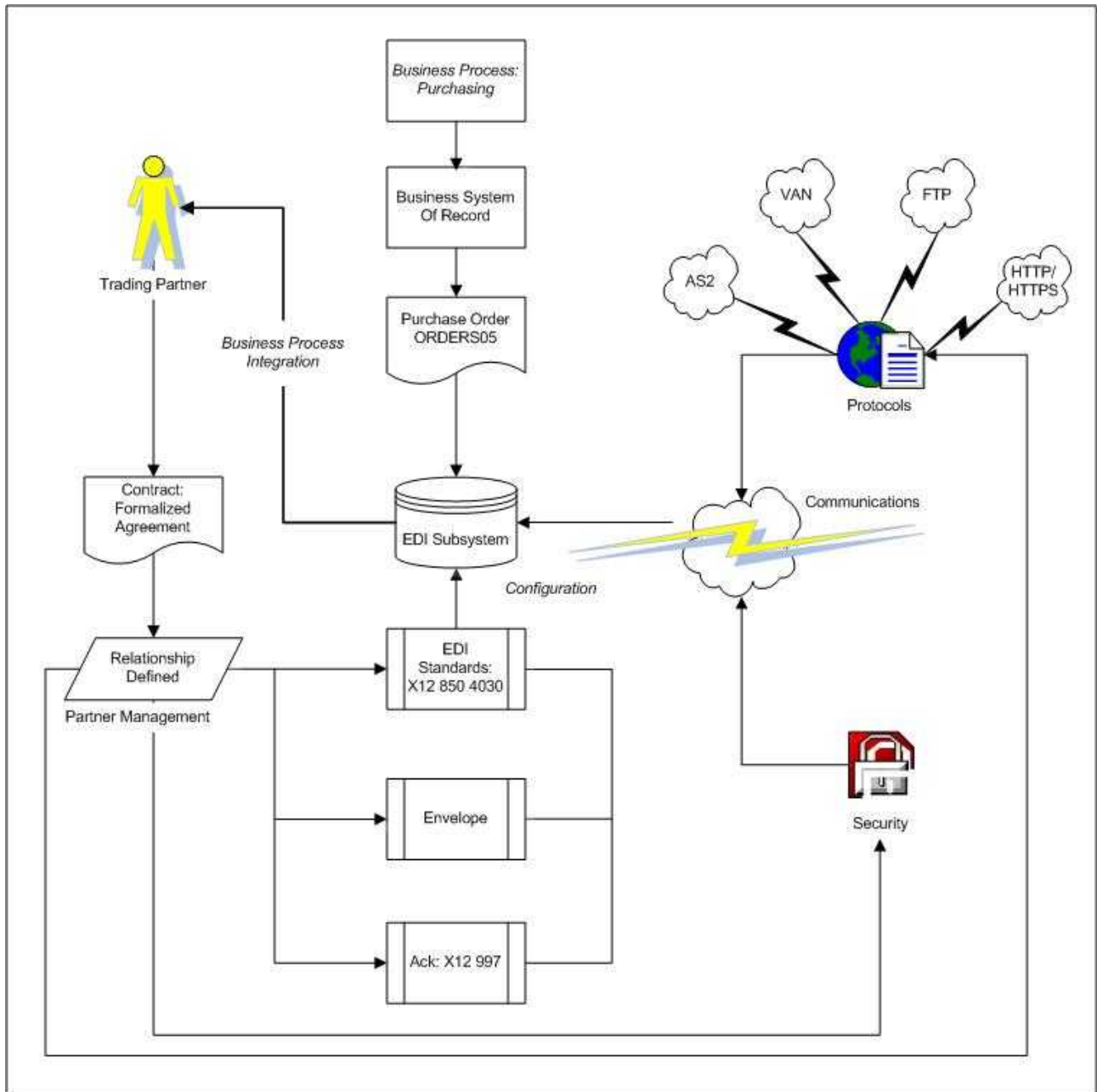
An agreeable relationship

This generally means an agreement must be reached between the trading partners on how the X12 transactions will be populated and transmitted. This is done through informal contacts between both organizations at the business and technical levels, and as part of formal negotiations that result in a detailed definition, codified in a legal contract, of the EDI relationship that will exist between the two partners.

Figure 3 provides a simplified overview of the elements that will be defined. The business process provides the data. The backend system that enables the business process, together with the EDI and Communications subsystems, provide the technical processes and tools to translate the agreement, the codified and formalized business relationship, into an EDI architecture.

Some thoughts about EDI In an XI Environment

Figure 3: EDI requirements and setup overview.



The following elements are required to build this EDI architecture:

Some thoughts about EDI In an XI Environment

1. An agreement about EDI standards and transactions that will be used to exchange data between the partners. This speaks to the cross-company business processes being integrated and includes:
 - Formats, such as X12, EDIFACT.
 - Versions, such as 4010, 4030.
 - Transactions, such as 850 (Orders), 810 (Invoices), 867 (Inventory Movements), 944 (Warehouse Receipts), and so on.
2. This agreement also determines how these transactions will be used. Both the 867 and 944, for example, can be used to post Inventory Movements and Goods Receipts. Whether one or the other or both are used could be affected by what either partner is exchanging with their other partners, the setup of their EDI subsystems or even the knowledge and comfort level with particular transactions within either partner's EDI group.
 - While simplification is always preferred, the overall context and other existing relationships can also play a role in determining the standards selected.
3. Type of acknowledgements required (i.e., 997) and the point at which they are returned, as well as error handling procedures and means of communicating problems back to the partner.
4. Details of communications protocols and/or services to be used in the exchange of data including VAN mailbox addresses; AS2, HTTP/S and FTP servers; and use of Perimeter Servers in the DMZ or other means of poking through the Fire Wall.
5. Server IP addresses and/or URLs or any other communication addresses that will be used to exchange messages.
6. Security details including Encryption Standards, Digital Certificates, Trusted System setup, User Keys, Server Login, Password policies and procedures and so on.

