Doktorat der Sozial- und Wirtschaftswissenschaften

1. Beurteilerin/1. Beurteiler: a.o. Prof. Dr. Alexander Prosser

2. Beurteilerin/2. Beurteiler: o. Prof. Dr. Alfred Taudes

Eingereicht am: ————————————————————
Thema der Dissertation:

Assessing the Impact of XML/EDI with Real Option Valuation

Dissertation zur Erlangung des akademischen Grades
einer Doktorin/eines Doktors
der Sozial- und Wirtschaftswissenschaften an der Wirtschaftsuniversität Wien
ingereicht bei
1. Beurteilerin/1. Beurteiler: a.o. Univ. Prof. Dr. Alexander Prosser
2. Beurteilerin/2. Beurteiler: o. Univ. Prof. Dr. Alfred Taudes
von Mag. Shermin Voshmgir
Fachgebiet: Allgemeine Betriebswirtschaftslehre
Ich versichere:

1. dass ich die Dissertation selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

2. dass ich diese Dissertation bisher weder im In- noch im Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

3. das dieses Exemplar mit der beurteilten Arbeit übereinstimmt.

__________________________    ______________________________
Datum                              Unterschrift

Wien, im August 2001
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. Introduction</td>
</tr>
<tr>
<td>0.1. General Background of Study</td>
</tr>
<tr>
<td>0.2. Motivation for Research</td>
</tr>
<tr>
<td>0.3. Research Objectives and Target Audience</td>
</tr>
<tr>
<td>0.4. Thesis Outline and Research Methodology</td>
</tr>
<tr>
<td>Part One</td>
</tr>
<tr>
<td>1. Electronic Markets</td>
</tr>
<tr>
<td>1.1. Electronic Markets vs. Traditional Markets</td>
</tr>
<tr>
<td>1.2. How do Electronic Markets affect traditional Market Structures?</td>
</tr>
<tr>
<td>1.2.1 Information Phase</td>
</tr>
<tr>
<td>1.2.2 Agreement Phase</td>
</tr>
<tr>
<td>1.2.3 Settlement Phase</td>
</tr>
<tr>
<td>1.3. Electronic Market Theory and Reality</td>
</tr>
<tr>
<td>1.3.1 Information Phase</td>
</tr>
<tr>
<td>1.3.2 Agreement Phase</td>
</tr>
<tr>
<td>1.3.3 Settlement Phase</td>
</tr>
<tr>
<td>1.4. The next step – XML and the Semantic Web</td>
</tr>
<tr>
<td>2. The Semantic Web</td>
</tr>
<tr>
<td>2.1. The idea of XML</td>
</tr>
<tr>
<td>2.1.1 XML Syntax</td>
</tr>
<tr>
<td>2.1.2 XML Structure</td>
</tr>
<tr>
<td>2.1.3 XML Style</td>
</tr>
<tr>
<td>2.1.4 XML Navigation</td>
</tr>
<tr>
<td>2.2. Benefits of XML</td>
</tr>
<tr>
<td>2.2.1 Better Document Management</td>
</tr>
<tr>
<td>2.2.2 Better Searching Capabilities</td>
</tr>
<tr>
<td>2.2.3 Multiple uses for Hyperlinking</td>
</tr>
<tr>
<td>2.2.4 Business Impact of XML</td>
</tr>
<tr>
<td>2.2.5 Increase in the Number of XML Applications</td>
</tr>
<tr>
<td>2.3. XML Applications for E-commerce</td>
</tr>
<tr>
<td>2.4. Summary and Outlook</td>
</tr>
<tr>
<td>3. EDI</td>
</tr>
<tr>
<td>3.1. How it works - some general</td>
</tr>
<tr>
<td>3.1.1 EDI Message Standards</td>
</tr>
<tr>
<td>3.1.2 Components of Message Standards</td>
</tr>
<tr>
<td>3.1.3 Converter and Communication Software</td>
</tr>
<tr>
<td>3.1.4 EDI Networks</td>
</tr>
<tr>
<td>3.1.5 Cost of setting up and EDI network</td>
</tr>
</tbody>
</table>
3.2. Opportunities through EDI ................................................................. 40
3.2.1 Operational Benefits - General Cost reduction ............................. 41
3.2.2 Strategic benefits ................................................................. 41

3.3. Barriers for EDI penetration .......................................................... 42
3.3.1 Limited number of standard EDI documents defined ................. 42
3.3.2 Current EDI usage ................................................................. 43
3.3.3 Legal Problems .................................................................. 43
3.3.4 Conclusion ...................................................................... 43

3.4. EDI an the Internet ................................................................. 44
3.4.1 Internet/EDI .................................................................. 45
3.4.2 Web/EDI ...................................................................... 45
3.4.3 XML/EDI ...................................................................... 46

3.5. Summary and Outlook ................................................................. 47

4. XML/EDI .................................................................................. 48

4.1. The Idea of XML/EDI ................................................................. 49
4.1.1 Fusion of Five .................................................................. 49
4.1.2 XML/EDI Core Models .......................................................... 51
4.1.3 XML/EDI as a Framework ..................................................... 52
4.1.4 Progress in XML/EDI Standards and Frameworks ............. 54

4.2. Opportunities through XML/EDI ............................................. 55
4.2.1 Impact of XML on EDI ............................................................. 56
4.2.2 Reduction of General Costs ..................................................... 56
4.2.3 Hub and Spoke Relationship Improvement ............................. 57
4.2.4 Consequences for Business ............................................... 58

4.3. Challenges of XML/EDI ................................................................. 59

4.4. XML/EDI Example .................................................................. 60

4.5. Summary and Outlook ................................................................. 64

Part Two .................................................................................. 66

5. Assessing the Value of IT Projects ............................................. 66

5.1. The IT Productivity Paradox ..................................................... 67

5.2. Traditional Methods – Investment Under Certainty ..................... 69
5.2.1 Static Methods .................................................................. 70
5.2.2 Dynamic .......................................................................... 70

5.3. Investment under Uncertainty ..................................................... 71
5.3.1 Decision Tree Analysis .......................................................... 71
5.3.2 Real Option Methods ............................................................. 72

5.4. Empirical Studies of Accounting Methods used ......................... 72

5.5. Option Value in IT – Real Options .......................................... 74
5.5.1 Real Options – Some General Facts ......................................... 74
5.5.2 The Formula .................................................................... 75
5.5.3 Simplifying the calculation....................................................... 77
5.5.4 Capturing Flexibility ............................................................. 80

5.6. Practical Application of Real Options ....................................... 81
5.6.1 Disadvantages of Real Option Models .................................... 82
5.6.2 Comparison of Real Options with Net Present Value ....................................................... 82
5.6.3 When to use Real Option Pricing Techniques ................................................................. 84
5.6.4 Are Option Valuation Methods in Use? ........................................................................... 85

5.7. Summary and Outlook ........................................................................................................ 86

6. XML Investment Decision as a Deferral Option ................................................................... 87
6.1. General Assumptions .......................................................................................................... 88
6.2. Going into Details ............................................................................................................... 90
6.2.1 Project Uncertainty ($\sigma^2$) ......................................................................................... 90
6.2.2 Project Returns ($S$) ...................................................................................................... 95
6.2.3 Project Costs ($X$) ......................................................................................................... 97
6.2.4 Time Value of Money .................................................................................................... 98
6.2.5 Time to expiration ($t$) .................................................................................................. 99
6.2.6 Summary ....................................................................................................................... 99
6.3. Company Profiles ............................................................................................................. 99
6.3.1 Determining Diffusion ................................................................................................... 99
6.3.2 Type of Adopters ......................................................................................................... 100
6.3.3 Parameters that Influence Diffusion ............................................................................. 101
6.3.4 Market Power ............................................................................................................... 103
6.4. Putting it all together ....................................................................................................... 106
6.5. Criticism .......................................................................................................................... 109

7. Conclusion .............................................................................................................................. 109

Appendix 1: XML Tutorial ........................................................................................................ 112

Appendix 2: XML/EDI Working Groups .................................................................................. 146

Appendix 3: XML Business Applications .............................................................................. 149

Appendix 4: Black-Scholes Option Pricing Table .................................................................... 152

References ............................................................................................................................... 153
List of Figures

Figure 1: Phases of a Market Transaction .................................................................6
Figure 2: The Business Media Framework ..................................................................7
Figure 3: Technologies Supporting the Phases of an Electronic Market ......................10
Figure 4: Historic Overview of SGML, HTML, XML, and XHTML ............................15
Figure 5: How XML works .......................................................................................18
Figure 6: The Complexity of Converting Data to In-house Standards .......................34
Figure 7: The Basic Functions of EDI Software .......................................................36
Figure 8: XML/EDI the Fusion of Technologies .....................................................50
Figure 9: XML/EDI Transaction Models ...................................................................52
Figure 10: XML/EDI Layered Architecture .............................................................53
Figure 11: Plain purchase order ...............................................................................61
Figure 12: Fragment of ANSI X12 transaction set corresponding to Figure 11 ............61
Figure 13: XML document analogue to the X12 transaction set in Figure 12 ...............62
Figure 14: XML/EDI Transaction Schematics Example ..........................................64
Figure 15: Rating of Usage and Importance of Evaluation Techniques .....................73
Figure 16: Mapping the Option Parameters on an IT Investments Decision ...............76
Figure 17: Mapping the Option Parameters on XM/EDI Investments Decision ..........89
Figure 18: Company Profiles in XML/EDI Diffusion Matrix ....................................106
Figure 19: Mapping the Black Scholes Parameter on the Company Profiles .............108
Figure 20: XML/EDI Historic Highlights .................................................................146
0. Introduction

Companies today are trying hard to find a way to link their internal business processes with those of trading partners without plunging themselves into a maze of technical headaches and integration costs. No robust standards have yet emerged to support interoperable, cross-platform e-business. The typical e-business software products today cover topics as customer relationship management (CRM), electronic data interchange (EDI), enterprise application integration and enterprise resource planning (ERP), but the industry is nowhere near delivering anything resembling plug-and-play technical integration at the Business-to-Business (B2B) level.

Yet the basics of e-business are quite simple. They revolve around two familiar technologies that companies have known for years: EDI and workflow, where EDI stands for structured document routing between companies, and workflow for structured document routing within each organization. But, EDI and workflow have been separated by their divergent spheres of application.

The virtual business is constantly repositioning itself through dynamic alliances with an ever-changing set of trading partners. Therefore, an Internet-based trading community can only be reality where all participants share a common process model, an agreed-upon set of business rules that govern structured transactions. Just as important, all participants will have to implement a common B2B interoperability framework that allows them to enter easily into new trading relationships without the need for costly, time-consuming, (bilateral) or multilateral coordination of legal agreements and technical interfaces, as is the case with classic EDI.

Traditionally, EDI has been effective for communicating transactions between companies, but only large enterprises performing a high volume of transactions have realized significant benefits using it. Now the XML standard promises an open standard for cost-effective B2B Internet commerce that will enable midsize and smaller companies to use EDI.

But it is hard to predict future developments of XML in general, and the integration of XML with EDI in particular, as regards to how far it will be accepted as standard,
what impact it will have on the way of exchanging business, and when it will reach a critical mass of companies deploying it. Nevertheless, XML's pledge of better data portability among trading partners is the wave of the future.

0.1. General Background of Study

The costs for participating in an electronic marketplace are lowered by emerging public and private standards for access and information exchange over the Internet, such as the HTTP protocol, technologies for secure information exchange like SSL and standards for electronic commerce. While EDI and the Internet have very different origins, they are the most ubiquitous infrastructures for electronic markets, and recently the boundary between them seem to be blurring.

The idea behind EDI is a good one, unfortunately it has not worked out so well in reality. After almost 30 years, only 2% - 5% of the companies today in Europe and the US make use of EDI (Segev et al. 1997). One of the major problems with the current implementations of EDI is that they often require a unique solution for each pair of trading partners, making EDI costly and time-consuming to implement. Classic EDI Solutions are based on established EDI standards, like EDIFACT or ANSI X.12 that need exact coordination of message structures between communications partners. This time consuming coordination as well as the development of message converters are some of the reasons why classic EDI solution have only been used for long-term and intensive business relations.

With XML there are now moves to enable EDI data to travel inside XML, which means to make EDI portable for the Internet. The vision of XML/EDI is to allow organizations to deploy smarter, cheaper and more maintainable systems to a global audience. Thus, XML/EDI shall be equally accessible to small business as to large corporations.

The question for a company today is not whether to invest in XML, but rather when and to what extent. The problem that companies are facing is that they do not know how and when to invest into XML. “Will our competitors, clients or suppliers use XML? If yes when and how? Shall we invests now, or shall we wait and see?”

These questions are hard to answer, as there is a big uncertainty in the diffusion of XML/EDI. For a proactive company who wants to be market leader, the “wait and
see" strategy might not be the option. As the XML/EDI investment decision is a highly strategic one, and if managers want to avoid making decisions intuitively, the only way to answer this question is try and assess the necessary IT investment.

0.2. Motivation for Research

Companies that trade electronically with their business partners are said to enjoy strategic benefits such as a higher responsiveness to customers resulting from the higher speed of internal processing, and an optimization of inter-organizational business processes through electronic integration with customers and suppliers. Many existing studies in this field emphasize that a mayor obstacle to evaluate IT investment is the inability to measure productivity gains.

My dissertation derives its motivation from the question of how a company today can decide whether and when to invest into XML/EDI, as it is too early to comment about the value of XML/EDI and other XML vocabularies and frameworks like ebXML. We are currently between the development and implementation stage of the extensible markup language, thus the value that XML proponents have claimed can only be proven by time.

The aim of this dissertation is to review available capital budgeting techniques and identify the most appropriate set of measures to demonstrate the business value of XML/EDI. This dissertation aims to contribute to research with an investment valuation framework based on the theory of real option pricing.

0.3. Research Objectives and Target Audience

The overall objective of the research is to contribute to the theoretical understanding of the XML standard and its applications, in particular XML/EDI, as well as to develop a framework for the assessment of the value of an XML/EDI investment project. As this framework is based on the complex Black-Scholes option pricing formula, my particular aim is to make this framework as easy to understand and easy to as apply as possible.

The target audiences of the thesis are academic scholars as well as business practitioners. The technical introduction to XML and its applications, as well as methods of IT assessment in general, addresses academics as well as business
practitioners. The practical conclusions of the valuation of XML/EDI are primarily directed towards management of any scale of business.

0.4. Thesis Outline and Research Methodology

This thesis is arranged in two parts. In the first part, XML, EDI and XML/EDI, will be analyzed regarding their underlying technologies, as well as their advantages and disadvantages of supporting e-business transaction in an electronic market.

In the second part, IT-investment assessment methods will be analyzed with the conclusion, that in the case of the XML/EDI investment decision, real options pricing theory is the appropriate capital budgeting technique to be used.

Based on this perception the Black-Scholes option pricing formula will be broken down into its single parameters for the XML/EDI investment decision. Analyzing the underlying fundamentals and theories of real option pricing and based on the theory of technology diffusion, conclusions will be derived for creating a framework for facilitating the XML/EDI investment decision.
Part One

Analyzing the phases of an electronic market, it will be derived in the first chapter that current technologies only support certain electronic markets transactions, but not continuously and seamlessly. Chapter two explains the idea of XML and analyzes the benefits and applications for e-commerce. The third chapter will discuss the EDI standards, its advantages and fall-downs, and the perspectives of integrating it with the Internet. The particular interest of the fourth chapter will be to analyze possibilities to combine the idea of EDI with the flexibility of XML and the Internet by defining XML/EDI standards.

1. Electronic Markets

Buyers often face substantial search costs to obtain information about price and product offerings. These costs enable sellers to extract monopolistic rents in otherwise competitive markets and create inefficiencies in the allocation of economic resources (Bakos 1991). Due to the recent progress in information technology and the emergence of the Internet in particular the structure of traditional markets have changed and electronic markets have surfaced that increase the efficiency and effectiveness of traditional market processes (Bakos 1998).

In this chapter the term electronic market will be defined, discussing the characteristics and functions of electronic markets. The aim is to analyze how and why electronic market theory differs from what is possible in reality. It will be derived that current technologies only support single phases of a business transactions, but not seamlessly and not for all market players.

1.1. Electronic Markets vs. Traditional Markets

Markets have the economic function of facilitating the exchange of information, goods, services and payments. Thus they create an economic value for buyers, sellers, market intermediaries and society as a whole. Traditional market theory defines a market as a place of exchange, where aggregate-demand meets
aggregate supply, by allocating resources, skills and products. Coordination is the key factor for optimizing allocation and to yield maximum economic welfare.

Schmidt (1993) defines 3 phases that are substantial for any process of market coordination, thus also for electronic markets: (1) **Information Phase**: information on available products and services, their specifications, and delivery terms. (2) **Agreement Phase**: after evaluating all information, contacting potential transaction partners and negotiation of terms and conditions, as well as establishment of a legal basis. (3) **Settlement Phase**: physical exchange of goods and/or services and flow of financial transactions and information.

![Figure 1: Phases of a Market Transaction (Schmidt 1993)](image)

According to Bakos (1991) "an electronic market place is an inter-organizational information system that allows the participating buyers and sellers to exchange information about prices and product offerings [...] Information systems can serve as intermediaries between buyers and sellers in a vertical market, thus creating an electronic marketplace."

In the narrow sense, Schmidt 1993 defines electronic market as a market put into action by telematics, supporting all phases of transactions, including the formation of prices for goods and services. He characterizes electronic markets by three features: (1) ubiquity, i.e. 24-hour access, (2) easy access of information, and (3) low transaction costs. In and broader sense he defines Electronic Markets as
information systems supporting one or more phases and functions of coordination within market systems.

Zimmermann (1998) splits the three basic phases of a market transaction, in four different views: The community view, process view, transaction view, and infrastructure view (see Figure 2).

![Figure 2: The Business Media Framework (Zimmermann 1998)](image)

The community view defines the relevant business community and its underlying, basic business model. Electronic Markets constitute the space for an exchange of goods within the community. The community view defines the basic business-, organization-, and communication models of a specific business community.

The process view defines the inter-organizational business procedures and processes along the value chain of the respective business community. These processes enable the business community to realize their defined business models.

---

1 Zimmermann (1998) defines Business Communities as the quantity of agents (e.g., people, organizations, computers), whose rights and duties of agents are defined through the assignment of roles.
At the transaction view level generic services have to be defined that enable the realization of the defined business processes. The offered services support the different functions in the three distinguished phases. Examples are intelligent electronic product catalogs including search functions, services supporting the negotiation of contracts, logistic and payment systems to settle the transaction. The overall goal is to enable a continuous settlement of business transactions without incompatible interfaces and to provide an infrastructure of services that is available to different business communities.

The infrastructure view defines the necessary network services and other technical components to realize the settlement of the transaction from a technical point of view. Examples are message handling-, EDI Networks, and clearing center services.

1.2. How do Electronic Markets affect traditional Market Structures?

The Internet affects markets by changing the structure of product offerings, as for example: (i) Aggregation and disaggregation of information based product components. With electronic markets in place the role of intermediaries will be reduced or even eliminated, leading to disintermediation. It will become easier to match buyers and sellers, reducing the costs or market transactions. (ii) The costs of logistics are decreased when electronic marketplaces improve information sharing between buyers and sellers by promoting quick, just-in-time deliveries and reduce inventories. (iii) Increased personalization of product offerings.

1.2.1 Information Phase

Electronic marketplaces reduce the costs of obtaining (communicating) information about price and product features, thus increasing efficiency of the market. Lower search costs enable new markets to emerge, as for example creating a market for second hand cameras where otherwise the search costs would be too high to enable potential buyers and sellers to find each other on a conventional market (Bakos 1998).

The ability of Internet marketplaces to reduce search costs for price and product information may significantly affect competition. Lower buyer search costs in electronic marketplaces promote price competition between sellers (Bakos 1997).
The dynamics of friction free markets will change product-pricing strategies of companies that had previously depended on geography or customer ignorance. As geography becomes less Important, new source of product differentiation such as customized features, service or innovation will become more important (Bakos 1998).

1.2.2 Agreement Phase

New types of price agreements are possible in electronic markets with auctions, or bulk buying, where customers can obtain higher bargaining power.

The ability to customize products, combined with the ability of sellers to access substantial information about prospective buyers, such as demographics, preferences and past shopping behavior, is greatly improving seller’ ability to price discriminate. Price discrimination is a powerful tool that allows sellers to increase their profits, and reduces consumer surplus enjoyed by buyers.

On the other hand, price discrimination enables sellers to service buyers that would otherwise be priced out of the market, an outcome that increases economic efficiency (Bakos 1998).

1.2.3 Settlement Phase

With digital information goods such as news, music or software, the Internet dramatically reduces the marginal cost of distribution of these goods (Bakos 1998).

Furthermore, electronic payment systems have the potential of lowering transaction costs of commercial exchanges, and micro payment systems will lower the costs of small transactions, enabling new pricing strategies such as metering of software use.

1.3. Electronic Market Theory and Reality

It as been discussed in the previous section that markets are increasingly carried out electronically. These transactions, leaving EDI aside, are mainly carried out on the Internet. Still, the volume of electronic transactions has not yet met industry expectations in absolute terms; why is that so? To answer this question it will be
necessary to analyze how well the phases of a market transaction are technically supported in reality?

![Figure 3: Technologies Supporting the Phases of an Electronic Market](image)

Still, the volume of electronic transactions has not yet met industry expectations in absolute terms; why is that so? To answer this question it will be necessary to analyze how well the phases of a market transaction are technically supported in reality? Figure 3 shows which technologies, based on Zimmermann’s (1998) transaction view, currently support the phases of a transaction, in other words, e-business.

### 1.3.1 Information Phase

Today’s web is created by Hand for Eyes only. The *Information IQ* of HTML (Hypertext Markup Language), the most frequently used document type in the Web, is too low to enable many desirable e-commerce applications. HTML was designed as a markup language with simple structures, strong emphasis on formatting, but weak for encoding content. It was not designed to encode structure and semantics needed for complex applications as is needed for e-business. It defines a single, fixed type of document with markup that describes a common
class of simple office style report, with headings, paragraphs, lists, illustrations, etc., and some provision for hypertext and multimedia. HTML was defined to allow the transfer, display and linking of documents over the Internet and is the key enabling technology for the WWW. Prior to the emerging of the Internet, it was unusual in the word of computing to hear the word page used to describe elements of data. But HTML web pages have amazing similarities with paper in their role of information publishing. Both HTML and paper pages:

- Are optimized for visual clarity.
- Focus on ultimate usability (but not on reusability).
- Contain no contextual information.
- Have no document structure to enable automation.

Search engines use software robots to survey the Web and build their databases by retrieving and indexing HTML documents. When a query is entered at a search engine website, input is checked against the search engine's keyword indices. The best matches are then returned as hits. There are two primary methods of text searching: **Keyword Searching** and **Concept-based Searching**.

(1) **Keyword Searching** is the most common form of text search on the Web. Most search engines do their text query and retrieval using keywords or meta-tags. Essentially, this means that search engines pull out and index words that are believed to be significant. Words that are mentioned towards the top of a document and words that are repeated several times throughout the document are more likely to be deemed important. Most search engines index every word on every page. Others index only part of the document, such as the title, headings, subheadings, hyperlinks to other sites, and the first 20 lines of text. Some of the search engines discriminate upper case from lower case; others store all words without reference to capitalization. There are many problems with keyword searching, i.e.: Keyword searches cannot distinguish between words that are spelled the same way but mean something different. This often results in hits that are completely irrelevant to a query. Search engines also cannot return hits on keywords that mean the same, but are not actually entered in your query.
(2) Concept-based searching, unlike keyword search systems, tries to determine what a user means. In the best circumstances, a concept-based search returns hits on documents on a specific topic, even if the words in the document do not precisely match the words a user enters into the query. This is also known as clustering, which essentially means that words are examined in relation to other words found nearby. There are various methods of building clustering systems, some of which are highly complex, relying on sophisticated linguistic and artificial intelligence methods. Concept-based indexing is a good idea, but it is far from perfect. The results are best when one enters a lot of words, all of which roughly refer to the concept one seeking information about.

In general, finding information on the web, based on HTML pages is a difficult task. One needs to know the retrieval mechanisms, which vary significantly from search engines to search engine, and often complex and time consuming refining search is needed.

Electronic Product Catalogues (EPC) provide firms an opportunity to showcase their products (characteristics, availability, price, etc.) on the Internet. EPC offer technical solutions, which facilitate a series of transactions between a client and a seller such as request for information, and in certain cases EPC can help customize product offerings to customers’ demand. Once a user has accessed an EPC site, the information is structured more intelligently as opposed to flat HTML pages, as the data is stored in a back-end database. The problem is that the information in the font-end is represented in HTML format, which was the problem in the first place.

Search Engine queries, EPC, and Email communication can be used for pull-based information purposes. Newsletters, on the other hand, serve for informing about new products and services as a push-based technology. The Internet reduces transaction costs of the information phase, but not yet to its full potential. All these technologies lack the structure and semantics of fully automated electronic markets.
1.3.2 Agreement Phase

Electronic contracting may be viewed from the perspective of its openness to new contracting relationships: (i) open electronic contracting, and (ii) closed electronic contracting (Lee 1998b).

Classic EDI uses closed electronic contracting to advance contracting among parties that already have a trading relationship established. The focus of such applications is to make customer-vendor trading more efficient. The more challenging case for electronic contracting is open electronic contracting, which allows the formation of contracts among parties with no prior trading relationships. Lee (1998b) points out that the type of electronic contracting (closed to open) is actually a continuum, allowing for intermediate cases. A central issue for open electronic contracting is in representing and negotiating ways of doing business among parties. The difficulty with classic EDI is that it does not allow flexible adaptation and customization to meet the needs of a particular situation.

Another way to negotiate is via Email. Electronic negotiation through Email can be an information rich medium through which people can negotiate not just only price but also other aspects such as subscription terms. Electronic negotiation can be further automated with Intelligent Negotiation Support Systems, which understand Email messages and negotiate with the customer using case based reasoning. The main problem of email negotiation is the integrity of a contract due to the lack of trust between two relevant parties, and the lack of structure of the negotiation process (Lee 1998a).

Electronic Auctions are special cases for automated negotiations. Auctions are consolidated market institutions with formal process rules for market access, trade interaction, price determination and trade generation, which are applied for the exchange of numerous kinds of goods ranging from commodities like raw materials to individual objects like fine art (Klein 1997).

1.3.3 Settlement Phase

For more than 25 years, companies have been using EDI to transmit structured business documents like orders or invoices electronically. As opposed to paper-based communication, EDI is designed to make communication between different
systems possible without media discontinuities. As Emmelhainz (1990) points out EDI works well for highly standardized transactions between businesses that have existing relationships. But although there is an undoubtedly large savings potential, the use of EDI is by far not as widespread as one could expect. The main reason is that especially small and medium-sized enterprises try to avoid the considerable setup- and operating costs of traditional EDI solutions and that the standards are not flexible enough. In addition, the variety of different industry-specific and national EDI standards makes it difficult for businesses to decide which standard to use. Therefore, the use of EDI is mainly reserved for large companies, and, as empirical analysis shows, one of the main reasons for SMEs to introduce EDI is usually pressure from larger business partners (Weitzel et al. 1999).

The Infrastructure underlying the public Internet allows to distribute electronic goods over the Web, thereby dramatically reducing the marginal cost of distribution. But the problems with the distribution of electronic goods lie in the settlement process: (i) rights management, (ii) payment, (iii) and flexible digital contracts. For big companies that use EDI, contracts including rights management issues are settled in advance, payment and other document handling transaction can be handled with EDI solutions. But for the other 95% of companies these issues still need to be solved.

1.4. The next step – XML and the Semantic Web

Today, many automated Web-based services already exist without semantics, but other programs such as agents have no way to locate one that will perform a specific function. HTML was the enabling technology of the Internet, but it lacks intelligence of information thus posing a significant barrier to reuse, interchange, and automation needed for e-commerce applications. EDI, the grandfather of electronic commerce, has been very successful in the past, but due to its inflexibility and because it is too expensive for small and medium sized companies, it is not in widespread use.
For this reason, XML (Extensible Markup Language) was developed by the XML working group (known as the SGML¹ Editorial Review Board) formed under the auspices of the World Wide Web Consortium (W3C) in 1996. XML is a highly functional subset of SGML. The purpose of XML is to specify an SGML subset that works very well for delivering SGML information over the Web. It is actually misnamed because XML is not a single Markup Language. It is a meta-language to let users design their own markup language.

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1986</td>
<td>SGML ISO 8879-1986</td>
</tr>
<tr>
<td>Nov 1995</td>
<td>HTML 2.0</td>
</tr>
<tr>
<td>Nov 1996</td>
<td>Simplified and stripped down SGML draft (dubbed XML)</td>
</tr>
<tr>
<td>Jan 1997</td>
<td>HTML 3.2</td>
</tr>
<tr>
<td>Aug 1997</td>
<td>XML working draft</td>
</tr>
<tr>
<td>Dec 1997</td>
<td>XML 1.0 proposed recommendation HTML 4.0 Recommendation</td>
</tr>
<tr>
<td>Jan 1998</td>
<td>XML</td>
</tr>
<tr>
<td>Feb 1999</td>
<td>XHTML</td>
</tr>
</tbody>
</table>

Figure 4: Historic Overview of SGML, HTML, XML, and XHTML

XML is a public format and not a proprietary format of any company. The v 1.0 specifications were accepted by the W3C as Recommendation on February 10, 1998. While retaining the beneficial features of SGML, XML removes many of the more complex features of SGML that make the authoring and design of suitable software both difficult and costly. But XML also lacks some important capabilities of SGML that primarily affect document creation, not document delivery. That is so

---

¹ In 1986 the Standard Generalized Markup Language (SGML) became an international standard for defining descriptions of the structure and content of different types of electronic documents. SGML, the "mother tongue" of HTML and XML, is used for describing thousands of different document types in many fields of human activity, from transcription of ancient Sumerian tablets to the technical documentation for steal bombers, and from patient's clinical records to musical notations. SGML has withstood the test of time. Its popularity is rapidly increasing among organizations with large amounts of document data to create, manage, and distribute as in the Defense, Aerospace, Semiconductor and Publishing industries. However, various barriers exist to delivering SGML over the Web. These barriers include the lack of widely supported style sheets, complex and unstable software because of SGML’s broad and powerful options, and obstacles to interchange of SGML data because of varying levels of SGML compliance among SGML software packages. These difficulties have condemned SGML to being a successful niche technique rather than a mainstream tool. Indeed some cynics have renamed SGML in 'Sounds Good Maybe Later'.

as XML was not designed to replace SGML in every respect. The idea of XML is
to support the vision of the *Semantic Web*, to bring structure to the meaningful
content of Web pages, creating an environment where software agents roaming
from page to page can readily carry out sophisticated tasks for users.

2. The Semantic Web

It has been discussed that most of the content on the Internet is designed for
humans to read, not for computer programs to manipulate meaningfully.
Computers can parse Web pages for layout and routine processing (headers,
links, meta-tags), but in general computers have no reliable way to process the
semantics. How does a tourist on the first day in a new city know where to go for
dinner? He can ask for recommendation from locals, he may have a list of
recommended restaurants in his tourist guidebook, or he might be even trying to
find some information on the Internet. From personal experience most people
know how pain seeking and time consuming this process can be.

The challenge of the semantic web is to provide a language that expresses both
data and rules for reasoning about the data and that allows rules from any existing
knowledge-representation system to be exported onto the Web. In the world of the
semantic web a tourist can connect to the Internet via his handheld device and
enter certain keywords when looking for a restaurant. A software agent searching
the web will find the information of a list of restaurant containing keywords such as
<dinner>, <lunch>, <restaurant>, <Italian> (as might be encoded today) but also
that the <opening-hours> at this restaurant are <weekdays> and then the script
takes a <time range> in <yyyy-mm-dd-hour> format and returns <table available>,
thus automatically booking a table when the tourist enters <ok>.

For the semantic web to function, computers must have access to structured
collections of information and sets of inference rules that they can use to conduct
automated reasoning. Knowledge representation, as this technology is often
called, is clearly a good idea, and some very nice demonstrations exist, but it has
not yet changed the world. It contains the seeds of important applications, but to
realize its full potential it must be linked into a single global system. Traditional
knowledge-representation systems typically have been centralized, requiring
everyone to share exactly the same definition of common concepts, but central
control is stifling, and increasing the size and scope of such a system rapidly becomes unmanageable.

Three important technologies for developing the Semantic Web are already in place: XML, the Resource Description Framework (RDF)\(^1\) and Ontologies\(^2\). XML allows users to add arbitrary structure to their documents but says nothing about what the structures mean. This is what RDF (an XML application itself) is used for, expressing meaning. The next challenge for the realization of the semantic web is that two databases may use different identifiers for what is in fact the same concept, such as `<chair>`. A program that wants to compare or combine information across the two databases has to know whether these two terms are being used to mean the same thing, a chair to sit on or the chairman of a conference. Ideally, the program must have a way to discover such common meanings for whatever databases it encounters. A solution is provided by ontologies, the third basic component of the Semantic Web.

The real power of the Semantic Web will be realized when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs. The effectiveness of such software agents will increase exponentially as more machine-readable Web content and automated services (including other agents) become available (Berners-Lee et. al. 2001).

This chapter will concentrate on the basic component of the semantic web, XML: discussing the concept of XML; analyzing opportunities and challenges; and giving an overview of current XML applications, in particular e-commerce applications.

---

1 RDF (W3C 1999b) encodes XML in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence. These triples can be written using XML tags. In RDF, a document makes assertions that particular things (people, Web pages or whatever) have properties (such as "is a sister of," "is the author of") with certain values (another person, another Web page). This structure turns out to be a natural way to describe the vast majority of the data processed by machines. Subject and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a Web page. The verbs are also identified by URIs, which enables anyone to define a new concept, a new verb, just by defining a URI for it somewhere on the Web. The triples of RDF form webs of information about related things. Because RDF uses URIs to encode this information in a document, the URIs ensure that concepts are not just words in a document but are tied to a unique definition that everyone can find on the Web.

2 Ontologies are a document or file that formally define the relations among terms. They can be compared to a mediator between the information seeker and the set of XML documents. The most typical kind of ontology for the Web has taxonomy and a set of inference rules.
2.1. The idea of XML

The idea behind XML is to add meta-tags (information about information) to data being transferred over the Internet in a much more flexible and robust way than HTML. This should allow the sending system, in essence, to describe to the receiving system the nature of the data being sent. An XML document system usually consists of three components: Semantics (XML file), structure (i.e. DTD), and presentation (Stylesheet).

![Diagram of XML components]

First, there is the set of documents itself, the XML file. This document holds the meaning of the information, the content (semantics). Furthermore, the structure of the XML document (or a set of documents with the same structure) is defined by the Document Type Definition (DTD). DTDs specify how elements stand in relation to each other and can be very complex and large.

To convey an XML document to a processing system the marked-up document (XML file) as well as the DTD need to be passed on to the system so that the system knows how to interpret the tags of the elements. A software module called an XML processor is used to read XML files and provide access to their content and structure. It is assumed that an XML processor is doing its work on behalf of another module, the application. This specification describes the required behavior...
of an XML processor in terms of how it must read XML data and the information it must provide to the application.

In order to visually deliver the document to a human reader, it needs to be presented in some fashion. Thus, the system needs a set of rules specifying how to render each element: the style sheet.

2.1.1 XML Syntax

XML enables users to create own tags `<address>` or `<name>` that annotate Web pages or sections of text on a page. Scripts, or programs, can make use of these tags in sophisticated ways, but the scriptwriter has to know what the page writer uses each tag for.

Each XML document has both a logical and a physical structure. Physically, the document is composed of units called entities. An entity may refer to other entities to cause their inclusion in the document. A document begins in a "root" or document entity. Logically, the document is composed of declarations, elements, comments, character references, and processing instructions, all of which are indicated in the document by explicit markup.

The XML Tutorial in Appendix 1 will give a brief overview of the XML Syntax. For the complete specification of the XML Syntax see the website of the World Wide Web Consortium (W3C 2000a).

2.1.2 XML Structure

The logical structure of the document and tag set can be specified in a Document Type Definition (DTD). In a DTD, (i) sets of elements and their attributes are defined; (ii) the names that are used as tags are assigned; and (iii) the element relationships or transaction is defined.