These definitions feed the configuration of the EDI and communications subsystems, enabling the exchange of business documents between the partners.

The match maker

Business Relationships, Process, Transactions and Communications: it all comes together in the EDI subsystem. A lot of the work of the subsystem is in identifying and matching up objects to each other. Figure 4 depicts a simplified view of the technical flow through the EDI subsystem.

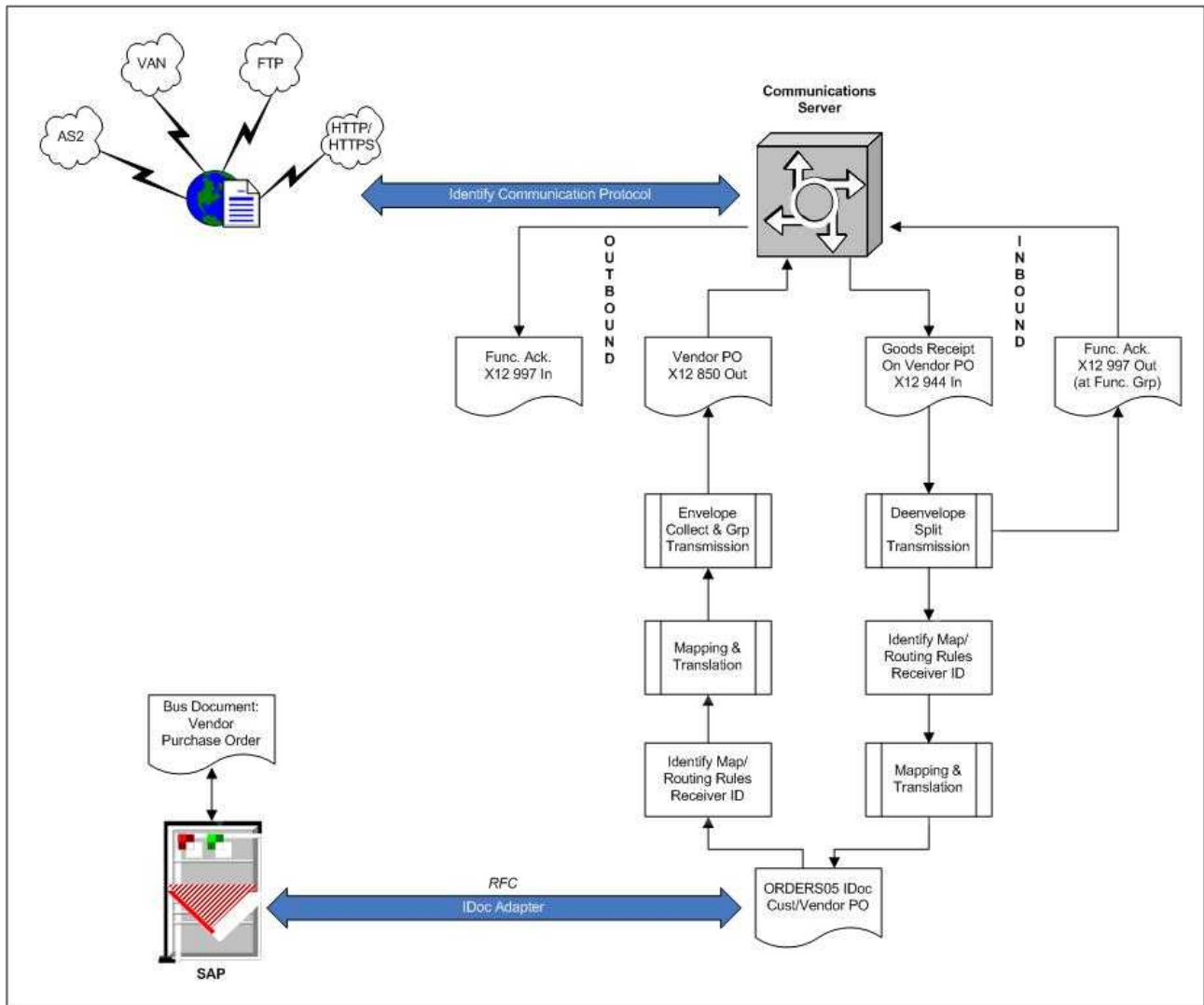
We'll get a clearer picture of what needs to be done by following the trail of an outbound Purchase Order and inbound Goods Receipt within Galactic's SAP backend and EDI subsystems. But first, a few words about IDocs, SAP and EDI.

SAP's traditional approach to EDI

Traditionally, EDI in SAP was implemented as an interface between SAP and an EDI Subsystem, such as Gentrant, Trusted Link, or the more modern ones like Mercator, TIBCO and GIS (Gentrant Integration Suite).

Some thoughts about EDI In an XI Environment

Figure 4: A bird's eye view of the EDI subsystem process flow.



IDocs – Intermediate Documents – were always central to this approach. Originally derived from EDIFACT EDI transactions, IDocs are both the medium and the message by which SAP business documents are exchanged with the outside world.

But IDocs, and the myriad array of software services that support them, do not natively support connections to trading partners via typical EDI protocols. The IDoc interface supports RFC transmission to remote systems, but was never designed as an EDI subsystem, although with enough time, money and experienced ABAP developers, a custom EDI system could be built within SAP. But it would be very expensive.

Some thoughts about EDI In an XI Environment

In the past, EDI interfaces with SAP were mostly handled through IDocs written to physical files moved between the SAP application server and the EDI subsystem, mostly because legacy EDI subsystems tended to reside on mainframes.

In the last few years EDI subsystems have migrated from the mainframe to a more nimble client server, web-centric model incorporating XML, BPEL, SOAP, Java and the whole alphabet soup of emerging open standards that we've come to associate with the modern approach to Enterprise Application and Business Process Integration.