If a DTD is used then programs can validate the transaction's structure. The best-known example of a DTD is HTML, which is defined by a SGML-DTD describing the structure of HTML documents.
The disadvantage of the DTD is the fact that it is not an XML document, thus Document Object Models\(^1\) (DOM) have no access to it. The W3C developed a new standard called the XML Schema (W3C 2001b) to overcome the problems of the XML DTD. XML Schemes have the same function as the DTD with the advantage that: (i) XML Schema is an XML Document. (ii) XML Schema support extended data types. (iii) XML Schema supports the concept of inheritance. (iv) Elements are extensible. (v) Possibilities of open definition.

The W3C Consortium envisions XML applications where a single XML document may contain markup vocabulary (elements and attributes) that are defined for and used by multiple software modules. One motivation for this is modularity; if such a markup vocabulary exists which is well-understood and for which there is useful software available, it is better to re-use this markup rather than re-invent it. But such documents, containing multiple markup vocabularies, pose problems of recognition and collision. Software modules need to be able to recognize the tags and attributes which they are designed to process, even in the face of collisions occurring when markup intended for some other software package uses the same element type or attribute name. XML Namespaces (W3C 1999a) avoid the collision of different semantics, when using different XML documents. XML namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by URI (Uniform Resource Identifiers) references.

2.1.3 XML Style

Frames and tables in HTML were the only tools to bring some layout into the web, which was quite frustrating, since this limited the control over how content appeared on Web pages. Furthermore there was always a problem of standardization between the web browsers. For the first time Cascading Style

\(\text{\textsuperscript{\textdagger}}\) DOM is a W3C Recommendation (W3C 1998), useful for scripting HTML and XML documents, including making changes to documents on the fly. The standard maintained by the W3C that specifies how the content, structure, and appearance of Web documents can be updated programmatically with scripts or other programs. The proposed object model for XML matches the Document Object Model for HTML so that scriptwriters can easily learn XML programming. The XML DOM will provide a simple means of reading and writing data to and from an XML tree structure.
Sheets (CSS) gave Web developers more control over style and layout. Style sheets work like templates where the style for a particular HTML element is defined once, and is then used over and over on any number of Web pages. In order to change how an element looks, the style sheet is modified and the element automatically changes wherever it appears. Before CSS, the only way to change the element was individually, each time it appeared. Style sheets let Web-designers quickly create more consistent pages, and more consistent sites.

Unlike HTML, XML has no implied presentation associated with any of the elements. XML merely provides the ability to format document content, whereas XSL (Extensible Stylesheet Language) provides the ability to define how the formatted XML content is presented.

XSL (W3C 2000b) offers more capabilities than the present CSS specification for HTML. As opposed to CSS, XSL can transform documents. XSLT, a subset of XSL, converts between different XML documents of converts into HTML CSS documents. XSL adds provisions for formatting of elements based on their position in the document, handling of generated text, and the definition of formatting macros. The new style language complements XML. It also introduces an extensible set of formatting objects and is the first Web style sheet language that accommodates languages that flow in different directions than right to left.

XSL lacks many features one might want to see in a print style sheet (e.g. multiple columns, widow & orphan control, etc.), but its rule-based approach to formatting on the fly is an exciting new development in Web publishing. It will enable an XSL-capable browser to do simple things like rearranging elements and reformatting them accordingly, something that previously could happen only on the server. Furthermore it will bring new capabilities, such as adjusting box rules to the dimensions of the type composed within the box, that are not possible in HTML today. Although it goes beyond CSS, XSL is being developed so that its style sheets can be easily translated to CSS for HTML documents.

For publishers who already use SGML, the rapid pace of XML-related development and the noticeably strong interest in it by the big industry players are welcome changes from the past decade, when SGML was relegated to a niche supported only by small software suppliers or by small pockets within larger
developers. Though the first XML implementations will likely focus on data processing, not document formatting, the introduction of a Web style sheet language for XML looks promising as the foundation for richer online style presentations in the future.

2.1.4 XML Navigation

XML Linking, formerly known as XLink and as XLL (the eXtensible Linking Language), is a work in progress of the Web Consortium. It is closely related to the XML Recommendation, adding functionality for high-function hypertext and hypermedia. It is now an independent Working Group, but its projects were started under the main XML Working Group, and progressed to solid Working Drafts there before being handed off to the new XML Linking Working Group.

The work of this working group has two parts: XLink provides advanced linking capabilities such as multidirectional and external linking, while the separate XPointer specifications provide a convenient and easily-understood way of describing locations in XML documents. Either can be used without the other, but they are most valuable in combination, and of course in combination with XML itself.

**XLink** (W3C 2001d) adds many kinds of advanced hypertext linking functionality to the Web (and other environments where it may be used). XLink provides the possibility of bi-directional links or links that lead to multiple destinations. Bi-directional links can traverse in either direction regardless of which way you went originally. HTML supports one-way links making the "Go Back" button on the browser the way to return to the location of the original link. A bi-directional link will allow a user to return to the original location via a corresponding link on the site. It will also make it easier to find dangling links when a document is moved or deleted. XLink furthermore enables to create links targeted to specific parts within a document. A specific part of information can be targeted rather than the entire document. HTML can do this as well but not with the expected ease of XML coding. Another advantage is the possibility of links having multiple functions. Hyperlinks can be established within a specific document or Web site to perform more than the one function that currently exists. For example, looking at a research document, one might find a descriptor hyperlinked to perform two
functions, one that request for a thesaurus definition and the other to reach other documents carrying the same term. The XML designer could designate different color links for each function, i.e. red for the definition and blue for the index. When the user placed a cursor over the link, the color would turn red to go to the definition or blue for the index. Last but not least, with XLink links will be able to execute special actions. Some of the proposed actions include hyperlinks that will replace the current document, open a link in a new window, and insert the contents of the link into the current document.

**XPointer** (W3C 2001c) provides better location specifications. XPointers are based on TEI extended pointer, a proven and multiple-implemented technology for the same purpose, developed by the Text Encoding Initiative. An XPointer is a sequence of "location terms" that, when interpreted left to right, specify a location in terms of the tree structure that XML markup expresses. It does not return whole documents; it is simply an abstract language that specifies locations. XLink can use XPointer as part of specifying where its link-ends reside. Furthermore, XPointer Links a point to specific places inside of documents, even when the author of those documents has not already provide an ID at just the right place. Also Fine-grained addressing to elements, other information objects, point and strings selections, and spans (also called ranges) inside documents is made possible. XPointer also provides a clear syntax for talking about locations and relationships in hierarchies, such as the structure of XML documents, so that locations are human-readable and writeable.

### 2.2. Benefits of XML

XML has some essential benefits. It is system independent, vendor independent, has metadata markup and is delivered via the Web. Furthermore, the advantage of XML is that it is not only a universal data format, but a set of standards has been developed, the XML family, that process XML data. As has been previously discussed, XML provides advanced maintainable linking, there are style sheets for the presentation of information, and many more.

As XML separates the content from the presentation, it is much easier for search engines, scripting tools and programming or scripting languages to work with the XML data. Presentation tags do not mess up the source-code. Document
structures can be matched with existing systems. Search queries can match the exact level in the XML document where the user needs to search for information. According to Webber (1998b: 22) XML:

- Define schema to suit individual information needs.
- Enables structured modeling with DTD (Document Type Definitions) and UML (Unified Modeling Language).
- Enforces error checking and validation.
- Explicitly states legal meaning, no ambiguity.
- Archives documents so they can be read in the far future.
- Allows multimedia, processes, and mixed data content
- XML replaces ASCII CSV for data interchange.
- XML provides the means for structured information exchange.

### 2.2.1 Better Document Management

More and more companies are moving toward using Intranets to help manage documents, document creation, and document distribution. Document formats such as HTML, PDF, word-processing systems, all create documents without any document management capability. Unless a user creates his own hierarchical structure, handling massive amounts of word documents is difficult at best. Finding information stored within these types of documents is even more frustrating. By using XML’s hierarchical document-handling capability, users can pick up and choose the exact data they want, without having to plow through all the rest. Furthermore XML allows storing documents in parts.

**HTML versus XML**

As already discussed, the creators of HTML intended it as a document description language, not a format for the graphics, sound, and video which have become major components of many Web sites today. These developments have resulted in an unintended lack of formatting control that often causes problems with the display of a document and control of the size of a browser window. In order to correct these problems, XML includes DTD and XSL components that make the structuring and formatting of documents more friendly. The problem of HTML is that IS departments must decide on standards: standards for creating files, standards for delivering information, and standards for storing, converting and searching for information on the company’s servers. As HTML offers no standards, these must be set by the developers themselves and therefore usually corresponds to what the designer wants, not what the company needs, whereas
XML requires for documents at least be well formed (Pitts-Moultis and Kirk, 1999: 98). With HTML, one actually has to create pieces of data, parse them down into workable size, and then think ahead to create hyperlinks that will allow users to navigate through the site to the information they need. XML on the other hand does this automatically. Its tree structure, almost like a table of contents, makes it easier for the user to select the data; it also makes it easier for computers to retrieve information. This makes XML a much more sophisticated document management system than HTML. With XSL one can create programming rules for how to pull information out of an XML document and transform it into another format, such as HTML.

**PDF versus XML**

PDF is an electronic file format which is entirely platform independent and which can be passed around from computer to computer. If users have the Acrobat Reader software or plug-ins installed on their browsers, they can read, search and print the file in the format in which the original designer created it. PDF is basically a display technology. It deals little with document format or structure. XML, on the other hand, is much more capable of handling and managing documents, because it is able to provide structured information that a wide variety of applications can understand. It furthermore allows the user to specify how the document is to be presented, whereas the layout and format of PDF documents is hard coded in the page. (Pitts-Moultis and Kirk, 1999: 99).

### 2.2.2 Better Searching Capabilities

Service discovery can happen only when there is a common language to describe a service in a way that lets other agents understand both the function offered and how to take advantage of it. In the current Web, services and agents can advertise their function by, for example, depositing such descriptions in directories analogous to the Yellow Pages. Many current search engines geared towards HTML markup and search queries offer this feature which focus on text in the URL, title, summary, or first heading, depending on the structure of the document.

The Semantic Web, in contrast, is more flexible. The consumer and producer agents can reach a shared understanding by exchanging ontologies, which provide the vocabulary needed for discussion. Semantics also makes it easier to
take advantage of a service that only partially matches a request. Thus, XML goes beyond the limitations of the fixed tagging structure of HTML and allows for the introduction of custom tags, resulting in enhanced searching on the Web. A search engine seeks the term in the tags, rather than the entire document, giving the user more precise results. Web authors, on the other hand, can approach content creation more strategically with XML than with HTML. When they create a document, they can use the XML metadata tag to think about how other people will consume and access that document.

In order to search XML Web pages, however, a search engine must support XML-based search queries. This will require search engine technology to build XML parsers and processors. The processor would need to recognize the structure of XML tags based on the filtering done by the parser. As XML applications develop for usage on the Web, it is expected that XML-based search engines will emerge.

2.2.3 Multiple uses for Hyperlinking

As previously discussed, Xlink provides the possibility of bi-directional links, making the "Go Back" button on the browser obsolete. It will also make it easier to find dangling links when a document is moved or deleted. Xlink furthermore enables to create links targeted to specific parts within a document. A specific part of information can be targeted rather than the entire document. HTML can do this as well but not with the expected ease of XML coding.

Another advantage is the possibility of links having multiple functions. Hyperlinks can be established within a specific document or Web site to perform more than the one function that currently exists.

Last but not least, with Xlink links will be able to execute special actions. Some of the proposed actions include hyperlinks that will replace the current document, open a link in a new window, and insert the contents of the link into the current document.

2.2.4 Business Impact of XML

Many voices in the industry say, that XML will revolutionize the exchange of business information similar to the way the phone, fax machine, and photocopier
did when those devices were invented. XML enables integration of legacy systems with Web enabled SMEs, and provides end-to-end document flow with smart process control. Centralized repositories of information, documents, rules, structures and formats, as well as loosely coupled distributed systems are other business impacts of XML (Webber 1998b: 29).

XML based protocols can interoperate, such as VoxML, the voice recognition protocol, with the banking industry's Open Financial Exchange protocol, which facilitates the exchange of financial data between financial institutions, businesses, and consumers. Banks, for example, can build personal financial management applications that can use voice recognition simply by adding a few lines of XML code, instead of incorporating an entire voice recognition application into the financial software.

With XML specific tags, search engines can give users more refined search results. A search engine seeks the term in the tags, rather than the entire document, giving the user more precise results. Web authors can approach content creation more strategically with XML than with HTML. When they create a document, they can use the XML metadata tag to think about how other people will consume and access that document.

### 2.2.5 Increase in the Number of XML Applications

In terms of a uniform document exchange format, XML is already under way. Microsoft is using XML as Exchange's native file format to speed access to and from the message store. Enterprise resource planning (ERP) software producers such as PeopleSoft Inc., Oracle Corp., and SAP AG already incorporate XML syntax into products to help companies lower the cost and labor involved in exchanging data with business partners. Third-party software vendors are pushing XML to enable data exchange over the Web between separate financial systems. But XML is more than just a uniform data exchange format.

With the flexibility of XML to create new vocabularies, which are actually the applications that use the XML language, industries can create their own elements, attributes and entities that fully explain the structure (with DTD and XML Schema) of the specific types of data they use. That means that XML allows anybody to
construct entirely new applications, not through conventional programming methods, but rather by defining the contextual structure of the document. Several XML-based standards are emerging and there are many ways to classify them. Here are some examples for XML vocabularies in use:

- **Extensible HTML (XHTML):** The WWW Consortium (W3C), seeking to expand the use of XML without making existing HTML elements obsolete, recommended the XHTML 1.0 specification as a bridge between the two language environments. The specification is designed to allow developers to create Web pages that combine the data structuring of XML and the presentation of HTML. XHTML 1.0 was created by rewriting HTML 4 as an XML application, creating a specification that will work with HTML browsers and leverage XML's device-independent access capabilities. Developers already writing HTML 4 documents should have a smooth transition to XHTML 1.0. XHTML will simplify the Web development process by obviating the need for developing multiple versions of a single Web site based on the type of device upon which the site will run. XHTML is the immediate link between the two standards, allowing developers to program their Web Sites without having to go through and strip out everything and reprogram what they already have in order to take advantage of XML. Pages developed in XHTML 1.0 instead of the current HTML 4.0 can be processed by standard XML tools without becoming completely useless to older HTML only technologies.

- **Resource Definition Format (RDF)** is the XML application for Metadata under development by the W3C to address areas such as better search engine capabilities and cataloging content. RDF will establish the syntax for how different tags relate to Web pages and to one another. Users or groups will be able to name the tags and decide the collections of tags to use. The Dublin Core, an RDF standard under development by the library community, consists of 15 tags that give basic information about electronic documents.

- **Meta Content Framework (MCF)** is Netscape's application for XML based on RDF for processing metadata. MCF will represent meta information about content that includes Web pages, desktop files, gopher and FTP files, e-mail, and structured databases.
- **Channel Definition Format (CDF)** is an XML-based file format for the description of channel information developed by Microsoft. Supported by Microsoft's Internet Explorer 4.0 and above, CDF is already in use by such sites as PointCast. The concept of Channels involves the organization and classification of information from Web servers and Web pages.

- **Chemical Markup Language (CML):** One of the earliest XML applications, CML is being developed to represent molecular science and related technical information on the Web, using a CML DTD and CML tags. Files would require a viewer such as CML Viewer to display. The sponsor is Open Molecule Foundation.

- **OFX and VOXML:** XML-based protocols can interoperate, such as VoxML, the voice recognition protocol, with the banking industry’s Open Financial Exchange protocol, which facilitates the exchange of financial data between financial institutions, businesses, and consumers. Banks, for example, can build personal financial management applications that can use voice recognition simply by adding a few lines of XML code, instead of incorporating an entire voice recognition application into the financial software.

- **Synchronized Multimedia Integration Language (SMIL):** is an XML-based application for the delivery of multimedia content on the Web. SMIL permits multimedia synchronization of the delivery of sound, video, graphics, and text, reducing the bandwidth.

- **Web Interface Definition Language (WIDL),** introduced in 1996, was one of the first applications of XML. WIDL provides a means of describing automated access to Web-enabled resources and enterprise applications through well-defined interfaces much like COM and CORBA. By abstracting information about Web and external application resources, applications can be protected from changes in the structure, appearance, and location of data.

- **XML/EDI:** The XML/EDI group has developed an XML standard for electronic commerce that would move it from the value-added network onto the Internet reaching a broader cross section of business (see chapter 4).
These are just some examples for XML applications. Many XML vocabularies are just for the purpose of enhancing XML, such as RDF, XML Schema, XPath, or Xlink, many of which have been discussed in previous sections. XHTML has publishing purposes. XML/EDI is a business data exchange. CML is a scientific vocabulary. The next section will analyze the development of XML applications for e-commerce, thus excluding applications designed for strictly enhancing XML, such as XSL even though they too may have use in business. The purpose of the next section is to show the extent of XML’s penetration into the world of business data exchange, which covers transactions between businesses and also with consumers, including scientific and technical.

2.3. XML Applications for E-commerce

In February 2000, XML.com published a survey that found 124 different XML business applications at various registries and reference sources. By August 2000, the number of XML business vocabularies listed with these sources appears to have doubled to over 250, putting even more pressure on standards bodies to develop a way for these applications to interact (Kotok 2000).

To help understand the various ways in which XML for business data exchange has grown, applications can be broken down into three major categories (See Appendix 3 for more detailed information on XML business applications):

(i) **XML Application Functions** set the guidelines for specific business operations that cut across industry boundaries.

(ii) **Vertical XML Applications** are one-industry applications, defining messages for exchanges within a specific industry. They represent the majority of the business vocabularies (146 of the 251).

(iii) **XML Application Frameworks** define the specifications for structuring XML messages between parties for exchanges both within and among industries. Frameworks represent the fewest number of vocabularies. They offer interoperability among the business languages and provide the most functionality, thus frameworks have received the most attention.

The question of whether XML would enter the market was answered when Microsoft, Adobe, Netscape and other big market players not only supported the
development of the new standard, but also began making sizable product investments to this new format. The momentum building behind the XML effort means that XML is inevitably destined to become the mainstream technology for powering broadly functional and highly valuable business applications on the Internet, intranets, and extranets. The problem of XML for e-business is the proliferation of so many business vocabularies, which have overlapping functions. Some consolidation among vocabularies has begun, but which standards will be used in the future is not yet clear. The question that is open is not whether XML will succeed as a widespread data format for e-commerce applications, but rather how fast, to what level of success and with what vocabularies.

2.4. Summary and Outlook

Open standards in general and XML in particular will play a major role in emerging electronic marketplaces. So far, there was no ubiquitous way of describing data, e.g. on websites or for heterogeneous applications, in a commonly understandable way. There was no universal data format. XML has the potential, especially in combination with Java, to be the data format of choice to enable the next step in the evolution of Electronic Data Interchange in particular, and e-commerce in general. XML is system-independent, vendor independent, proven with HTML on the Web. XML provides advanced maintainable linking. XML has metadata markup and is delivered via the Web. There are style sheets for views, transforms, and information presentation. XML enables integration of legacy systems with Web enabled SMEs, and provides end-to-end document flow with smart process control. Centralized repositories of information, documents, rules, structures and formats, as well as loosely coupled distributed systems are other business impacts of XML.

Together with the increasing importance of the Internet, the power and flexibility of XML, as well as its overwhelming industry support, or at least the strong pre-announcements, is good indications for widespread adoption of XML as a basis for integrating business partners, processes and applications. Eventually, XML’s ability to separate structure from content is expected to contribute to the
emergence of open markets with non-proprietary XML interfaces being the foundation of business-to-business communication.

But, in spite of the positive features and prospects of XML, two things must be kept in mind. It is not yet clear when and how and which XML business vocabularies will penetrate e-commerce applications. Furthermore, XML is solely a description language to specify the structure of documents and thus their syntactic dimension. Document structure can represent some semantic properties but it is not clear how this can be deployed outside of special purpose applications. DTD and document structure are not enough to give XML a sound semantics. Therefore it will be necessary to appoint true semantics to XML documents with RDF and ontologies.

3. EDI

Companies today have put great effort into constructing computer applications to help them in their business processes. While this has resulted in significant improvements in efficiency, that efficiency has not been extended to external processes. External processes are processes that involve interchange between applications or business processes at different companies. Companies have in many cases created islands of automation that are isolated from their suppliers, trading partners, and customers. EDI has been heralded as the solution to this problem.

The origins of EDI may be tracked down to the United States when in the 1960s various industry sectors (airlines, car manufacturing and health) established the idea of EDI. In 1968 a number of companies in the transportation industry formed a committee to evaluate the feasibility of developing standards for electronic communication, called Transportation Data Coordinating Committee (TDCC). The actions of TDCC formed the foundations for EDI as used today. In 1987 the EDFICAT (EDI for Administration, Commerce, and Transport) standard was defined by the UN together with ISO as an international standard using the ANSI X12 standard and the TDI standard as a basis. The International Standardization Organization (ISO) is responsible for developing syntax rules and the data dictionary. The United Nations Economic Commission for Europe is responsible for the development of document standards (Emmelhainz 1990). Over the years,
EDI has evolved from bilateral, individually negotiated agreements between pairs of two trading partners (e.g. a buyer and a seller engaged in an ongoing business relationship), to multilateral. Today, EDI is the business-to-business exchange of electronic documents in a standardized machine-processable format. EDI enables partners to electronically exchange structured business documents, such as purchase orders and invoices, their computer systems. EDI is a means of reducing the costs of document processing and, in general, to achieve strategic business advantages made possible by shorter business process cycle times (Weitzel et al. 1999). EDI standards in wide use are the American ANSI ASC-X12 and the international UN/EDIFACT. However, these standards predate Internet e-commerce and reflect a flat-file-based, least-common-denominator approach to computer data interchange. With the explosive growth of Internet, such interchange standards are considered by many to be inflexible and inadequate. New methods of exchanging structured data based on Internet technologies are now dominant.

Traditional EDI has much to offer semantically and structurally to Internet B2B, but its syntax is definitely antiquated. XML/EDI working groups are working on helping to ensure that valuable domain knowledge codified in EDI standards that is, the semantics and structure of business documents is not lost in the rush to deploy newer technologies like XML. However, XML-based approaches have the potential to go beyond the mere reproduction of EDI transactions in XML format. XML brings the potential for document processing and validation. Of course, there are still plenty of legacy systems that need to be fed X12 or EDIFACT documents, so developers need to integrate Internet messaging and data transfer mechanisms with translators that can talk to the back-end legacy systems.

This section will briefly outline the history and technology behind classic EDI, and its advantages as well as disadvantages, and give an overview on various possibilities of integrating traditional EDI with the Internet, Internet/EDI, Web/EDI, and XML/EDI. The following section will analyze in detail the idea behind XML/EDI.
3.1. How it works - some general

Computer Systems, whether big or small, are today considered to be an integral part of most trading operations. The data for such systems will typically be entered manually before processing and subsequently written to such output documents as invoices, orders, acknowledgements, delivery notes, etc., which are then typically distributes via the postal services. With EDI, this data is extracted in an electronic form, structured according to the appropriate in-house file format. This is subsequently converted to an agreed document format standard and is the in a suitable form for transfer via some form of EDI network function, to the relevant trading partner. When received, the information will be decoded into the electronic in-house format of the receiving company.

3.1.1 EDI Message Standards

For a conversation to be effective, both parties must speak the same language. The same is true for the use of EDI. In order for computer-to-computer communication to be successful, both of the computers involved must be able to read and understand data the same way.

EDI provides this common language need. "In normal business communications, most messages follow generally accepted rules of format and syntax. These rules
of format and syntax provide an accepted way of communicating between business and ensure that the message sent is understood upon receipt. Similar rules of format and syntax exist in the world of EDI" (Emmelhainz 1990: 14).

"EDI Standards refer to standardized ways of describing the component parts of trading documents known as data items (article number, article price, unit price, name, street, postal code, etc.), and of grouping and present these data items in the form of messages or trade information (invoice, purchase order, etc.) " (Parfett 1992: 41).

The desirable solution for all trading partners is to agree upon common standards from converting data from and into their in-house systems. This is where a standard like EDI makes sense. Standards can be either (i) proprietary or (ii) common (open). Proprietary EDI standards are limited to the use of one company and its trading partners. These are systems in which the organizations have developed a standard format used in communication only with that organization’s trading partners on a closed system. While these systems are often referred to as EDI, they do not use common EDI standards. Common EDI standards are guidelines adopted by industry-wide or cross-industry users (Emmelhainz 1990).

3.1.2 Components of Message Standards

*Data Elements* are the core components of EDI holding the information. *Segments* refer to the logical grouping of data. *Message* refers to a specific Type of EDI dialogue, which is formed to achieve a particular trade function. *Functional Groups* are groups of messages of the same type, e.g. all purchase orders to one company. *Interchange* is the envelope that contains the message standards and functional groups thereof. Interchange forms the entirety of an EDI communication between trading partners. *Syntax Rules* effectively specify the grammar for EDI dialogue. *Message Design Guidelines* allow groups engaged in designing new messages or modifying existing messages to do so in a consistent manner, which will allow other users to understand them.

3.1.3 Converter and Communication Software

As explained above, EDI standards provide the structure and the common format for electronic messages. However, since no organization has its data file
structured in EDI format, some method of transforming data from a company specific format to the EDI standard format is needed. EDI software performs this function. Just as every company has its own paper form on which information is placed, every company has its unique format and structure for its database. Converting Software is needed to translate information from the company specific database into the EDO standard format for transmission.

EDI standard format are converted into a format suitable for their in-house system. The data can then be inserted into a suitable in-house file ready for subsequent processing by the appropriate application software. However, studies have shown that many companies, especially in SMEs, often printed out the received EDI message (Parfett 1992: 49).

The basic aspects, which need to be considered when looking at EDI software, as illustrated above are: (i) Data extraction, (ii) Data encoding, (iii) Data transmission, (iv) Data receipt, (v) Data de-coding, and (vi) Data insertion (Emmelhainz 1990).

![Figure 7: The Basic Functions of EDI Software. Source: (Parfett 1992: 49).](image-url)
EDI software translates information from unstructured, company specific format to the structured EDI format and then communicates the EDI message. EDI software also performs this activity in reverse (receives the message and translates from standard format to company specific format). EDI software can be either developed in-house or it can be purchased from a number of commercial software vendors.

3.1.4 EDI Networks

Classic EDI documents are transmitted electronically through phone or data lines from one computer to the other. There are two different EDI Networks, the (i) direct networks and the (ii) indirect networks.

Direct Network: Computers of trading partners are linked directly, usually through dial up modems or leased lines. When one of the partners wants to transmit documents to the other, the sender simply dials up the receiving party and transmits the document.

Value Added Networks, also known as third party networks or indirect networks, serve as an intermediary between trading partners. The VAN in an EDI world is similar to the post office in the paper world. VANs comprise two components, a network and an application. A VAN maintains a mailbox for both the sender and the receiver. In this way, if a company has a large number of electronic purchase orders to send to different sellers, the company would transmit all of the purchase orders to the VAN in a single connect or call. The application provides added value to data network and allows users to purchase facilities as electronic mail, on-line database access, and EDI. (Emmelhainz 1990: 17).

VANs offer different Services: (i) Network Services: Connects Network Services and has high level of security and systems resilience. (ii) Post/ Mailbox Services: Often referred to having 'clearing house' or 'store and forward' functions. (iii) Enabling Software Services: VAN suppliers will often also offer software, both to make connection to the network and to convert in-house document standards to and from an adopted trade standard. (iv) Consultancy Services: (v) Support Services: Executive seminars, user training, hotline support, etc (Parfett 1992: 59).
3.1.5 Cost of setting up and EDI network

Because of the organizational gap, and EDI network cannot start right away, but requires an initiative to be launched to close the gap. The first contributors to the collective action, i.e. the initiators of an EDI network, do not only have to plan, design, and implement the required techno-organizational infrastructure and the related services, but they also have to bear the associated costs.

There are different cost categories such as: (1) Coordination costs of a new network, (2) Implementation costs of a new network, (3) Individual expenses, and (4) Opportunity costs (Ritz 1995: 51-52).

The set-up process of a coordination and a subsequent implementation stage. During the coordination stage, the network's future characteristics are determined, and plans to implement the specified conception are drawn up.

The subsequent implementation of the network infrastructure yields even higher costs for the organizations involved in the collective action at that stage. Infrastructure investments foremost include the direct capital cost of acquiring the systems hard- and software, the highering of staff to operate and market it, the provision of facilities to house the system, etc. Moreover, the cost of establishing an intermediary organization may also be subsumed under this heading. Depending on the scope of this project and the situation at the outset in terms of pre-existing technical infrastructure and standards, the implementation of the network infrastructure may entail altogether substantial costs to the initiators. Some of these investments may actually represent sunk costs that display asset-specificity. Transmission costs for third party charges (VANs) or communication costs. Training costs for both in-house and purchased training for your company as well as for trading. Software costs for development or purchase of software, internal modifications to in-house systems, and maintenance. Hardware Costs. (Emmelhainz 1990: 21).

Beside the cost of implementing of the communication infrastructure, initial contributors to the collective action will have to carry the individual expenses of joining the network as participants. Usually, an organization that decides to participate in an EDI network needs to invest in its internal hard- and software, e.g. by purchasing converter software and by adapting business applications to the
communication requirements of the clearing center. At an organization level, internal business processes may have to be re-shaped in order to be able to interface with trading partners and to reap the full benefits from electronic trading. Also, retraining of employees are affected by innovations. (Ritz 1995: 52-53).

Last but not least, *opportunity costs* also have to be mentioned in this context. Those arise from the uncertainties associated with the collective action of establishing and EDI network. Should EDI initiative fail for the lack of attracting sufficient participants, investments in the joint infrastructure will be at least partly be sunk costs. Furthermore the giving up of established forms of trade data exchange when joining an EDI network also yields opportunity costs from the perspective of an individual participant (Ritz 1995: 53).

Risk of *Sunk Costs*: A very large proportion of the investments in the joint infrastructure outlined above have to be made up-front. Coordination costs arising during the process of determining the network's future characteristics and planning it's implementation. Hard- and software systems have to be acquired and installed before trial operations may begin. Intermediary organizations need to be founded and incorporated before the diffusion process starts. Once the network infrastructure is established, the marginal cost of adding trading partners is comparably low from an initiator's perspective. The reason is that the infrastructure represents fixed costs. Increasing trading volume augments total costs only slightly. As a consequence, unit cost per transaction falls, leading to significant cost based economics of scale. The danger of competing EDI initiatives superseding a focal network is very high at that early stage, and so are the related opportunity costs. The reason is the inherent uncertainty of whether a sufficiently large number of network participants will be attracted to guarantee the success of the collective action. As the number of network participants increases, i.e. as the point of critical mass is being approached, the risk of failure decreases. This leads to a stabilization of expectations among community members, which in turn attracts additional participants and enables the self-sustained growth of an EDI network. For the same set of reasons, the opportunity cost of an individual organization, that has to give up and established platform of trading partner communication, are high at that early stage (Ritz 1995:53-54).
3.2. Opportunities through EDI

EDI has potential for rationalization, possible organizational improvements, flexibility improvements, competition improvements, as well as new products- and services. In contrast to the cost of setting up an EDI network, which is predominantly arises upfront, economic benefits start to occur only during the diffusion stage. The reason is that all three categories of benefits described below are positively related to the size of an EDI network in terms of subscribers (Ritz 1995: 55).

EDI investment projects used to be justified on the basis of savings on postal charges and processing charges alone, and figures were calculated from the amount saved on every order raised or invoice processed. But time and EDI have moved on. There is a greater realization that EDI is just an enabler, and the mayor benefits that derive from the use of today's computer and telecommunications revolution will be in the change in business relationships and practices that use of timely and accurate information can bring (Parfett 1992: 2).

Parfett (1992: 12-13) concludes from a survey that the most important arguments for implementing EDI are:

− Strategic benefits - To increase the competitive edge (68%): Of crucial, long-term significance to the functioning of the organization. These benefits will affect the very business the company is undertaking, i.e. its central operating function.

− Operational benefits - To reduce costs (56%): Of mayor importance to daily operations of the company, usually only impacting on certain department within the organization.

Benefits from future opportunities - To provide new business opportunities (45%): EDI is not necessarily seen to be crucial to the current operations of the company, but as offering potential future benefits. The list of opportunity benefits includes such factors as enhanced image and competitive edge, which although perceived as beneficial, are difficult to quantify.
3.2.1 Operational Benefits - General Cost reduction

This has long been seen as the mayor advantage of EDI. The savings come from improvements in a number of areas (O’Callaghan and Turner 1995: 9-11; Seffinga 1996: 15; Parfett 1992:16):

*Better use of personnel* through reduction in document handling tasks. The savings in document handling are dependant upon how the document was processed prior to EDI.

*Improved cash flow* through reduction of inventories (Just-in-time production) and therefore reduction of money tied up in stock. EDI furthermore enables suppliers to send accurate and timely invoices thus a higher proportion of invoices get paid in time.

Reduction in other costs such as *exception handling* and *premium freights* due to a mistake in a purchase order. This includes the elimination of errors in transcribing information from one medium (paper) to another (computer) and the reduces chance of mismatching orders, ie. fewer good returned, and a cost reduction in repeat deliveries to put errors right and reduction of redundant work.

Cheaper and more efficient information processing reduces the *transaction costs* of writing contracts, and monitoring and ensuring secure commitments, thus making external provision of production more attractive. Furthermore an EDI system may reduce the amount of management influence as the routine nature of EDI transactions short circuits many internal operations and thus reduces the necessity for management intervention.

3.2.2 Strategic benefits

The implementation of EDI within an organization provides a chance to *improve internal operations* and even to redesign business processes altogether (Seffinga 1996:15).

*Improved service quality* through faster information availability, better information planning, higher services for clients and suppliers (Seffinga 1996: 5). EDI facilitates the adoption of these JIT techniques by providing the timely and
accurate information required on a daily basis, whilst promoting the trust and commitment required on a long-term basis (Parfett 1992: 14).

*Higher bargaining power:* In a highly competitive market, EDI can be useful as it provides the ability to offer significant cuts in product delivery time, which may be the difference between winning and losing a contract. (Parfett 1992: 14).

### 3.3. Barriers for EDI penetration

One of the major problems with the current implementations of EDI is that they often require a unique solution for each pair of trading partners, making EDI costly and time-consuming to implement. Classic EDI Solutions are based on established EDI standards like EDIFACT or EANCOM, need exact coordination of message structures between communications partners. This time consuming coordination as well as the development of message converter are some reason why classic EDI solution have only been used for long-term and intensive business relations.

#### 3.3.1 Limited number of standard EDI documents defined

The development of EDI standards is crucial to the widespread adoption of this form of trading. The foundations to a truly international set of documents standards have been laid in the form of the EDIFACT syntax standard. However, the number of fully approved EDIFACT messages is still small and proprietary subsets or even supersets are often used (Parfett 1992: 19).

Another problem with traditional EDI is that it is based on the use of rigid transaction sets with business rules embedded in them. These transaction sets are defined by standards bodies such as the United Nations Standards Messages Directory for Electronic Data Interchange for Administration, Commerce, and Transport (UN/EDIFACT) and American National Standards Institute’s Accredited Standards Committee X12 sub-group (ANSI X12). Transaction sets define the fields, the order of these fields, and the length of the fields. Along with these transactions sets there are business rules, which in EDI-language are referred to as *implementation guidelines* (United Nations 1999 and ASC12 1999).

A fixed transaction set prevents companies from evolving by adding new services and products or changing business processes. The bodies that make the standard
transaction sets are ill equipped to keep up with the rapid pace of change in the various business environments they impact. It is also very hard, if not impossible, to develop a one-size-fits-all solution.

According to Webber (1999b: 14) the main problems of the implementation of EDI are: (a) Version control, (b) implementation mechanics, (c) maintaining and updating the standards, (d) cost of implementation, step on ramp, (e) time to implement, (f) no provision for process and information exchange, (g) data and structure only.

3.3.2 Current EDI usage

It is symptomatic of electronic office communications technology that a critical mass of users is required before a 'big bang' style adoption of the techniques is achieved. Potential users must assess whether the population is sufficient for them to realize their investment, and certainly some business are more mature than others in this respect (Parfett 1992: 19-20).

The figures quoted for the number of companies using EDI vary from source to source. The figures lie between 2% and 5% of US and European companies have EDI. The companies using EDI come from a range of business sectors, including retail, manufacturing, transport, construction, finance and the public sector (Parfett 1992: 49).

3.3.3 Legal Problems

The legal aspects include the provision of new forms of contract, and deciding when and where the contract is formed. This issue is of particular importance as EDI transactions cross national boundaries. Computer fraud and theft of data need also be addressed (Parfett 1992: 19).