At its most basic, the SAP EDI architecture consists of EDI-enabled applications that support automatic processing of business transactions and the IDoc interface, which consists of:

1. IDocs to contain, structure, process and move data to and from SAP business documents, with their underlying processing functions, including an extensive system of customer and user exits.
2. Configuration settings that control the population and process flow of IDocs into, out of and through SAP. These settings include, but are not restricted to:
 - Logical Systems identifying source and target systems.
 - RFC destinations defining connection and login parameters for target systems.
 - Ports pointing to the mechanism by which the IDoc is exported from SAP. The most commonly used are RFC Ports, pointing to an RFC destination, and File Ports, pointing to a Directory and Filename where the IDoc is spooled out to a physical file.
 - Partner Profiles detailing systems and trading partners, who must exist in SAP master data, and the inbound and outbound messages exchanged with them.

EDI Partner Profiles are critical for the routing of messages into and out of SAP as they link the trading partner to the message and the port and the process code which itself is linked to the function module that processes the IDoc.

The Sending (inbound) or Receiving (outbound) Partner in the IDoc control segment comes from, or is confirmed against, the Partner Profile. It is used by the EDI subsystem to identify the EDI Sender or Receiver ID, used to route the EDI transaction to its final destination.

3. A sequenced set of standard function modules that are executed in a defined order. These functions manage all processing services required by the interface as defined in the configuration described above, including:
 - Inbound: RFC enabled import, by file or API, callable by an external system.
 - Outbound: RFC enabled export, by file or API, to an external EDI subsystem.
 - Validation of control segment data including message and basic type structure and syntax; sending and receiving partners; output control parameters; and other key interface configuration data defined through, and linked to, the Partner Profiles.
 - Posting of inbound IDoc data to SAP transactions and extraction of outbound data from the SAP business object.
 - Error and status reporting, including workflow notification of failure.
4. Services such as IDoc monitoring, editing, reprocessing, reporting, testing and archiving.

Some thoughts about EDI In an XI Environment

5. Any custom ABAP programs that may be written to manage the processing, trouble-shooting, reprocessing and reporting of IDocs.
6. An EDI subsystem.
7. Audit and reporting tools to keep track of IDoc/EDI status across the entire architecture.

Typical of SAP, there's a lot of stuff, but at the risk of belaboring my cliché, it really is not rocket science. You do need to get a handle on it, however: the standard IDoc interface provides an exhaustive technical foundation for business process integration through IDocs and EDI in SAP. So let's get back on the trail and follow the process through SAP and the EDI subsystem.

Outward bound

Galactic cuts a Purchase Order in SAP. The system writes an output record to table NAST checking for a Partner Profile to match Message Output configuration options defined for DSI in the PO.

Since we've configured Output to immediately generate and export an IDoc, our PO output record in NAST is immediately checked against the Partner Profile and matched against the profile's Message Control, where the Application, Message Type and Process Code are identified.

This match links both to the functional use of the IDoc basic type that will be populated (i.e., PO for Finished Goods rather than Stock Transfer Order) and to the outbound function module that will be called.

The outbound processing function is then called and the relevant data extracted from the PO. An ORDERS05 basic type is populated with this data and created on the IDoc database.

The Partner Profile also tells us that the IDoc will be transferred immediately so the outbound process kicks in. The IDoc is read from the database, the Port and RFC Destination identified from the Profile and the IDoc sent out via RFC to the EDI subsystem. We're using RFC here for the sake of argument: the IDoc can just easily be output through a file port to a physical file.

Moving on out

The EDI subsystem picks up the IDoc transmission and identifies it, from the Control Segment, as an outbound ORDERS05 destined to be sent to Disc Services International as an X12 850. DSI is identified from the IDoc Control Segment (EDI_DC40) field RCVPRN and the 850 transaction from field STDMES.

A translation map is then identified and executed and the IDoc is transformed into an outbound 850 at the transactional level.

It's important to remember at this point that one SAP IDoc equals only one Purchase Order. We could be issuing hundreds or even thousands of Orders: rather than send each one individually, they need to be grouped together and sent in a single transmission.

Within the X12 transmission, each Order is contained inside one ST segment (Transaction Set Header, identifying the X12 transaction being exchanged) and one SE segment (Transaction Set Trailer). The ST segment identifies the transaction (850, 944) and a Transaction Set Control Number, a unique sequential ID that identifies a particular transaction (one Purchase Order) within a larger Functional Group.

The transaction sets are collected into a Functional Group that is contained within one GS segment (Functional Group Header) and one GE segment (Functional Group Trailer). One GS group can contain an unlimited number of PO's, each within its own ST set.

Some thoughts about EDI In an XI Environment

The GS segment contains key control information used to logically group the ST transactional groups. This includes the following data elements:

1. Functional Identifier Code: identifies a group of application related transaction sets such as 850 Orders.
2. Application Sender's ID: the identification code agreed to by both partners and recorded in the EDI subsystem that identifies the sender of a transmission.
3. This code, and the Receiver ID below, can represent one of any number of standard identifiers such as a DUNS number, DUNS with suffix, UCC, EAN, and so on, or it can be a unique standalone identifier agreed to by the trading partners. This ID code is generally different from the SNDPRN (Send Partner) or RCVPRN (Receiving Partner) in the control segment of the IDoc.
4. Application Receiver ID: the identification code agreed to by both partners and recorded in the EDI subsystem that identifies the receiver of a transmission.
5. Date, Time and Functional Group Control Number.
6. Group Control Number: a unique ID that identifies each GS group within the Interchange. It is also used to identify the GS group received in the 997 Functional Acknowledgement sent back to a partner when an EDI transmission has been received.
7. X12 version number for the included transactions.

The GS Loops are themselves grouped within an Interchange, defined by an ISA segment (Interchange Control Header) and IE segment (Interchange Control Trailer). There is a one-to-many relationship between the GS and ISA groups.