3.3.4 Conclusion

What is wrong with traditional EDI? Since the late 1960s, especially large enterprises have been using Electronic Data Interchange to expedite the exchange of business documents such as sales orders, invoices etc. In contrast to paper-based communication, EDI is designed to make communication between
different systems possible without media discontinuities. The sending company eliminates the costs associated with paper documents. It also experiences reduced order lead-time, which permits decreased inventory requirements. Invoices are also delivered sooner, allowing the potential for quicker receipt of payment. The receiving company significantly reduces key entry time and associated key entry errors. This permits business transactions (e.g., orders, invoices, and payments) to be processed faster and with fewer errors. Mailing and handling costs are also reduced or eliminated. Both the sending and receiving company gains the ability to fully control and audit the end-to-end business transaction, yielding time advantages and supporting Just-in-time production.

While this is often true for large enterprises, there is another side to EDI. Presumably, nowadays only 2 - 5% of all companies who could benefit from EDI actually use it. Especially small and mid-sized enterprises often name the high costs of implementing and running EDI as dissuading factors, and a current empirical study found pressure from larger business partners to be among the main reasons for the implementation of EDI. Furthermore, many of today’s solutions are platform dependent, meaning additional investments in both, hardware and software. Thus, EDI is predominantly implemented by large firms, preventing the electronic processing of data flowing through entire supply chains in open networks. Generally speaking, traditional EDI lacks the flexibility and efficiency required for state of the art electronic business.

### 3.4. EDI an the Internet

So far companies have mainly used the Internet for presenting themselves over the web. However, some companies have already been very successful in positioning themselves as pure Internet firms; retailers like Amazon or Dell, Portals of any kind, brokers, etc. But few traditional companies have been able to use Internet more than just as a marketing medium. There is still a big potential for them to incorporate the Internet into their business processes. Security aspects, as well as infrastructural aspects have so far been stated as reasons for avoiding the Internet. Before having a deeper look at XML/EDI in particular, it is important to point out the differences between Internet/EDI, Web/EDI and XML/EDI.
3.4.1 Internet/EDI

The Internet offers open communication infrastructure, providing a direct link to a growing Internet community. It represents an infrastructure for the transmission of EDI data of a new dimension, where the Internet can substitute the role of a Value Added Network (VAN). Internet communication will therefore allow more users to have EDI-via-internet connection than traditional VANs or direct lines. With Internet/EDI, traditional communication services such as X.400 are substituted with Internet-services. EDI Data can be attached to email messages or be transmitted via FTP. A supplier might, for example, transmit an EDI message in file format to the server of the receiver into a predefined secure directory. A considerable advantage of Internet/EDI is the Open System of the Internet and the constant expansion and cheap transmission of data via the web, which is particularly important for companies with small transmission amounts. Any company with an existing Internet access can use the same for EDI Data transmission (Klagge et al. 1998: 31-32).

3.4.2 Web/EDI

The term Web/EDI, also referred to as Form-based EDI, describes the use of the WWW as the basis for both, EDI applications and the transfer of business documents. In this case a WWW client acts as EDI application, where a company (the EDI user) provides his partner electronic forms for ordering and invoice settlement of services and goods via Browser. The orders are then transformed by the WWW-application into EDI messages and can be used like in traditional EDI data exchange by the EDI user. Existing solutions generally offer an HTML input mask, which allows the manual input of structured data, similar to the HTML front-end to a shopping system. Thus, there is no machine-to-machine connection and no way the client, usually the smaller partner, can import the EDI data into his in-house systems.

Web/EDI therefore alters the original EDI idea of frictionless data transmission, as manual data processing is partially needed. The obvious advantage of using the Web as a medium for EDI communication is that the only (client side) prerequisite is an Internet connection and a Web browser. Thus, it provides a good solution for small business with low data volumes, as only small investment into the own IT
system, coordination, and training efforts are needed to conform the EDI standards of the usually bigger business partner. On the other hand, traditional EDI users can trade more electronic data with more small business partners (Klagge et al. 1998: 32).

One pioneer of Web/EDI is Wal-Mart Stores, Inc., which has extended its private EDI network to its entire trading community using an Internet-based system. Wal-Mart has also given the required client software to several thousand smaller suppliers, who can now receive purchase orders via the Internet and send back invoices simply by completing a Web-based form. The invoices are filtered through a server at Wal-Mart headquarters and integrated into the company's existing EDI system. Like Wal-Mart, many of the Fortune 1000 companies that have made large investments in EDI are unlikely to abandon it anytime soon. Instead, they will probably use Internet-based EDI to augment their existing infrastructure and allow smaller companies to join the trading community and share in its benefits.

3.4.3 XML/EDI

By using XML to implement EDI many of the old problems of EDI (as discussed in previous chapter) are eliminated. The XML tags can actually replace existing EDI segments or data-element identifiers, which produce a somewhat bigger file than an EDI file (before compression), but in which all the labels of the data elements (in other words the description or the explanations) can optionally be used as XML tag names. The XML DTD that works with the XML carries the structural information and attribute information from the original EDI (Peat and Webber, 1999). XML therefore maintains the content and structure, but separates the business rules from the data. By focusing on exchanging data content and structure, trading partners can apply their own business rules. Using XML it is also very easy to extend the communication to support new business processes, EDI will no longer be limited to rigid standards.

The fact that XML documents can easily be distributed using the Internet is another major advantage. This combined with the fact that XML is self-explaining provides the entire framework for what the XML/EDI group calls a new supply web. Earlier many EDI implementations only worked within their own Intranets. With XML this is no longer the case since XML provides connectivity through the
Internet. All applications are then able to communicate and exchange data, thus the old point-to-point solutions are history. To many people XML/EDI is also what they call a politically correct way to merge ANSI X12 (the US standard) and EDIFACT (the international standard).

3.5. Summary and Outlook

Internet/EDI as well as Web/EDI is a good solution for companies with low trading volume and low time critical process to avoid high investment costs of classic EDI implementation. The higher the data volume and necessity for JIT processes, classic EDI solutions will be necessary. The fear of data security can nowadays be solved by firewall protecting the local servers and cryptography for the data transmission (Klagge et al. 1998: 33).

Web/EDI proves to be a good idea for large companies seeking ways of having their small customers send their data in a standardized format without forcing them to invest large amounts of money (Weitzel et al. 1999b). But in order to procure the exchange of information throughout the entire supply chain, the solutions need to be more flexible. While Web/EDI is a means of integrating small partners into existing sub-networks, the emergence of new electronic marketplaces required the transformation of "supply chains" into "supply Webs".

But there is also a downturn to the issue of EDI over the Internet. As opposed to traditional EDI, where communication flows over private Value Added Networks, the Internet is a public network and there are serious security issues to be considered. Although more than 50% of the Fortune1000 enterprises in both, Germany and the U.S. plan to use Web/EDI in the future, it is so far only implemented by 7.4% in Germany and 16.9% in the U.S (Westarp et al.1999). Questionable security is, by far, the greatest stumbling block to widespread use of the Internet for EDI. Given these considerations, compared to traditional EDI a state of the art Web/EDI solution must meet the following five requirements: (a) lower setup costs and time, (b) reduced operating costs, (c) security, (d) easy integration in in-house systems especially of the smaller partners (clients), and (e) flexibility and extensibility (Weitzel and Buxmann 1999).
In spite of the security aspect, the greatest benefits of the Internet for EDI will derive from:

− Adoption of common standards and proven inter-operable systems.
− Adoption and deployment of a distributed Directory Service capability, so that one can readily contact electronically any other organization in the world.
− Explicit commitment by participating organizations to cooperatively route traffic, work to resolve addresses, and meet required standards.
− Ubiquitous (anytime, anywhere) network coverage from many service providers. This allows the customer to choose the level of service needed.
− Layering of applications (such as EDI) over existing, proven, applications.
− A standards process with reference implementations which all vendors have equal access.
− Widely available public domain software including but not limited to applications, protocol/transports and multiple platform development tools.

The long-term solution to meet these requirements is the integration of XML with EDI. The following section will analyze in detail the concept and business impact of XML/EDI.

4. XML/EDI

The growth of the Internet and the expansion of e-commerce systems have accentuated pre-existing problems related with systems interoperability and the definition of common semantics throughout business organizations. Consequently, an increasing amount of cost and effort is dedicated to the integration of systems and applications. The issues involved, however, are not just related to infrastructure and technology. Fundamental problems exist around the understanding of what concepts are shared, how they relate and what mechanisms should be adopted to allow systems to communicate and interoperate at all levels.

XML/EDI provides 100% backward compatibility to existing EDI transactions, while moving EDI forward to the next generation. This means that it will not be necessary to discard the investment in existing EDI systems and knowledge (Webber 1998a: 40). But simply redefining message format to make the web deployable for EDI is not enough of itself. Genericity mechanisms, such as
patterns, frameworks and components, have the potential for defining, modeling, designing, and implementing shared concepts. Such mechanisms have inherited many object-oriented principles and built upon them to enhance the generalized nature of business problems and solutions both vertically (within the same domain) and horizontally (across different domains). Increasingly, emerging ontological definitions and related languages like XML supports these mechanisms. Thus a whole framework will be necessary, allowing companies to flexibly establish XML/EDI based e-commerce transactions.

In this chapter the idea behind XML/EDI framework will be introduced. Furthermore the business opportunities and challenges of XML/EDI will be discussed, followed by a simple example for demonstration purposes.

**4.1. The Idea of XML/EDI**

By using the XML extensible tag set, EDI objects can be either passed or dynamically referenced to objects stored in repositories. The XML/EDI Group (see chapter 4.1.4. and Appendix) is proposing the use of XML as a carrier for the document information so that the transaction can carry not only data, but also code (at each level in the transaction tree). With an element having data properties and code methods, this allows the business elements to be manipulated as objects.

First, the idea behind the XML/EDI framework and the layer architecture will be introduced. Furthermore future scenarios of how XML/EDI standards may evolve will be discussed.

**4.1.1 Fusion of Five**

Classic EDI is captive to its own fixed structures and inflexibility. To save XML/EDI suffering this same fate, the XML/EDI guidelines proposed by the XML/EDI Group adds three additional key components: *Process Templates, Software Agents, and Global Entity Repositories* (Webber 1998a: 40).

These three additional components transform old EDI into XML/EDI that allows full dynamic e-commerce between business partners and avoids the mistakes of the past. Each component adds unique tools that leverage the other pieces to make
XML/EDI a dynamic process that can be infinitely extended. The combination of XML and EDI semantic foundation will provide a complete framework where a set of different technologies work together to create a format that is usable by applications as well as humans. These components all work together to create an XML framework for business use (Peat and Webber, 1999).

- **XML** (transport, metadata, parsing)
- **EDI** (business transactions)
- **Repositories** (business process entities)
- **Templates** (business rules & information exchange)
- **Agents** (control and facilitation)

Figure 8: XML/EDI the Fusion of Technologies 8 (Peat and Webber, 1999).

XML itself provides the foundation. The Web was born on the abilities of the HTML language, itself a very limited subset of the original and highly complex SGML document syntax. Now XML has been created that sits between the two, not as complex as SGML, but vastly more capable than HTML. XML tokens and frameworks are the syntax that transports the other components across the network. XML tokens replace or supplement existing segment identifiers. XML also brings with it all the rich capabilities and transport layers of the Web and the Internet in general.

**EDI** is the grandfather of the current e-commerce. Using EDI, it is not necessary for the trading partners to have identical document processing systems. When the
sender sends a document, EDI translation software can convert the proprietary format into an agreed upon standard. When the receiver receives the document, his EDI translation software automatically changes the standard format into the proprietary format of his document processing software. But EDI is costly and time consuming to implement, and its standards are predefined by standardization bodies, and therefore inflexible.

*Process Templates* are the rules that determine how XML files should be interpreted. They can define the layout of the file and are supplemented by DTDs that enable transaction operability. Templates hold the whole process together. They travel inside the XML as a special section and let two organizations understand each other’s data and instructions for that data. Templates can be stored in repositories for global reference (Webber 1998a: 40).

*Software Agents* interpret the templates to perform the job need to be done (such as accessing local data stores), and also interact with the transaction and help the users to create new templates for each specific task (Webber 1998a: 41).

*Repositories* are locations where shared and agreed-upon Internet directories are stored and where users can manually or automatically look up the meaning and definition of XML/EDI Tags. The repository is in fact the semantic foundation for the business transactions (Webber 1999a: 41).

### 4.1.2 XML/EDI Core Models

According to Peat and Webber (1999) XML/EDI provides four core models of use. These include traditional EDI deployment methods, along with new document-centric capabilities. Figure 6 shows at a glance the four core models: (i) *Star Model*, (ii) *Ad-hoc Model*, (iii) *Hybrid Model*, and (iv) *Web Model*.

The *Star-model* shown is the classic EDI model, where a major business partner or organization sets the standards for its trading partners. The *Ad-hoc-model* is the new net-based model. Smaller trading partners setup their own ad hoc interactions, these in time may evolve into more formal methods, or they may not.
The Hybrid-model is a combination of the first two. Here a "Star" model is extended by trading partners, by creating new versions of frameworks, and by linking in their own ad hoc ones. The Web-model is a document-centric model, where content is the most important information being exchanged. Content can either arrive driven by pre-set rules, or be requested, or be broadcast. The classic example for this is an electronic catalog, and the associated Request for Quotations (RFQ) dialogs.

### 4.1.3 XML/EDI as a Framework

XML/EDI provides for the infrastructure of a wide variety of e-commerce systems, from searchable on-line catalogs to robust machine-to-machine transaction subsystems. The XML/EDI initiative is to provide open solutions for implementing e-business systems today, as there is more than just one standard solution for all of the e-business scenarios. Each scenario has its own requirements and goals. Thus a framework and not an application or module is required. The goal of the framework is to provide formal interfaces for commercial e-commerce components to interoperate. For XML/EDI to be successful these interfaces will be open and yet standardized. The business model is either of ad hoc interactions between...
small groups, or agreed upon national or international frameworks, such as those by trade associations or industry bodies (Peat and Webber, 1999).

Figure 10: XML/EDI Layered Architecture (Peat/Webber, 1999).

Figure 10 depicts the technical layers that a base XML/EDI system can be built on. Not all layers are required to achieve results with XML/EDI. The framework builds upon and enhances the XML and EDI standards and defines the blueprint of e-commerce component interaction. For example, catalog vendors may only want to implement the XML tagging. The tags are defined in standard repositories. The various layers support different targeted e-commerce systems. Overall each successive layer provides increasingly more sophisticated capabilities so as to handle more demanding needs (Peat and Webber, 1999).

The framework is not an all-or-nothing implementation. Each e-commerce component is designed to be able to be used independently, interfacing with each other as defined by XML/EDI standards. DTDs formally define the structure of EDI messages without having to use it to validate well-formed XML documents. EDIFACT/X12 messages can simply be placed in an XML shell element or the message (or parts of it) can be coded in XML. Messages can be formally validated, or simply checked to be well-formed. Messages can be linked to data stored elsewhere, or completely stored internally. Documents can be linked to rule-templates, or be in-built (or no) rules for processing the received data. There is a variety of possibilities (options) as how to integrate EDI with XML. End Users interact with these technical layers through highly visual desktop tools. Developers
use e-commerce components to build these tools. Users interface with their documents, as they would use any tool in their desktop applications, word-processing, spreadsheets, databases or whatever metaphor is most suited to the application.

4.1.4 Progress in XML/EDI Standards and Frameworks

The XML/EDI Group was set up in 1997 to determine how the expertise built up by the EDI community over many years of study into the problems of business-to-business communication could best be expressed in terms of XML. Their goal was to allow EDI-based transactions to be integrated with other types of communication currently taking place over the Internet to form a virtual electronic environment for business transactions (XML/EDI Group, 1999).

The XML/EDI Group has formed a working group in 1998 to organize the development of a global repository. The goal of the XML/EDI repository working group, was to develop a repository for business transactions based on the Extensible Markup Language. The repository would store the tag sets, document type definitions, Extensible Stylesheet language templates, and other schema needed for effective business-to-business communication (XML/EDI Repository Group 1999a). Posting DTDs online and sharing them among partners is not a new concept, but previously it had only existed among small groups. By opening up the repository, it will serve as a global library for companies to share their preferred formats. The repositories are not limited to text, however. Anything that can be sent via MIME can be housed in the repository, which opens it up to scripts, Java applets, and various forms and templates.

Furthermore, the set of draft Guidelines for the use of XML for EDI needed to be tested in a European scenario to ensure that they are adequate to meet the multilingual and mixed trading practices found in Europe; this is done by the European XML/EDI Pilot Project. Subject and Scope of this group is to test the applicability XML for interchanging data, between small businesses and their business partners, of the type currently exchanged using EDI messages for the European market (Bryan 1998a).

All these XML/EDI movements realized that there is a big downside to XML. In spite XML’s positive features and prospects it must be clearly stated that XML is
solely a description language to specify the structure of documents and thus their syntactic dimension. The document structure can represent some semantic properties but it is not clear how this can be deployed outside of special purpose applications. XML allows individuals and organizations to create tag sets that describe more than just how to display their information. But there still is an interoperability problem. If a university and a furniture store both use the tag <Chair>, do they mean the same or different things? What if another furniture store uses the tag <Seat>? XML DTDs can be used to ensure that a set of documents use the same set of tags, but it would be impossible to create a single DTD that describes everything. But today’s E-commerce requires rich data. Specific XML formats for each exchange task improves the situation, but ones misses the network effect of being able to share 90% of the processing software, because the XML data model is too low-level.

Thus, the ebXML Initiative was formed to try to solve this problem. ebXML is supposed to take things a step further on a couple of levels than XML, by allowing the business process as well as the data to be communicated. But ebXML is not the only initiative that tries to coordinate data and process. There are others like RosettaNet, which are backed by big players like Intel and Siemens. See Appendix two for some detailed information on current developments and special focus of the working groups.

4.2. Opportunities through XML/EDI

As discussed above, history shows that in the past traditional EDI failed to create a broad based acceptance for a number of reasons. The most important one being that setting up traditional EDI is an expensive and time consuming manual process as trading partners attempt to synchronize their internal systems with the external systems of their partners. Worse, some large companies, such as the mayor manufacturers have potentially three hundred thousand trading partners in the USA alone (Webber 1998a: 38).

The Internet proposes a vast opportunity for businesses to conduct a form of electronic communication. In contrast to traditional EDI, Internet use has grown astronomically in the last few years. This section will discuss the impact of the Internet, and thus XML, on the future of EDI.
4.2.1 Impact of XML on EDI

What XML provides is primarily (a) Self-explaining syntax, (b) Modularity, (c) Extensibility, (d) Presentation, and (e) Transformation.

**Self-explaining syntax**: Since the elements in XML are discoverable using document type definitions, the elements used to describe a supplier's product, its pricing, or other attributes can be gleaned without first having to agree upon a single format beforehand. Previous visions of EDI could not use this kind of ad hoc partnership approach. At best, an industry might define a set of EDI templates or forms for specific transaction types. Generally, these would be established by the largest company in a supply chain as a de facto set of transaction standards and data types.

**Modularity**: The XML/EDI messages can be constructed using a combination of several standardized modules. It is possible to provide a number of standardized and publicly accepted building blocks that can be used to construct more complex EDI messages. This is quite different from the current implementations where all the functionality has to be included in the messages from the beginning, resulting in that people are adding all kinds of messages in the standards because they may be used in the future.

**Extensibility**: Since the EDI standard using XML is no longer under the eye of standardization committees, such as UN/EDIFACT and ANSI X12, it is much easier to add new support and functionality in the EDI messages.

**Presentation**: The XML/EDI messages can be presented directly to the user using the XSL specification (when and if such support becomes available in the browsers).

**Transformation**: The XML/EDI messages can be processed transformed and analyzed using the proposed XML/QL standardization.

4.2.2 Reduction of General Costs

Compared with conventional EDI approaches, the Internet offers less costly transactions, easier access, broader reach, and improved functionality. With its flat-rate fee structure, for instance, the Internet is substantially cheaper than the
value-added networks (VANs) traditionally used for EDI, which charge by the byte for each EDI message. Using traditional EDI, it costs about $2.50 to process a purchase order, compared to around $50 to process a paper-based version, according to Forrester Research, Cambridge, Mass. Internet-based EDI can lower the transaction cost still further, to less than $1.25 (Edwards 1999).

The Internet also makes EDI accessible to the companies previously shut out by the high start-up fees and complexity of implementing and maintaining the required systems. In large companies, these costs can be justified by the increase in efficiency and the additional revenues generated from trading partners. For smaller companies, though, the EDI costs may be prohibitive, so millions of firms are denied these revenue opportunities. With a Web-based EDI service, companies can now be part of the action for less than $1,000 a year, compared to the estimated cost of $50,000 to join a traditional EDI network.

4.2.3 Hub and Spoke Relationship Improvement

For the moment, though, rather than eliminating EDI, organizations are combining it with Internet transaction systems to make the service faster and more comprehensive, and to add new partner, supplier, and customer connections, especially with smaller firms. This is a win-win development, since the more partners and suppliers a company connects, the larger the savings and production efficiencies and, often, the greater the revenues.

Changes are also occurring in the traditional hub-spoke EDI business model, where the hub is the organization that initiates the transaction and the spoke the trading partner supplying products and services to the hub. Increasingly, organizations are adopting a dual-capacity role as both hubs and partners, creating an extended EDI model. The Internet takes this evolution a step further by adding extranets with the potential for connecting many more partners, suppliers, distributors, and customers.

According to market researcher International Data Corp. (IDC), dual-capacity hub-partners will account for the strongest growth in EDI transactions over the next few years, jumping from one-quarter of EDI service revenues in 1998 to two-thirds by 2003. IDC also predicts that the Internet will handle 41% of EDI transactions by 2003, up from 6% in 1998. It still sees a rosy future for EDI, projecting an 18.6%
compound annual growth rate in EDI service revenues to reach $2.3 billion by 2003, with the Internet portion accounting for 33% of the revenues (Edwards 1999).

4.2.4 Consequences for Business

EDI has traditionally used unique segment identifiers like tokens to separate and identify data items within messages. So replacing those same segments with web tokens allows XML to express EDI and carry EDI via Web delivery methods. This moves EDI from the arcane, obtuse and static into the dynamic mainstream of computing. Webber (1998a: 41) predicts that XML will move EDI from the inadequate 10,000 companies using traditional EDI in the US to millions of desktops and companies instead. Webber (1998b: 42) furthermore states that XML/EDI enables:

- Delivering affordable automation for Small and Medium Enterprises.
- Automation of the supply chain with e-commerce sites.
- Delivering scalable cost effective solutions for mayor sites to reach tens of thousands of partners.
- A robust desktop metaphor that integrates into tomorrows office solutions.

XML/EDI allows each trading partner to quickly synchronize their systems by exchanging not just EDI data, but also process control templates as well. Thus not only this data exchanged but also the enabling underlying process information. Additionally the process control templates are supported by the use of software agents (typically Java and ActiveX components) and Internet based global Reference repositories that allow the required process to be both directed and centrally coordinated. This means that large companies can provide a foundation that their smaller trading partners can easily download and re-use or simply adapt for their local needs. Extended rule based businesses are now possible going beyond traditional process flow forms, and including dynamic and ad-hoc systems that adapt. This is the model needed to create full electronic enterprises where documents and business information flows and is exchanged easily. Other business models are also enabled. The use of HTML on the web has already allowed product catalogues to be integrated into the sales order process. Better
yet with XML markup creating objects and meta-data new uses for catalogues, that allow customers to request information based on criteria and rules can be setup. Also XML documents include web-style content such as graphics and multimedia, not just bare text (Webber 1998a: 39).

4.3. Challenges of XML/EDI

In spite of all advantages of XML/EDI that were outlined in the previous section, there are many challenges to this new standard. First of all, the standard is still evolving. Many believe that multiple proposals may cause incompatibilities. XML is touted as an industry neutral language that could revolutionize information exchange in much the same way that HTML has defined the Web. But XML's greatest strength, which is allowing developers to custom-build systems for data exchange, could also be its downfall, leading to incompatible versions that could tarnish XML's cross-industry appeal, analysts said. (Cnet News 1998).

XML is still young in terms of practical development experience. Critics fear that the relentless flood of specification development and marketing endangers the discovery and communication of solid XML programming experience. They see a threat in overselling the XML idea, and would prefer to see the hype machine slow a little and getting on with building practical systems instead.

As XML standardization efforts have not really caught on yet, organizations find it easier to create custom tags and just map data to achieve interoperability. Many corporations today are building effective XML utilities without waiting for standardization. Experts point out that this is not much of a problem after all, as it is easy to map one application's tag set to another if and when tagging standards change. XML is flexible enough, as long a high-level set of rules is followed. Companies are not looking for external definitions of XML tags, they are rather building their own sets of tags for internal use and translating them to communicate with external partners.

XML has barely penetrated the vast established base of Web browsers and other HTML tools, which means XML data may not display properly in all applications. Older browsers do not natively render XML. Companies still have to convert XML to HTML on the fly. That is one feature to look for in an XML application server.
XML/EDI documents are also much larger than in traditional EDI messages. Overall response time of XML applications slows as the data store grows larger. Object databases, which store XML content, are not as highly tuned as standard SQL-based relational database management systems. Using XML is not necessarily proven in terms of performance. Many believe that these problems will disappear as XML settles in.

The XML/EDI groups are working to define XML namespaces and DTDs equal to the EDIFACT standard to support commercial usage of EDI over XML. Even though most organizations agree that EDI using XML is a good idea some are still skeptical. This is due to the fact that they think that EDI is too complex to be dealt with easily. In the healthcare industry alone, there are 400 different formats for a claim. One of the key issues will be how compatible the new implementations of EDI will be to the old ones; currently there is no answer to this question.

Furthermore a big debate is going on between the different XML/EDI groups on what the structure of the XML tags should be. Some advocate the use of the full description in the tag and others the use of the ANSI element number as the tag name. Others want to use a unique number that would include the standard version of the directory, the segment it resides in and the data-element it represents.

Concerning the current developments in the ebXML area: “None of the vendors quite knows where we are running to, but they are running faster and harder than before to get there before their rivals” (Dumbill 2001). Vendors claim to be cooperating on specifications, but compete on implementations. Despite vendor assurance, it seems plain that the race to compete on implementation puts the viability of multivendor institutions at risk.

4.4. XML/EDI Example

EDI is quite different from sending electronic mail messages or sharing files through a network, a modem, or a bulletin board. The straight transfer of computer files requires that the computer applications of both the sender and receiver (referred to as “trading partners”) agree upon the format of the document. The sender must use an application that creates a file format identical to the receiver’s computer application.
A retailer stocks inventory from a major manufacturer. For years this retailer has been ordering merchandise from the manufacturer using a human-readable purchase order such as that in Figure 11 for 1000 fuzzy dice. Now fuzzy dice manufacturer is demanding that the retailer begin using EDI for transactions or surcharge will be added to every order to offset the cost of handling manual transactions. The small retailer can't find a more lenient supplier, so it has little choice.

![Figure 11: Plain purchase order](image)

Unfortunately, EDI compliance means that the retailer will now have to transmit its purchase orders in a format specified in one of the several standard sets for EDI format, the most widely used being ANSI X12 and UN/EDIFACT. Figure 12 shows part of an ANSI X12 version of the purchase order in Figure 11.

![Figure 12: Fragment of ANSI X12 transaction set corresponding to Figure 11](image)

It's possible that the retailer's current accounting software supports EDI transactions, but not likely. It will probably have to either convert to software that
does, purchase software that can make the translation (if available), or contract
with a third party to convert the data on an ongoing basis.

Would there be any advantage, for the retailer or the manufacturer, to using an
XML format instead? Perhaps the retailer's accounting software is tailored to
Internet commerce, and already uses XML formats. In this case, there are utilities for converting one data type definition (DTD) to another, and a generic tool accomplishes the conversion to EDI. The resulting transmission would certainly be more comprehensible to humans, and perhaps even feasible for an employee with a text editor or XML editor to generate. An example of how the sample purchase order might look in an XML rendering of X12 is given in Figure 13. This example is based on the ongoing work of the XML/EDI group. One disadvantage with using XML, as can be seen below, is that the size of the EDI messages gets a lot bigger than the X12 messages. In this example, the XML message is about 8 times bigger than the X12 message.

The idea of XML repositories is to provide a means for industries to store the document-type definitions, that identify the data elements and their relationships exchanged among parties doing business electronically, on the Internet. Repositories contain logic components, such as Java applets, template scripts, forms and object definitions needed to process message components. With common registration procedures for these components, repositories will act as global libraries, and enable industry groups, government agencies, and businesses of all sizes to make their preferred message formats widely available to current and potential trading partners. The intent of the Global Repository is to be a dynamic mechanism that responds through an open Application Programming Interface (API). It is anticipated that fully functional global repositories will evolve in phases:

**Phase 1** Limited Intranet implementation, proof of concept, with manual Web interface.

**Phase 2** Definition of basic API allows extranet implementations between specific partners.

**Phase 3** Definition of full API and available repository products allows full implementations.

**Phase 4** DNS style service established and standards bodies adopt support for API and long-term maintenance and alignment.
(1) The Retailer queries the Global Repository for the structure (DTD) of those common business objects that are to be passed to the trading partner, the manufacturer.

(2) The business objects are passed to the Manufacturer, using XML documents structured according to the DTDs fetched from the Global Repository.

(3) The Manufacturer maps the received data into the organization's application system. The mapping is received from the Global Repository.

4.5. Summary and Outlook

The Fundamental B2B Problem so far has been the lack of shared semantics. The main challenge in using XML is not with document structure and syntax, but with semantics. It is much easier to share syntax than to share semantics. Each application domain needs to have both specific document structure and agreed-upon semantics to make effective use of XML-defined documents. Various vertical industry segments have to agree on the meaning of their domain-specific XML tags. Only then can the appropriate action be taken upon parsing: translating components into internal business objects for use within the enterprise.

XML can make EDI transactions accessible to a broader set of users. EDI is already a powerful tool that has been deployed by large organizations around the globe to exchange data and support transactions. But today, EDI transactions can
only be conducted between sites that have been specifically set up to exchange information using compatible systems. XML will allow data to be exchanged, regardless of the computing systems or accounting applications being used. XML should create new possibilities for how data is used, and there are many initiatives under way to move EDI to XML.

Formerly expensive and inflexible EDI-processes can become manageable for SMEs, too, enabling the next step towards open communication and cooperation networks and supporting the flow of information throughout entire purchasing and supply chains. The use of XML also enables the efficient support of various data formats, including EDI or EDIFACT standards, and different business documents such as orders, invoices or catalogs opening EDI networks to SMEs and enabling more dynamic business partnerships.

Based on the findings about possible advantages of XML/EDI for a company in particular and whole industries in general, the next chapter will discuss how a company today can decide whether and when to invest into an XML/EDI project.
Part Two

Benefits of electronic markets that are realized by individual participants increase the more organizations join the system. This property, known as network externalities, can significantly affect the dynamics of the introduction and adoption of electronic market technologies like XML/EDI. Thus, a company facing the XML/EDI investment decision today, must assess its strategic market position and try to evaluate the costs and benefits of adopting this novel technology at a certain date.

Therefore, chapter five will analyze existing investment decision techniques, concluding that for strategic IT investments real option valuation techniques are appropriate. Unfortunately this type of assessment method is very mathematical, and some of the formula’s parameters are too complex to estimate. Thus Chapter 6 will introduce an investment decision framework based on the Black Scholes formula and the theory of technology diffusion, to help visualize the XML/EDI investment decision for a company facing the investment decision today.

5. Assessing the Value of IT Projects

When it comes to deciding about an IT project, the costs can be defined and controlled quite easily. But the benefits are usually only covered by a few statements full of hope about the effect on large labor costs, and deal with intangible expectations which cannot easily be defined (Strassmann 1985: 81). This section will give an insight into traditional methods of evaluating IT investments.

The demand for increased productivity, and the desire to improve responsiveness to customer needs are the driving forces behind Information Systems integration. For more and more companies today investing in IT is the only way they can stay in business. During the past decades, organizations have spent enormous sums of money in computer hardware, software, communication networks, databases, and specialized personnel, collectively known as Information Technology. Leading-edge firms all over the world in various industries have increased their overall IT expenditures by double figure percentages annually. Some surveys even state
that up to 50 percent of business's total capital expenditures for IT. Obviously, investments in information technology must meet the same criteria of justification as any other investment. The aim of this chapter is to show why traditional capital budgeting methods have failed to assess the true value of many IT projects in the past, and to introduce the concept of Option Pricing Methods (OPM).

The first part will discuss traditional capital budgeting techniques, their accurateness of IT investment evaluation, and whether and to what extent they are used in the industry. The second part will outline the history and idea behind the Real Option concept, analyze the Black-Scholes formula, and discuss several types of Real Options. Finally the advantages and challenges of OPM will be outlined, comparing OPM with the Net Present Value (NPV) technique, and discussing for which projects OPM make sense to be applied to.

5.1. The IT Productivity Paradox

Researchers and practitioners have very mixed observations about the value of IT investments. Existing literature provides little evidence of a correlation between IT investment and business performance, which is referred to the IT Productivity Paradox. The question as to whether it is even possible to measure the value of IT has not been agreed upon yet. While in recent years progress has been made in the development of approaches and techniques for ex-ante justification of IT projects and to some extent for the justification of investments in IT infrastructure, little progress has been made in approaches and instruments that serve the ex-post valuation of IT investments. Many studies have been carried out to see whether money spend on IT leads to improved business performance. None however provide conclusive proof that it is so.

Strassmann’s correlation studies from 1989, typical of the kind of studies that have been carried out, show no correlation between the proportion of corporate revenue spend on IT and either returns on assets of shareholder’s investments.

Peter Weill (1989) conducted a 6 year study of IT investment in Australian Business. He explored the effect of IT investment on firms’ performance and concluded that on average, there is no evidence of any positive result. The reasons therefore cited are:
A large amount of information technology investment (about 50%) is made for infrastructure purposes and is unlikely to be associated with measurable financial gains or losses.

The traditional methods of analysis under DCF are not appropriate with this type of investment.

Firms which associate heavily in strategic IT were associated with relatively lower non-productive labor productivity and sales growth in the short term. In the long term the effects were neutral. Weill believes that “strategic investments give a short-term edge, which is lost when the technology becomes common”.

Henry Lucas (1999) introduces the investment opportunity matrix, where he distinguishes different types of investment. These are:

- Infrastructure IT.
- IT investment that is required (in this case firms do not expect returns).
- IT Investments where a certain technology is the only way to do the job.
- IT investments where there is no direct return from IT.
- IT investments with indirect returns (hard to measure). Cash flow will be in the future. Increased market share etc.
- Transformational IT

He analyzes each of these investment types regarding predictability of future cash flows. He summarizes the outcome of many studies in this area and concludes from ex-ante case studies that, except for the case where there is direct return, productivity is hard to estimate. He makes qualitative analysis of past IT-investment case studies of major companies, like the SABRE System of American Airlines, Chrysler’s introduction of EDI, and many more. Although he cannot calculate a rate of return due to the lack of numbers, he shows strong evidence that the companies analyzed have obtained significant value from their investments in IT looking at their overall company performance. Like Strassmann, he suggests that depending on the kind of investment, different capital budgeting techniques must be applied. He introduces the Real Option Method as the one to be used for investments with many sources of uncertainty.
The next chapter will briefly outline the traditional investment decision rules used, and discuss the reasons for them not being able to correctly assess the value of many IT investments.

5.2. Traditional Methods – Investment Under Certainty

In the past decades, a large body of literature has developed on the subject of capital investment decision-making. Although there are several methods traditionally used, only the Net Present Value (NPV) method has so far been accepted by the accounting profession as correct and consistent with shareholder wealth maximization. However, surveys show that for the sake of convenience, many companies today still use, less exact methods, as they are simpler to apply.

Investment decision rules are usually referred to as capital budgeting techniques. According to Copeland and Weston (1992: 26) the best technique will possess the following essential property: It will maximize shareholder’s wealth. This essential property can be broken down into separate criteria:

− All cash flows must be considered.
− The cash flows should be discounted at the opportunity cost of funds.
− The techniques should select from a set of mutually exclusive projects the one that maximizes shareholders wealth.
− Managers should be able to consider one project independently from all other (this is known as the value additivity principle).

These theories assume that capital markets are frictionless¹ and that the stream of cash flows estimated without error, which is also referred to as “Investment under Certainty” (Copeland and Weston 1992: 25). The traditional methods can be divided into static and dynamic methods, the latter considering the time value of money.