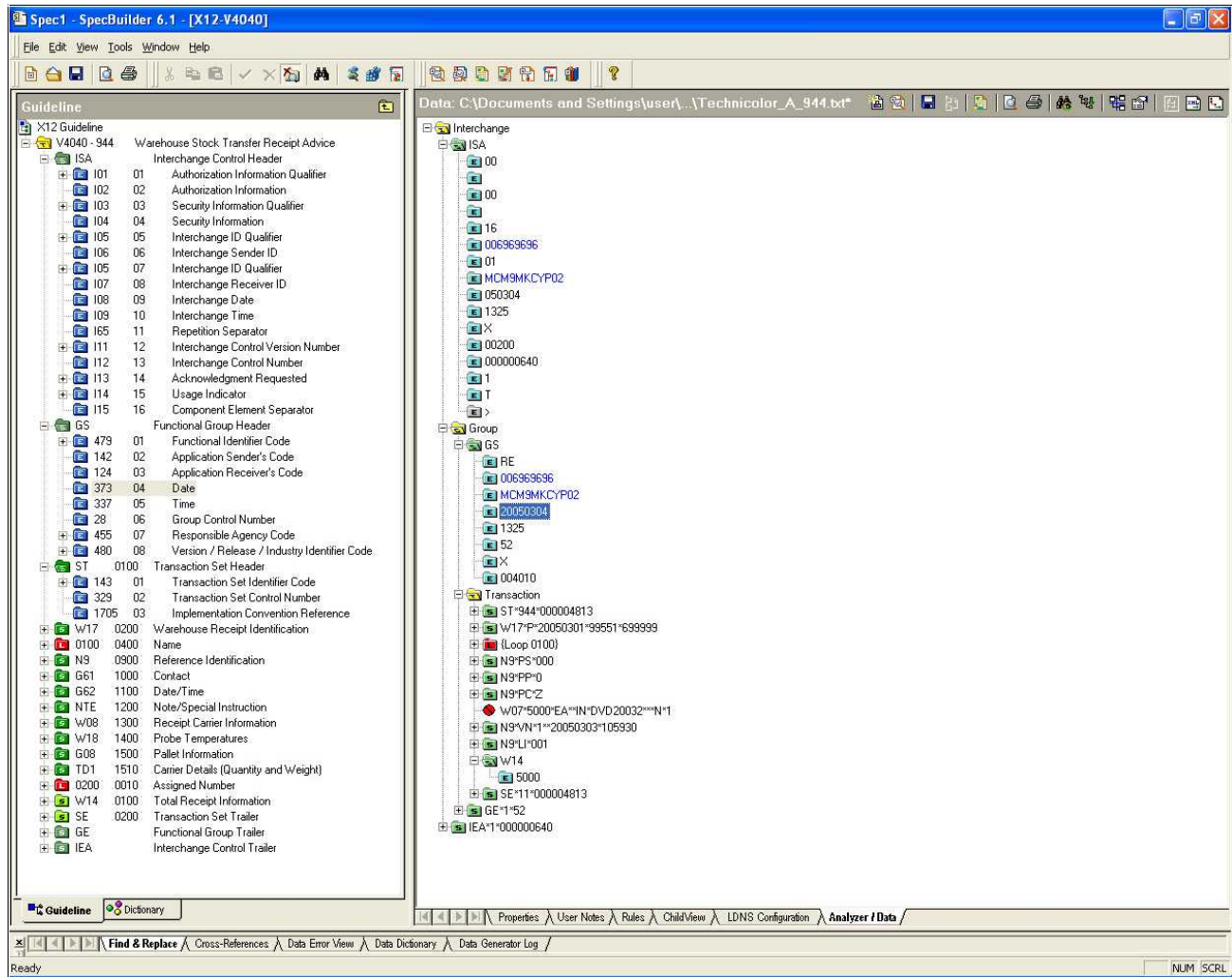
Like the GS segment, the ISA header contains control information that logically groups the entire interchange. This includes data elements such as:

1. Authorization and security information with qualifiers.
2. Sender and Receiver IDs with qualifiers identifying the type of code, standard or custom, being used for Sender and Receiver.
3. In the ISA segment, the Receiver ID is used in the communications server of the Sender to route the transmission while the Sender ID is used in the communications server of the Receiver to identify the source of the transmission and to begin the process of identifying all the other elements of the EDI relationship – security, standards, transactions, etc., enjoyed with the partner.
4. Interchange date and time.
5. Repetition and Element separators used.
6. Interchange Control Number: yes, there can be many Interchanges within a Transmission and each needs to have its unique ID.
7. Acknowledgement requested flag: 1 means an acknowledgement must be sent when the Interchange has been received and processing begun, generally when the GS groups are being processed.

An X12 Interchange for an inbound 944 is illustrated in Figure 5. Sender and Receiver IDs have been changed to protect the innocent.

Some thoughts about EDI In an XI Environment

Figure 5: View of an X12 944 EDI transaction.



This process of grouping electronic business documents together within the Transactional, Functional and Interchange groups is known on the outbound as Enveloping and is one of the critical functions of the EDI subsystem. More than a map, it accumulates and packages in logical units all the business documents that need to go out to a partner in a single communications session.

A critical step in this process is the identification of the EDI Receiver ID which almost always involves a translation of the SAP Vendor number in EDI_DC40-RCVPRN. As we have noted, this translation is required for both the GS and ISA segments. This information is critical for identifying routing and communications rules to move the X12 Transmission from the EDI subsystem to the Communications Server that will ultimately send the X12 to DSI.

Some thoughts about EDI In an XI Environment

The Receiver ID translation could be hard-coded in a map, if one map is built for each transaction and version used by each partner. Or it could be pulled from a reference table, in the EDI subsystem or SAP, linking the SAP Vendor Number in EDI_DC40-RCVPRN to the EDI Receiver ID.

An equally important translation is the Sender ID identifying Galactic that goes to DSI. SAP defaults the Logical System name for the Client and System that exports the IDoc in the Control Segment Send Partner field (EDI_DC40-SNDPRN), a meaningless value to DSI.

Galactic has a unique ID, agreed to with DSI, that DSI uses to identify it as both a Sender and Receiver. This ID is almost always different from the EDI ID's that Galactic uses for every other partner.

Communications and Security Protocols and procedures, including, in our case, target URLs to the AS2 Server and beyond to DSI's AS2 Server, are linked to these Sender and Receiver ID's.

So now we have it and the EDI system knows how to call the AS2 Server with the bundled transmission. The Server is called and it immediately recognizes the Receiving partner, gets the URL and whatever login information and security certificates it needs, and posts to DSI's AS2 server.

DSI receives the 850 and passes it to their EDI subsystem. The ISA segment is read and it is determined that the Acknowledgement flag is set to 1. DSI's subsystem needs to generate and return an X12 997 Functional Acknowledgment informing Galactic that the transmission has been received.

But this tells us nothing about whether or not the Order posted in DSI's system. In our actual process, Galactic needs to know this information. Rather than send back a 997, the partners have agreed to wait until the Order posts successfully in DSI's system, which then generates an 855 PO Acknowledgement to send back and update the PO in Galactic's SAP system.

Bringing it home

The inbound 944 will complete the Purchase Order by posting a Goods Receipt, updating inventory and setting up the financials for receipt of an invoice from DSI.

Processing is triggered by the inbound HTTP call from the Sender's AS2 Server. Galactic's AS2 Server identifies and verifies the Sender through the ISA segment, admits the transmission and immediately routes it to the EDI subsystem through another HTTP call.

The EDI system must first parse the transmission into separate interchanges which are then subjected to Deenveloping, the process of stripping away the ISA, GS and ST envelopes to get at the Transaction Sets beneath and to prepare these for mapping and translation to IDocs.

A lot of information is gleaned from the segments during this splitting process including the fact that a 997 Acknowledgement must be sent back to the Sender. The 997 is a standard X12 transaction that must be populated, Enveloped and sent back out to the Sender through the AS2 server.

Since the acknowledgement is at the Functional Group Level, segment AK1 of the 997 is populated with the Functional Identifier Code (RE for the 944 Warehouse Stock Receipt Advice that carries our Goods Receipt) and GS Group Control Number. Most dedicated EDI subsystems automatically generate these Acknowledgements.

When the 997 is built and enveloped, the EDI subsystem routes it back to Galactic's AS2 Server for transmission to DSI.

Some thoughts about EDI In an XI Environment

Meanwhile, the EDI subsystem identifies the Receiving system (the SAP Logical System for the Client and System in which the Goods Receipt will post) and the map that will translate the 944 Transaction Sets to MBGMCR03 IDoc basic types. The map is called and the IDocs created.

The IDoc adapter is then called which transfers the data to SAP through an RFC call to SAP function IDOC_INBOUND_ASYNCHRONOUS. This kicks off standard IDoc interface services. EDI technical system parameters are checked, the IDoc control segment is translated from the external EDI_DC40 format to the internal EDIDC and the data segments translated from the external EDI_DD40 to the internal EDIDD.

The Partner Profile is checked and verified. If the Check Syntax flag is set in the Profile, a check is run on the syntax – the structural integrity – of the IDoc. It is then written to the IDoc database.

If there are any errors in the Partner Profile or the syntax check, the IDoc is written with an error status. If there are no errors, the IDoc is written to the database at status 64, ready to be passed to the application.

At this point, the application takes over, either immediately or by scheduling a call to program RBDAPP01. Either way, function IDOC_START_INBOUND is called, which in turn calls function IDOC_INPUT. The IDoc is read from the database, the inbound processing function identified from the process code in the Partner Profile, and the business document is posted, in this case, a Goods Receipt recorded against the Vendor Purchase Order.

The EDI processing cycle is now complete. So where does XI stand in all of this?

EDI and XI

Considering how recent an arrival it is to the Enterprise Integration space, SAP XI provides an impressive environment for developing interfaces and supporting business process integration, particularly between SAP and other systems.

We can set up XI to handle basic EDI interfaces. It natively includes most of the functionality required to build a working EDI subsystem. We can design and configure an interface that translates an IDoc to an X12 transaction through a map developed within XI and then send it to, or receive it from, a trading partner. Communications can be to a traditional VAN or to an AS2 server for encrypted transmission across the internet through XI File/FTP, HTTP or SOAP adapters.

You can even use these, and other, adapters to communicate directly with your trading partners' systems, assuming your communications and monitoring requirements are fairly basic. Of course, you can always extend the capabilities of the adapters by adding custom Java code to the Adapter's module subscreen in Remote or Local Java Beans or Java Archives.

But as far as industrial strength EDI goes, we still have some important gaps in the standard XI product. These gaps can be addressed by developing some custom XI business processes and by investing in some external, third-party tools.

By looking at some of these gaps we can consider ideas that may be helpful in plugging them. The aim here is to raise ideas, topics for discussion, in which XI may be used on its own as an EDI platform, without looking at expensive third party plug-ins such as Seeburger's or iWay's EDI Adapters.

This is not, however, an exhaustive discussion of XI EDI architecture nor is it meant to provide recommendations for any specific business situation.

Some thoughts about EDI In an XI Environment

The Source ... Templates and Maps

Hang on to your hats. We're about to engage in a rather convoluted discussion about standards and mapping. The good news here is that the end point is very simple.

We can't build maps without source and target file structures. But unlike dedicated EDI subsystems such as Gentrans Integration Suite 4.0, XI does not provide a library of EDI standards.

SAP does provide IDoc metadata and an environment in XI to manually create custom Data and Message Types. But standard EDI transactions are complex hierarchical documents with multiple versions that are updated on a regular basis.

They also contain long lists of qualifiers, codes and rules that define how certain fields are populated in different contexts by different trading partners. Enough of these must be included to conform to the EDI standards used by your trading partners.