---

¹ In a frictionless market (1) financial managers can separate investment decision from individual shareholder preferences, and (2) monitoring costs are zero, so that managers will maximize shareholder's wealth. All they need to know are cash flows and the required market rate of return for projects if equivalent risk (Copeland and Weston, 1992: 25).
5.2.1 Static Methods

Accounting Rate of Return: The ARR is the average after tax profit divided by the initial cash outlay. It is very similar to the Return on Assets, or the return on investment. Copeland and Weston (1999: 28) do not regard the ARR as an appropriate capital budgeting technique as it uses accounting profits instead of cash flows and it does not consider the time value of money.

Pay-Off-Method (Pay Back Method): The payback period is the number of years it takes to recover the initial outlay. The investment cost is divided by the net annual savings in order to determine the time required to recoup the investment. The investment with the shortest payback period is chosen. If management were adhering strictly to the project, it would choose the project with the shortest payback period. The difficulty of the payback method is that it does not consider all cash flows and it fails to discount them. Copeland and Weston (1992: 26-28) reject the Payback Method because it violates three of four properties that are desirable in capital budgeting techniques as stated above (1,2 and 3).

5.2.2 Dynamic

Internal Rate of Return (IRR): The IRR on a project is defined as that rate, which equates the present value of the cash outflows and inflows, i.e. where the NPV is zero. Hence this is the rate of return on invested capital that the project is returning to the firm. If the IRR criterion is used and the projects are independent, any project that has an IRR greater than the opportunity cost of capital will be accepted. According to Copeland and Weston (1992: 36) the IRR rule errs in several ways. First, it does not obey the value additivity principle, and consequently managers who use the IRR cannot consider projects independently of each other. Second, the IRR rule assumes that funds invested in projects have opportunity costs equal to the IRR of the projects. This implicit reinvestment rate assumption violates the requirement that cash flows be discount at the market determined opportunity cost of capital. Finally, the IRR rule can lead to multiple rates of return whenever the sign of cash flow changes more than once.

Net Present Value (NPV): NPV techniques originally evolved to value assets in financial markets. Later, the concept of NPV was extended into the corporate realm to evaluate individual investment proposals. The NPV is computed by
discounting the cash flows at the firms opportunity cost of capital. The NPV method aims to find projects with a positive NPV (greater than zero). Copeland and Weston (1992: 28) state that the NPV of a project is exactly the same as the increase in shareholder's wealth. According to them, the NPV technique is the correct decision rule for capital budgeting purposes.

5.3. Investment under Uncertainty

The methods discussed above only consider projects where future cash flows are certain. While discounted cash flow (DCF) evaluation methods can work well when a project generates predictable cash flows, they can perform poorly when a project provides managers with flexibility regarding future decisions. Most recently, a new set of valuation techniques using probabilistic methods have been developed.

5.3.1 Decision Tree Analysis

A decision tree is a graphical extension of the Expected Value (EV) concept. To calculate the expected value of a certain scenario, the values of the endpoints (possible scenario) are simply multiplied by the probabilities of reaching these endpoints. To consider the time value of money the value of each endpoint are discounted back to the present. One of its strengths is that the tree diagram clarifies the decision problem with a visual approach. The diagram clearly shows all important decision elements, including contingencies (outcomes of chance events); decisions and alternatives, the logical sequence of decision points and chance events. One can often solve decision trees by hand, and the method is helpful at providing insights, especially related to the value of additional information and additional control. Branch and outcome values must be value measures (Present Value works, but, for example, internal rate of return (IRR) does not), and probabilities must be assessed or calculated for all chance event outcomes. One of the weaknesses of decision tree analysis is that one must represent all possibilities by a finite number of paths through the tree. This limits the practical number of random variables that can be accommodated, and one must use discrete distributions as approximations to continuous probability functions. Thus, range extremes are not represented. Other weaknesses of this approach include the fact that the analyst must limit the representation of uncertain time-spread events, such as for prices and inflation, to a few scenarios. Also, in
tree diagrams an output probability distribution (risk profile curve) is usually not obtained, only the EV. However, with a modest extra effort, the distribution curve can be calculated by "rolling-forward" the tree.

5.3.2 Real Option Methods

One of the most frequently mentioned probabilistic valuation technique is the "real options" methodology, which allows investment projects to be valued using methods developed to price financial options.

A financial option conveys the right but not the obligation, to buy or sell an asset at a predetermined price at a future time. Options on real assets behave rather like options on financial assets (puts and calls on shares or currencies). The similarities are such that they can, at least in theory, be valued according to the same methodology.

The real option method has generated much interest in recent years and tends to be regarded as the very best an latest in valuation methodology, at least by many academics and consultants. In practice, however, many companies tend to avoid applying this technique.

5.4. Empirical Studies of Accounting Methods used

This section summarizes a study identifying the adoption of the capital budgeting methods discussed above.

Lilleyman (1984) examined the capital budgeting practices of a sample of Australian organizations. Questionnaires were mailed to 371 organizations selected among public companies, private companies and semi-government organizations. The number of valid responses received was 98. The table below shows the techniques used and the importance that respondents attached to the technique. Some respondents used more than one method hence the total exceeds 100%. In the importance column 20% did not reply.

Although the payback method was extensively used because of its simplicity and ease of understanding, the discounting techniques were recognized as being most important. The main argument by the responders for not using the DCF were that:
− Present methods give satisfactory results.
− Risk and uncertainty of business make discounting methods not worth the time and cost involved.
− Business and economic data is unreliable.
− Management is skeptical of the practicality of the discounting techniques.
− The problem areas identified were: forecasting future cash flows, recognition of future expenditure opportunities and recognition and assessment of non-profit-factors.

<table>
<thead>
<tr>
<th>Methods</th>
<th>% Using Technique</th>
<th>% Importance of technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payback</td>
<td>69</td>
<td>10</td>
</tr>
<tr>
<td>Internal Rate of Return</td>
<td>47</td>
<td>23</td>
</tr>
<tr>
<td>Net Present Value</td>
<td>44</td>
<td>17</td>
</tr>
<tr>
<td>Accounting rate of return</td>
<td>61</td>
<td>15</td>
</tr>
<tr>
<td>Other</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

Figure 15: Rating of Usage and Importance of Evaluation Techniques (Lilleyeman 1984)

Strassmann (1985: 100) stated that when executives where asked about the mayor problems encountered in IT productivity projects, they emphasized that productivity was very difficult to measure, especially when the objectives where qualitative.

In as more recent study PriceWaterhouseCoopers (1999) interviewed 150 companies. They found that “notwithstanding the emergence of more sophisticated techniques ‘old-fashioned’ DCF remain remarkably widespread in practice. The survey shows that 67% of the companies use DCF to evaluate and select investment projects. About 80% would not consider probabilistic analysis or option pricing analysis.

Bain conducted a survey comprising 6 years of data and over 4000 respondents from all continents, which shows that 46% of North American firms that tried real
option evaluation techniques would not use them again (Bonduelle and Schmoldt 2000: 36).

This chapter showed that in the case of investment under certainty other decision criteria than the NPV method, such as the payback method the ARR, and the IRR, do not necessarily guarantee undertaking projects that maximize shareholder's wealth (Copeland and Weston 1992: 41). Nevertheless, although NPV methods are increasingly in use, static methods are still widely adopted for assessment purposes. It was further discussed that researchers like Strassmann and Lucas state that most IT investments incorporate highly uncertain future outcomes, thus methods of investment under uncertainty have to be applied. In the case of investment decisions that face high levels of future uncertainties probabilistic methods and real option methods would be appropriate. Trigeorgis (1993) points out that early critics (Dean, Hayes and Abernathy, and Hayes and Garvin) also recognized that standard DCF criteria often undervalued investment opportunities, leading to myopic decisions, underinvestment and eventual loss of competitive position, because they either ignored or did not properly value important strategic considerations.

5.5. Option Value in IT – Real Options

A revolutionary concept is emerging in corporate finance called real options theory. In a nutshell, it says that when the future is highly uncertain, it pays to have a broad range of options open.

5.5.1 Real Options – Some General Facts

The concept of a real option arose from earlier research on financial options by Fischer Black and Myron Scholes (1973) as well as Robert Merton (1973). The research, which transformed the valuation of financial instruments such as derivatives, led to Scholes and Merton being awarded the 1997 Nobel Prize for economics. It was noted by Stewart Myers (1977) that similar ideas could be applied to real (i.e. non-financial) assets and the term, real options was born.

An option conveys the right but not the obligation, to buy or sell an asset at a predetermined price at a future time. Myers saw that many business investment decisions have similarities to a financial call option. By commencing an uncertain
research and development project, a firm has in effect purchased an option. At various stages in the future, the firm's managers will be able to respond to the preliminary results of the research and will have the option to continue, abandon, scale up or scale down the project. Typical applications of DCF appraisal techniques ignore this option value and could undervalue the project by using appropriately high discount rates, causing it to be rejected.

5.5.2 The Formula

The formula for pricing European call options\(^1\), \(c\), on financial assets developed by Black and Scholes (1973) has been described as a function of five parameters: The price of the underlying asset, \(S\); the instantaneous variance of the assets returns, \(\sigma^2\); the exercise price, \(X\); the time to expiration, \(t\); and the risk-free rate, \(r_f\).

\[
c = f(S, X, r_f, t, \sigma^2)
\]

The exact Black and Scholes (1973) formula is expressed below:

\[
c = \text{Present Value of a call option} = SN(d_1) - Xe^{-r_f t} N(d_2)
\]

where:

\[
d_1 = \left[\log\left(\frac{S}{X}\right) + r_f t + \frac{\sigma^2 t}{2}\right] \cdot \frac{1}{\sigma \sqrt{t}}
\]

\[
d_2 = \left[\log\left(\frac{S}{X}\right) + r_f t - \frac{\sigma^2 t}{2}\right] \cdot \frac{1}{\sigma \sqrt{t}}
\]

\(N(d)\) is the probability that a normally distributed random variable will be less than or equal to \(d\). \(N(d_1)\) in the Black Scholes formula is the option delta\(^2\). Thus the

---

\(^1\) European options can only be exercised upon their maturity, as opposed to American options, which can be exercised at any date up to maturity.

\(^2\) Option Delta, also referred to as the hedge ratio, represents the number of shares that are needed to replicate the payoff from a call option (see Breadly and Meyers: 1981)
formula tells that the value of a call is equal to an investment of $N(d_1)$ in the common stock less borrowing of $Xe^{-r_t}N(d_2)$.

In the case of real option valuation, the parameters used in the Black Scholes Formula will have following significance:

**Figure 16: Mapping the Option Parameters on an IT Investments Decision**

**Current value of the asset ($S$):** Current value of the asset ($S$) is the present value of the future cash flows expected from the investment. The higher the current value, the higher the value of a (previously issued) option.

**Cost of exercising the option ($X$):** Cost of exercising the option ($X$): is the present value of making the investment. The higher this amount, the lower the value of the option.

**Length of option's life ($t$):** Length of option's life ($t$) is the time within which the investment can be made. As with financial options, the longer the life of the option the more flexibility there is as to when the investment is made and therefore, the more the option is worth.
Uncertainty (Volatility) of future asset price ($\sigma$): Uncertainty represents a measure of the unpredictability of future cash flows associated with the investment. The more uncertain these cash flows, the more valuable the option because the firm has full exposure to the upside but only limited exposure to the downside (the firm need not exercise the option).

Interest rate ($r_f$): The interest rate $r_f$ represents the risk free interest rate. The higher this rate, the more valuable the option. The actual valuation of options in practice has been greatly facilitated by Cox and Ross's (1976) recognition that an option can be replicated (or a "synthetic option" created) from an equivalent portfolio of traded securities. Thus, investors are independent of risk attitudes or capital market equilibrium considerations. Such risk-neutral valuation enables present-value discounting of expected future payoffs (with actual probabilities replaced with risk-neutral ones) at the risk-free interest rate, a fundamental characteristic of arbitrage-free price systems involving traded securities. Rubinstein (1976) further showed that standard option pricing formulas can be alternatively derived under risk aversion, and that the existence of continuous trading opportunities enabling a riskless hedge or risk neutrality, are not really necessary.

As can be seen from the formula, the willingness of individuals to bear risk does not affect the option value, nor does the expected return on stock. The value of the stock increases with the level of the stock price to the exercise price ($S/X$), the time to expiration times the interest ($r_f t$), and the time to expiration times the stock’s variability ($\sigma^2 t$).

5.5.3 Simplifying the calculation

The Black-Scholes option valuation formula, with all these mathematical expression, seems a little removed from the real world. However, every day dealers on the option exchanges have been using exactly this formula for years. These dealers are for the most part not trained in the formula’s mathematical derivations. Instead, they just use special computer programs or a set of tables to find the value of an option. The table (see Appendix) allows the use of the Black Scholes formula to value options in three simple steps:
Step 1: Calculating Cumulative Volatility

When deferring an investment decision, the asset value may change and affect the investment decision for the better while waiting. That possibility is very important, but naturally it is difficult to quantify because of the uncertainty as to how the future asset values will change. The most common probability-weighted measure of dispersion is variance $\sigma^2$. Variance is a summary measure of the likelihood of a random value far away from the average value. The higher the variance, the more likely it is that the values of an investment project will be either much higher or much lower than average.

How much things can change while one waits, also depends on how long one can afford to wait. For business projects, things can change a lot more waiting two years than if waiting only two months. Thus the measure of the total amount of uncertainty is variance per period $\sigma^2$ times the number of periods $t$. This is referred to as cumulative variance.

$$\text{Cumulative Variance} = \text{Variance per period} \times \text{number of periods} = \sigma^2 t$$

For mathematical convenience, instead of using the variance of project values, Luehrman (1998) suggests to use the variance of project returns, in other words to work with the percentage gained or lost per year. There is no loss of content because a project's return is completely determined by the project's value:

$$\text{Return} = \frac{\text{Future Value minus Present Value}}{\text{Present Value}}$$

The reason for doing this is that the probability distribution of possible values is usually quite asymmetric; value can increase greatly but cannot drop below zero. Returns, in contrast, can be positive or negative, sometimes symmetrically positive or negative, which makes their probability distribution easier to work with.

Second, it helps to express uncertainty in terms of standard deviation $\sigma$, rather than variance $\sigma^2$. Standard deviation, which is simply the square root of variance, expresses just as much about uncertainty as variance does, but it has the advantage of being denominated in the same units as the investment project being measured.
Multiplying the standard deviation $\sigma$ of the proportionate changes in the asset’s value by the square root of time $\sqrt{t}$ to the option’s expiration gives the so-called cumulative volatility:

$$\text{Cumulative Volatility} = \text{Standard Deviation} \times \sqrt{\text{time}} = \sigma \sqrt{t}$$

Cumulative Volatility is the value on the vertical axis of the Black Scholes table of option values as percent of share price (Table in Appendix).

**Step 2 : Calculating NPVq**

Luehrman (1998) furthermore describes the ratio of the asset value $S$ to the present value of the option’s exercise price $X$ as the $NPV_q$. He simply converts the difference:

$$NPV = S - PV(X)$$

to a ratio:

$$NPV_q = S \div PV(X)$$

As a result, absolute values are converted to decimals between zero and one. He uses the $q^1$ to reminds us that we are expressing the relationship between cost and value as a quotient:

$$NPV_q = \frac{\text{Asset Value}}{\text{PV of exercise price}} = \frac{S}{PV(X)}$$

$NPV_q$ is the value on the horizontal axis of the Black Scholes Table.

**Step 3: Finding Option Value in the Table**

Let us assume a four year call option on a stock with standard deviation 0.40, interest rate of 12.47%, asset value $140$, and exercise price $160$.

---

^1 q as an equivalent of Tobin’s $q$. 
Step 1: Cumulative Volatility \( \sigma \sqrt{t} = .40 \times \sqrt{4} = .80 \)

Step 2: \( NPV_{q} = S ÷ PV(X) = 140 ÷ [160 ÷ (1.1247)^4] = 1.4 \)

The only thing that one needs to do is calculate the \( NPV_{q} \) and cumulative volatility according to steps 1 and 2 as indicated above, and look up the corresponding value in the Black Scholes Table:

Step 3: corresponding value for \( NPV_{q} \) and Cumulative Volatility in table = 431%

Looking up the corresponding entry in the table in Appendix 1, the option is worth 43.1 percent of the stock price, or $60.34 in absolute terms.

5.5.4 Capturing Flexibility

The major advantage of adopting a real option approach to a project appraisal is that real options can capture and measure the value of maintaining flexibility when making investment decisions.

A series of papers gave a boost to the real options literature by focusing on valuing quantitatively, in many cases, deriving analytic, closed form solutions. A variety of real options scenarios have been analyzed in isolation. These options provide investment decision flexibility and can arise in a number of forms.

**Timing option (option to defer):** A timing option arises when an investment can be delayed or when a project can be temporarily shut down after its commencement. A timing option can create value because by waiting, a firm can obtain more information about likely market events before committing to a course of action. Deferral is most attractive when uncertainty is great and immediate project cash flows, which are lost or postponed by waiting, are small.

**Growth option:** A growth option arises when an investment can be scaled up if new, favorable market information emerges. Companies often cite so-called strategic values when taking negative NPV projects. This is not necessary. In many cases a close look at the projects payoffs reveals a call option on follow-on projects in addition to the immediate project cash flows, thus the downside risk can be limited.
**Abandonment option:** An abandonment option arises when a project can be shut down before the original, expected end of its life. The ability to terminate an investment places a stop-loss on a firm's exposure to downside losses. This option equals a put option, where the exercise price is the value of the project's assets if sold or shifted to a more valuable use.

**Switching option:** A switching option arises when a firm has the ability to switch the inputs to (or the outputs from) a process. These options are particularly valuable in the motor, consumer electronics or toy sectors where competitor actions and market demand are hard to forecast.

**Multiple Real Options:** Despite its enormous theoretical contribution, the focus of the earlier literature on valuing individual real options (i.e., one type of option at a time) has nevertheless limited its practical value. Many investment projects will have one or more of the above types of flexibility and thus will have an associated option value. Real-life projects are often more complex in that they involve a collection of multiple real options whose values may interact (Trigeorgis 1993). Trigeorgis (1991) presents the valuation of options to defer, abandon, contract or expand investment, and switch use in the context of a generic investment, first with each option in isolation and later in combination. He shows, for example, that the incremental value of an additional option, in the presence of other options, is generally less than its value in isolation and declines as more options are present. More generally, he identifies situations where option interactions can be small or large and negative as well as positive.