In addition, trading partners upgrade their use of EDI standards, some much more than others. Wal-Mart, for example, upgrades fairly often and is currently moving from X12 version 4030 to version 5010. This upgrade could include new data elements and new qualifiers for existing data elements. All Wal-Mart suppliers are obliged to update their maps and processes if they want to keep doing business with Wal-Mart.

This would be a very difficult and time-consuming process if it was being done by manually creating and maintaining Data and Message Types.

XI supports import of file structures from external systems through External Definitions that can then be set up as Message Interfaces. But there's a problem with this approach: you can only import DTD, XSD and WSDL into External Definitions. To understand this problem, it's useful to look at how EDI standards, or Guidelines, are generated and exchanged.

Browsing the libraries

Aside from the more widely used dedicated EDI subsystems, there are standalone tools on the market that provide full libraries of EDI Standards. EDIFICS SpecBuilder and GEFEC EDIFIX are the two best known.

I'll be looking at SpecBuilder since I've worked with it in an XI environment. It includes over 18,000 templates for virtually every EDI standard in the world, in all its versions, including all qualifiers, codes and special rules.

It also provides a dynamic environment in which to create EDI specification documents that can be used internally and to help trading partners set up EDI exchanges. It can even generate specifications from existing EDI data transmissions.

For the EDI aficionado, it is an indispensable and way too cool tool.

SpecBuilder supports many standards for exporting its EDI templates, including Gentrans DDF, BizTalk, CSV and its proprietary formats such as ECS, EDS and its own XML Schema dialect called XData XSD. XI does not comfortably support XData XSD. You can import a SpecBuilder XSD template into an XI External Definition but every segment and loop in the transaction is treated as a separate message.

SpecBuilder can also export gXML (Guideline XML), an EDI Guideline format introduced by EDIFICS that is becoming a widely accepted industry standard for describing and exchanging EDI transaction templates.

In gXML, the structure and attributes of the EDI transaction, including envelopes, qualifiers and codes, are described in an XML document that is validated by a separate DTD.

Some thoughts about EDI In an XI Environment

XI does not support gXML. To get our EDI Transaction templates into XI, we're left with workarounds that leave us relying on additional tools.

You can, for example, build the maps in a standalone mapping tool that supports gXML, such as Contivo Analyst. You would then export the map into XI from Contivo by generating Java transformation code that would then be brought into XI as an Imported Java Archive and configured with an Interface Mapping.

But even with a Contivo map, the X12 Interface used is needed in XI as a Message Interface. It can be exported from Contivo as an XSD file and then imported into XI as an External Definition. XI does support EDI templates exported from Contivo as XSD files, regardless of their origin.

But you must first strip qualifiers from the XSD: an X12 850 4030 fully loaded with qualifiers can easily fill 85 megabytes. XI chokes when XSD's this large are brought into External Definitions. Without qualifiers, the 850 XSD is less than a megabyte in size. Qualifiers are not required in XI Message Interfaces, although they are important elements in mapping in Contivo and other mapping tools.

So you would need to export a second version of the X12 from SpecBuilder without the qualifiers into Contivo, re-export it as an XSD and then import it into XI as an External Definition.

Confused yet? It's not as difficult as it sounds, assuming you have all the tools.

The easy solution would be for SAP to build support for gXML and other EDI Guideline standards into the XI External Definition. gXML has a much smaller footprint than XSD: an 850 version 4030 with all qualifiers produces a file of about 7 megabytes.

This one change alone would greatly simplify development and maintenance of EDI maps within XI.

What's the big deal about qualifiers?

In mapping tools such as Contivo Analyst and in the standalone mapping program used by GIS 4.0, qualifiers can be used to create standalone instances of looping segments that are specific to a particular qualifier in both the source and the target structures. Contivo calls these virtual segments.

This eliminates the need to write conditional code to test for a particular qualifier when mapping fields from similarly qualified segments within loops. Fields within matching virtual segments are simply mapped to each other. For example, an identification field in an N1 segment of an 850 configured to hold a GLN number can be directly mapped to a corresponding field in an E1EDKA1 segment of an ODEERS05 IDoc configured for the same data type, without writing a line of conditional or looping code.

This sweet feature alone considerably speeds up mapping.

Addressing the envelope

This is a biggie. It touches on just about everything: transactional standards, mapping, splitting, collecting, trading partner management, communications and transmission. When the EDI transmission hits the VAN or the AS2 Server, the Envelope, with its Sending and Receiving IDs, provides the data that triggers the communications server's routing mechanisms that send the transmission on its merry way, whether to the trading partner or inbound to SAP.

To support this, we need some EDI specific data such as EDI Transaction and Version and Receiver and Sender IDs that are not natively accommodated in XI. This is most relevant for outbound transmissions.

Some thoughts about EDI In an XI Environment

EDI Transaction and Version can be stored in the EDI Standard subscreen of the outbound Partner Profile. They are then automatically inserted into the IDoc Control segment fields EDI_DC40-STDMEs and STDVRS during outbound processing.

Receiver and Sender IDs can be stored in SAP custom tables or even in an EDI extended Customer master. These values could then be read during IDoc outbound processing and put into the Control segment fields EDI_DC40-SNDLAD and RCVLAD (EDI Sender and Receiver logical addresses). Or they could be pulled from the SAP tables during XI mapping using custom Java functions.

But it only begins with finding the right data and mapping an IDoc to an X12 transaction. As we have seen, industrial strength EDI is a tricky proposition involving construction and deconstruction of highly complex data structures. We need to build XI Business Processes to handle this.

We'll take a quick look at potential scenarios for an outbound and inbound BP. The purpose here is not to provide detailed procedures for building an EDI-enabling Business Process in XI. It is simply to outline steps that may be used in an Enveloping/Deenveloping BP.

Outbound Enveloping

On the outbound, "collection" is the word of the day. EDI transactions mapped from IDocs need to be collected and associated with all three envelopes, which also need to be correctly populated for the Sending and Receiving Partners and other key EDI control data.

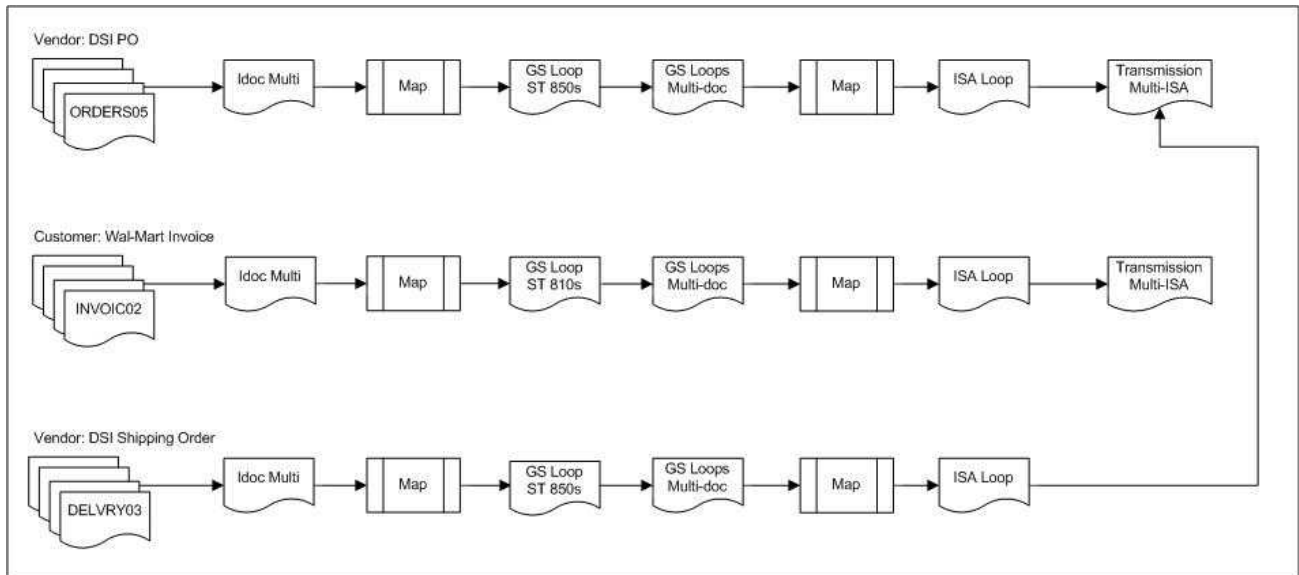
Then the highest level envelope – the ISA Loop – needs to be further consolidated into a single transmission, usually with multiple ISA Loops, potentially containing different types of EDI transactions, before being sent to the communications server.

Incoming IDocs (into XI from SAP) of the same logical message type and for the same Customer or Vendor would be collected (appended) into a multi-line IDoc container to be used as the source in a many-to-one map against a single-line target container defined with one GS and multiple ST Loops.

All ST Loops within the GS container would hold the same X12 transaction and version. The map would also need to assign a sequential control number to each of the ST Segments within the GS Loop.

Some thoughts about EDI In an XI Environment

Figure 6: Enveloping: Schematic overview of on-going processes of extraction and collection.



The individual GS Loops with their multiple ST Loops coming of the translation step would then need to be appended into another multi-line container which would then be run as a source file against a second many-to-one translation step targeting a single-line container defined with one ISA and multiple GS Loops.

All GS Loops within the ISA container would need to contain ST Transaction sets that followed the same X12 version, although theoretically they could contain different X12 transactions. The map in this step would need to assign a unique, sequential control number to each GS segment within the ISA Loop.

The individual ISA Loop containers would then be appended to a final multi-line container with multiple ISA Loops, each of which contains nested GS Loops with multiple ST Loops.

The multi-line ISA container would then be used as a source file for a final many-to-one translation step to a single container that would then be used to call the Send step for an FTP or SOAP Adapter into a VAN or an AS2 Server.

This last container could group ISA Loops holding different X12 transactions and versions. The map would also need to assign a unique sequential ISA control number to each of the ISA Loops within the transmission.

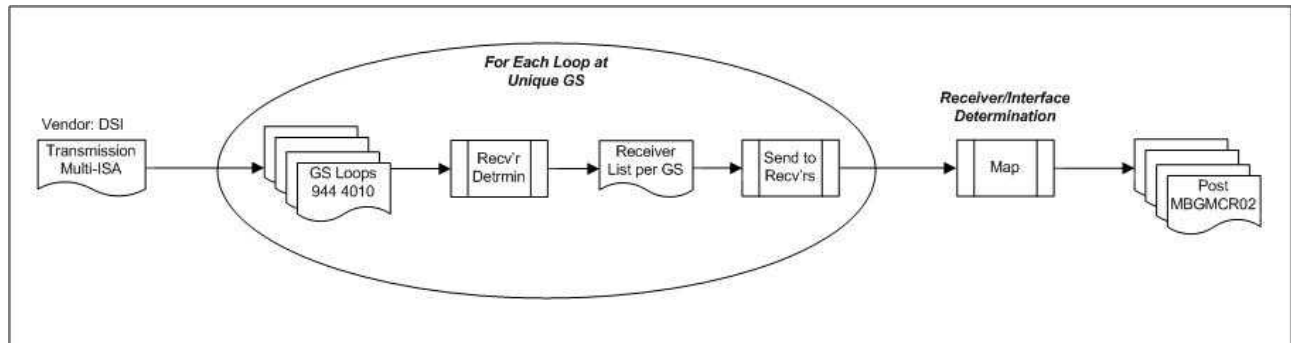
Inbound Deenveloping

The operative word on the inbound is “splitting”: ISA, GS and ST envelopes need to be stripped away, the transactional data sets split into individual transactions, and the translation map identified and called. But this would be a less complex process than the outbound collection.

The transmission would be received from the communications server into a container structured with multiple ISA Loops, each containing multiple GS Loops with their nested ST Loops.

Some thoughts about EDI In an XI Environment

Figure 7: Deenveloping: .Splitting out the unique transactions.



We need to remember here that each ISA Loop could contain different Transactions and Versions from each other. And that within each ISA, there could theoretically be GS Loops with different transactions, although all need to be the same X12 Version.

The split is at the GS Loop level, since each GS contains the same EDI Transaction and Version. We'll need a For Each or Par For Each block with a complex local correlation that would identify each GS Loop within the Transmission on the basis of EDI Transaction and Version.

Each unique GS Loop identified would be appended to a multi-line container with an abstract interface type structured like a GS Loop with multiple ST Loops nested inside it.

The For Each block processes the multi-line container one line at a time. In other words, the GS Loop being processed within each For Each pass is handled within a single-line container.

A Receiver Determination step would be called to get a list of receivers for each GS Loop Interface. Next would be a Send Step that would pull its Receivers from the Receiver List generated by the Receiver Determination Step.

The map that runs the final split and transforms the multi-transaction GS Loop interface into multiple IDocs would be called through Receiver and Interface Determination. XI SP14 supports a direct call to a splitting map (a 1 to n transformation) from within Receiver Determination: the split no longer needs to be supported by a Business Process.

At this point, the IDoc Adapter is called and the IDocs post to SAP.

Communications and Security

The most mission critical EDI transactions are transmitted through a VAN or AS2, both of which allow for a high level of security including stringent encryption standards.

VANs, which generally are owned by large networking companies and charge a per kilobyte fee that lead to very high monthly charges, are slowly being replaced by AS2, which offers secure encrypted transmissions across the internet, eliminating the need for costly VAN charges.

Of course, the means of transmission depends on what your partners are doing. If they're still on a VAN, you are too. If they, like Wal-Mart, are on AS2 or transitioning to AS2, you will have to be as well.

Some thoughts about EDI In an XI Environment

An AS2 server is generally a good investment. It not only sets you up for the day when all your partners will be on AS2, it can also enable and support secure communications to a VAN and to your XI system. Many industry-leading dedicated EDI subsystems, such as GIS 4.0, provide comprehensive AS2 services and some even sell standalone AS2 servers.

iSoft Commerce Suite provides a good example of the kind of functionality you'll find in a dedicated AS2 Server. It supports all EDI internet transmission standards including AS1, AS2, HTTP, HTTP/S, SFTP, SMTP and Rosettanet. There is also:

1. Support for multiple data types and security standards and a wide range of EDI and backend systems.
2. A high-performance multi-threaded, multi-tasking scalable architecture to maximize throughput and high-availability failover and restart.
3. Enables complete privacy, authentication, data integrity and non-repudiation of all transactions.
4. Supports encryption and decryption certificates for all major security vendors and provides a Public Key Infrastructure solution that can generate industry-standard certificates.
5. Provides a web-based interface for Trading Partner Management.

This last point is important. The EDI system needs a central place to define and maintain the critical pieces of information about the relationships between your trading partners and your organization that enable the exchange of EDI transactions.

This includes such basics as EDI Sender and Receiving IDs, EDI contact information, URL and IP addresses of your partner's systems, message delivery options, security parameters, digital signature creation parameters, encryption certificates used, and so on.

This information needs to be available at runtime, when the X12 transmission has been received and needs to be routed either into your organization or outward bound to your trading partner.

Wrapping Up

To sum up the approach addressed in this article, XI can form the backbone of a dynamic EDI subsystem with a little tweaking and a little help from its friends. The basic pieces that should be looked at include:

1. A library of EDI standards.
2. Custom BP development in XI.
3. Some custom IDoc development in SAP.
4. An AS2 Server to support EDI Trading Partner Profile management, security and communications with your partners.

I'll say it one last time: EDI is not rocket science. But don't underestimate the effort: EDI is still a complex and mission-critical development that does not end with Go-Live. We haven't even considered monitoring and reporting, for example, both functions that will be critical in the production environment.

This brings us to our concluding thought: once deployed, you are stuck with the EDI system you've built. You'll need to run it in a production environment for years to come. There will also be ongoing development tweaks as your partners upgrade their transactions, or as you upgrade, add new partners and transmissions: it never ends.

Some thoughts about EDI In an XI Environment

And no matter how well it's built, it will fail at times and you'll need to be able to recover from that failure with minimal business loss.

You owe it to your organization, to your trading partners, and to the people who depend on, and who will directly support, the new EDI system, to carefully think through all the business and technical issues.

In the meantime, SAP has a marvelous opportunity to better address the needs of a very large and rich market by extending XI functionality to more fully support standard EDI protocols and requirements: we're not far now and a little effort can bring us a whole lot closer.

Some thoughts about EDI In an XI Environment

Author Bio



Author Name: Emmanuel Hadzipetros

Company: NBC Universal, Universal City, CA

Emmanuel Hadzipetros has been an SAP contractor and consultant for more than 11 years, since R3 version 2.2. He has intimate and detailed knowledge of the IDoc interface and other standard SAP and external integration tools and technologies and has used this knowledge in more than 10 SAP implementations representing a wide variety of industries in four countries and three continents.

For the last four years Emmanuel has gone Hollywood as a key fixture on two major Studio implementations where he helped lead the SAP development effort in building complex VMI-based EDI transactional processing systems supporting billions of dollars a year in Video and DVD sales.

Emmanuel currently lives with his son Johnny in Westlake Village, California. He is SAP EDI Development Lead at NBC Universal Home Entertainment, where he is actively involved in all phases of designing and implementing a dynamic EDI architecture that includes Gentran Integration Suite for EDI services, Contivo Analyst for mapping, XI for integration logic, and SAP R3 as the enterprise system of record.