### 5.6. Practical Application of Real Options

Some options occur naturally in a project (a research and development project is naturally a staged project), whereas other options can sometimes be *built in* at extra cost. As an example of the latter point, suppose a manager is considering an investment in a new plant. One possibility is to build a fully automated plant (high fixed costs), another possibility is to build a less automated plant (lower fixed costs). The highly automated plant has a higher capacity and thus provides an option to expand output in the future if demand is greater than expected. A NPV analysis would tend to assign a higher discount factor to the first option, as it is higher risk due to its higher break-even point. However, the higher operating
leverage of the fully automated plant means that its future cash flows are more volatile. It is more profitable than the less automated plant as output is scaled up and less profitable as output is scaled back. Thus, the fully automated plant has a higher option value. The net result is that, without calculating the detailed option value of both investments, it is not possible to state categorically which investment is better.

5.6.1 Disadvantages of Real Option Models

The practical problem of real option models is the complex calculation itself as well as the uncertain future of highly strategic investments, which makes parameters hard to estimate. The term of real option, unlike that of financial options, is usually open-ended or indefinable. The volatility of the underlying asset can be difficult to measure or guess, especially since it is not always clear what it is if, for example, it is yet to be invented. How can one define the appropriate benchmark asset class in the case of a new drug for a rare disease? And there may be additional variables to consider, such as the strategic benefit of pre-empting a rival.

In the more complex real-life option situations, such as those involving multiple interacting real options, analytic solutions may not exist and one may not even be always able to write down the set of partial differential equations describing the underlying stochastic processes. However, the ability to value such complex option situations has been enhanced by computing technology.

5.6.2 Comparison of Real Options with Net Present Value

In large measure, the difficulties that arise in applying NPV techniques are due to the fact that unlike bond valuation, the investment of many strategic IT investments will not generate certain future cash flows and managers can take future decisions, which will alter those cash flows. Apart from these inherent difficulties in forecasting future cash flows, applications of NPV often involve the use of questionable risk-adjusted discount rates.

In contrast with real options, DCF valuation methodologies as typically applied only explicitly consider two of the five drivers in valuing an investment opportunity, the current value of assets $S$ and the cost of exercising the option $X$, and thus fail to adequately capture the value of flexibility.
Getting the cash flow projections right (or even close) is staggeringly difficult. But it is even trickier to choose the correct discount rate. Conceptually, that rate is the opportunity cost of not investing in another project of similar systematic risk\(^1\). So the higher a project's risk, the higher its discount rate and the lower its NPV. The CAPM (Capital Asset Pricing Model) often spits out negative NPVs for many of the most exciting strategic opportunities.

The main reason for this shortcoming is that the model can use only information that is already known. For highly strategic investments that information is typically not much, and the resulting uncertainty tends to be reflected in an excessive discount rate. Combining an NPV calculation with decision trees (which assign numerical probabilities to various possible outcomes) may help, but not much. For each branch of the tree, the analyst still has to pick and apply an appropriate discount rate, and that of course was the problem in the first place. More fundamentally, the flaw in the CAPM is that it implicitly assumes that when firms buy new assets, they hold these passively for the life of the project. But they do not. Instead, they employ managers precisely in order to react to events as they unfold. Obviously, this managerial flexibility must be worth something.

As with financial options, the longer the option lasts before it expires and the more volatile the price of the underlying asset the more the option is worth. This is in sharp contrast to the NPV, which deals harshly with both long time horizons and uncertainty.

Considering a proposed investment in an R&D project it is likely that managers in most firms would deem this project high risk. As a result, they may decide to use a high discount rate (relative to the discount rate they would apply to a cost reduction project) in evaluating the cash flows of the project. This approach does not reflect what happens in reality. The level of risk faced by the project will change over time as new information emerges. After, say one year of the project, a better assessment of its likely success can be made and thus, the project would face lower risk at this point as its future cash flows will be more predictable. A correct application of NPV will recognize this and will apply differing discount rates

---

\(^1\) Systematic risk is referred to risk that, in a large portfolio, cannot be diversified away.
over the life of the project. Although a correctly implemented decision tree NPV analysis can overcome the above problem, there is good evidence that companies rarely make the appropriate adjustments to their discount rates. The error of using a single (high) discount rate worsens over time due to the compounding nature of NPV.

Therefore NPV analyses generally penalize projects, which have a high level of uncertainty surrounding their future cash flows by applying higher discount rates. However, one of the lessons of option theory is that uncertainty is an option's friend rather than its enemy. Unlike NPV, option pricing recognizes that the possible gains and losses from an investment are not symmetrical. An investment will usually have a fixed downside. For example, if an investment is a staged investment, the firm can abandon the investment before completion if market conditions change. Conversely, if market conditions are better than originally expected, the firm may be able to scale up the investment. Option pricing theory explicitly acknowledges that flexibility to respond to changing conditions has a value and, unlike NPV, attempts to quantify that value. Thus, the value of an investment can be considered the composite of two components:

\[
\text{Expanded NPV} = \text{Static (passive) NPV} + \text{value of option(s)}
\]

The static NPV is the NPV of the cash flows, which the project is expected to generate when the benefits of flexibility are ignored. The value of the option is the additional value generated because of the opportunities, which could result from the project's flexibility. From the above formula, it is apparent that ignoring the option value when evaluating a project may incorrectly lead to a reject decision.

### 5.6.3 When to use Real Option Pricing Techniques

Before attempting to apply real option valuation techniques to an investment project, it would be useful to consider whether the project is likely to have significant option value. In general, the project should be sizeable, strategic in nature and the required investment should not consist of an up-front, irrecoverable cost. If a standard (correctly performed) NPV analysis shows that the project should proceed, a detailed valuation of the real option component may not be
required. However, if the investment appears marginal under a standard NPV analysis, then a detailed real option valuation may be justified.

According to Dixit and Pindyck (1994) two things are needed to introduce an opportunity cost into the NPV calculation: irreversibility and the ability to invest in the future as an alternative to investing today. The less time there is to delay, and the greater the cost of delaying, the less will irreversibility affect the investment decision.

In spite of the claims of advocates of real option theory, it is undoubtedly the case that the valuation of real options is not always an easy task. It may be that in certain cases, whilst an investment clearly has an option value, we are unable to calculate it precisely due to its complexity. Of course, it’s hardly better to rely on a DCF measure which ignores the value of flexibility simply because it is easier to calculate.

Despite the potential difficulties in calculating the value of a real option, the theory does provide a number of insights for managers. Thinking about investment projects in option terms encourages managers to decompose an investment into its component options and risks. This can lead to valuable insights. Where does the uncertainty come from? Perhaps the uncertainty is resolved over time (What will the Euro/US$ exchange rate be next year?) or perhaps it will only be resolved by making the investment (Is the project technically feasible?). Thinking of the investment in option terms will also encourage managers to consider how best to enhance the value of the investment by building in more flexibility.

Options Pricing Models are too complex to be worthwhile for minor decisions. And they are not useful for projects that require a full commitment right away, since the value of an option lies in the ability to spend a little now and decide later whether to forge ahead.

5.6.4 Are Option Valuation Methods in Use?

The last two decades have seen substantial improvements in both our understanding of options pricing and in the desktop computing power available to managers. The net result of these events is that our ability to value the real option
component of investment decisions has increased dramatically. Thus it is becoming possible to determine how much it is worth paying for an option.

But new ideas take time to diffuse. As indicated above, surveys show that in general practice financial managers still use the traditional capital budgeting methods for investment under certainty. Even though the theory on option valuation models is over 20 years old, it is typically not covered in the syllabi of professional accounting examinations and consequently, the concepts of real options are not yet widely known. Even once the knowledge barrier is overcome, the valuation of real options is often a non-trivial task. It is certainly more complex than the calculation of DCFs. On first sight, the relevant formula appears intimidating and it can be difficult to obtain the values of certain parameters of the option-pricing model. In particular, the calculation of an estimate for volatility is much more difficult for real options than for financial options.

Experts had developed rules of thumb that simplify the formidable math behind options valuation, while making real options applicable in a broader range of situations. And consulting firms are aware of this technique as the Next Big Thing to sell clients.

5.7. Summary and Outlook

In the first chapter it was shown, that from empirical research it is hard to prove the correlation between IT investment and company performance. In most instances it is very difficult to claim that the technology actually caused improvements on company performance. In organizational studies, this problem is referred to the casual ambiguity. However, academics as well as practitioners in the latest years have argued that traditional financial methods used are inappropriate for measuring the strategic value of investment projects with multiple sources of uncertainty, which in particular applies to many IT projects.

A traditional calculation of net present value discounts projected costs and revenues and examines the project as a whole and concludes it is a no-go, in case the NPV is negative. The real-options analysis will break the same project into stages and may conclude that it makes sense to fund at least the first stage. Real options analysis rewards flexibility, and that is what makes it more appropriate for
strategic investments with high levels of uncertainty, than today's standard decision-making tool, the NPV method.

In addition to the above, real option theory points out that in contrast with what our intuition would suggest, uncertainty is not necessarily a bad thing. The more uncertain the environment facing an investment project, the greater the value of the option component of an investment.

Researches of IT managers have also shown that IT practitioner have problems when determining model parameters like the time series of costs and revenues of an IT project, or the appropriate discount rate for the NPV model. “Consequently, simple, static models are often preferred to the NPV model, and important project decisions are based rather on intuition, experience and rule of thumb than on quantitative analysis.” (Tam 1992 cited in Taudes, Feuerstein, Mild 1999).

Due to the complexity of option pricing models, although conceived more than 20 years ago, they are just now coming into use. So intuition and simple static methods is making way for OPM, but it will be long before real option valuation comes into wide use.

6. XML Investment Decision as a Deferral Option

As outlined in the previous chapters, XML is a new evolving industry standard supported by many companies in the industry as a result of increasing demand for more structure and flexibility in the exchange of business data over the Internet. In general, a widespread adoption of a new technology like XML, even when it has obvious advantages, often takes a very long time. There is a wide gap in many fields, between what is known and what is actually put in use. According to Rogers (1983) technological innovations require a lengthy period, often of some years, from the time they become available, to the time where they are widely adopted, because they create uncertainty in the minds of potential adopters about the expected consequences.

In the case of XML/EDI in particular it is hard to predict future developments, as regards to how far XML/EDI will be accepted as standard, what impact it will have on the way of electronically doing business, and when it will reach a critical mass
of companies deploying it. The question that companies are facing today, is not whether to invest in XML/EDI projects, but rather when and to what extent.

As there is a big uncertainty tied to the answer, for a proactive company the wait-and-see strategy might not be the solution. If managers want to avoid making decisions intuitively, the only way to answer this question is to try and assess the necessary IT investment of this strategic project.

The parameters of this investment decision are so complex and hard to estimate, the DCF method would have to use such a high discount rate, that the project would very likely be underestimated and thus rejected. This chapter will try to set up a framework based on the idea of the option pricing model to enable companies to assess whether and when and how to invest into XML projects.

It is of particular importance for this investment decision to understand the impact of widespread XML/EDI adoption. The success of XML/EDI depends on the number of companies using this new standard, as it increases efficiency on an intra-organizational level. Thus, the scope of the framework introduced in this chapter includes SMEs as well as large organizations.

### 6.1. General Assumptions

This paper is based on the assumption that every company today facing the XML/EDI investment decision is the holder of an American deferral option on a dividend-paying asset. The asset underlying this option is the potential stream of revenues from an investment opportunity that will materialize only when a company decides to invest in an XML/EDI project at any given future date, where the dividends are the revenues lost during deferral time. The projects under consideration are (i) **XML/EDI based purchasing** and (ii) **XML/EDI based sales**. It will be concluded that in some cases it does make a difference for a company to consider them as two different projects, due to the differences in bargaining power up and down the supply chain.

The option value in this case is embedded in the possibility for any firm today to resolve some of the uncertainties concerning the adoption of the XML/EDI standard. A firm could passively observe the standard evolving and being applied
in the industry, and could actively try to lower the risk of expected revenues by preparing for an XML/EDI investment project at a future time.

This deferral option is obtained at no direct cost. Dixit and Pindyck (1994) believe that a company could obtain a deferral option at no cost if it faces no credible competitive threat of loosing the deferred investment opportunity. In the case of monopoly the only cost of deferring is the opportunity cost of delaying the entry. In cases other than monopoly, things are somewhat different. Full benefits of the option, exists for the market leader among two or more competitors. In the case of no market leader, all firms have the option, but only the first mover will enjoy full benefits. Benaroch and Kauffman (1999) believe that this is the case for duopoly, I would rather extend it to any market structure other than monopoly.

![Figure 17: Mapping the Option Parameters on XM/EDI Investments Decision](image)

The value of the underlying assets $S$ will be the present value of an XML/EDI project invested in, when and if the company exercises the option. The exercise price $X$ will be the expenditures required to invest into the XML/EDI project. The time to expiration $t$ is not yet known, as it is a deferral option, but a timeframe of 5 years is considered (Experts predict that XML/EDI will penetrate the B2B world
within the next 3 to 5 years). The five-year risk-free rate of interest $r_f$ is about 7%.

Finally $\sigma$ is the standard deviation of returns on the XML Project, where 3 different sources of the project value uncertainty have been recognized.

### 6.2. Going into Details

This part will have a detailed look on each of the 5 option parameters for an XML/EDI investment decision.

#### 6.2.1 Project Uncertainty ($\sigma^2$)

Data required for option pricing models is difficult to estimate. This especially applies to the estimation of the variance in the expected return of an IT investment. In the case of a financial option, the variance is the variability of the underlying stock, which the option trader can obtain by option valuation. But how is a company going to estimate the variability of future project cash flows of an IT investment?

Taking a closer look one can say that the future cash flows of an XML/EDI project are influenced by different sources of risk, where the sources of the project value uncertainty have been recognized as (1) market uncertainty $r_M$, (2) technical (project) uncertainty $r_P$, (3) uncertainty of conversion effectiveness $r_{CE}$. These sources of uncertainty are assumed not to be correlated with each other. When $\sigma^2(S)$ is the variance of the XML project, then:

<table>
<thead>
<tr>
<th>Variance of XML Project $\sigma^2(S) =$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f[ $</td>
</tr>
<tr>
<td>Market Variance $\sigma^2(r_M)$,</td>
</tr>
<tr>
<td>Project Variance $\sigma^2(r_P)$,</td>
</tr>
<tr>
<td>Variance of Conversion Effectiveness $\sigma^2(r_{CE})$</td>
</tr>
<tr>
<td>$]$</td>
</tr>
</tbody>
</table>

or: $\sigma^2(S) = f[\sigma^2(r_M), \sigma^2(r_P), \sigma^2(r_{CE})]$
where $r_M$ represents the market uncertainty contributing to the uncertainty of the future cash flows of XML/EDI Project $S$, and $\sigma^2 (r_M)$ represents direct contribution of $r_M$ to the variance of $S$. The market variance $r_M$ is a function of:

$$\sigma^2 (r_M) = f(BPs, BPc, C)$$

$BPc$ is a parameter representing the bargaining power of a company vis-à-vis its customers. It can take any value between 0 and 1, with 1 indicating a low bargaining power. This would be the case for a company, which is relatively small compared to its customers. A very good example is the automotive industry, where the suppliers are big companies themselves, but they have a low bargaining power compared to their customers, the even bigger international car manufacturers.

When a company today decides whether to invest in XML/EDI or not, a standard with a lot of potential but yet to be adopted, it has to consider its bargaining power vis-à-vis its customers. If the customer(s) decide(s) not to invest in XML/EDI and happen to have higher bargaining power, the company is taking a big risk of occurring sunk costs.

$BPs$ stands for the bargaining power vis-à-vis a company’s suppliers. The same values and logic, vice versa, apply to this parameter.

$MP$ is the parameter describing the market position. A company with a small market share and/or a lot of competitors depends very much of the technical standards that its competitors might set, or implement. If its competitor, the market leader, decides not to invest in XML/EDI, it will be hard for the competitor with the smaller market share to convince its trading partner of the importance of this new technology.
$r_p$ represents the project uncertainty contributing to the uncertainty of $S$, thus $\sigma^2(r_p)$ is the direct contribution of $r_p$ to the variance of $S$.

\[
\sigma^2(r_p) = f(PM, \text{Comp})
\]

\[
PM = f(\text{in time, in budget, meet technical targets})
\]

\[
\text{Comp} = f(n, \text{No.of multilateral agreements})
\]

$PM$ is the parameter assessing the project management, which is determined by three aspects: whether a project will be completed (1) in time, (2) in budget, and (3) whether it will meet technical targets. A late project delays the benefits a company is expecting from investment. In the worst case, the project is too late to meet a competitive challenge or a deadline and ends up being cancelled. An over-budget reduces the return that one expected when deciding to make an investment in information technology. A small budget overrun is probably not a problem, but anything significant could dramatically alter the expected return on investment (Lucas, 1999: 31). Last but not least, when project management fails technical project targets, there are no monetary benefits to expect. These three aspects are highly correlated with each other. Usually when a project is in overrun, the costs will be higher, or if the project fails technical targets in time, project management will assign more people to it and extend project duration, all causing the project to be over time and over budget. For better assessment of this parameter it would be helpful to have statistics on how late projects are on average, and how much they are over budget.
Forecasting IT projects is very hard due to their complexity. While not excusing cost and time overruns, the prudent manager should factor their likehood into any return on investment calculations (Lucas, 1999: 32).

$\text{Comp}$ refers to the complexity of the XML/EDI project. The complexity itself depends on the number of transactions $n$ supported and the number of business partners. The higher $n$ and the more multilateral agreements the new standard must meet, thus the higher $\text{Comp}$.

\[
\begin{array}{c|c|c}
\text{Low } \text{Comp} & \text{High } \text{Comp} \\
0 & 0.5 & 1
\end{array}
\]

According to Weill (1990) not every IT project is successful because management does not always take advantage of the opportunities that such an investment provides. Strassmann (1985:80) also sees the critical parameter in management's ability to use the new system to extract economic benefits this critical variable is referred to as conversion effectiveness.

Conversion effectiveness measures the ability of an organization to convert its IT investments into working applications. $r_{CE}$ is the uncertainty of conversion effectiveness contributing to the uncertainty of $S$, thus $\sigma^2 (r_{CE})$ being the direct contribution of $r_{CE}$ to the variance of $S$.

Lucas (1999) stated many variables that determine conversion effectiveness. He believes that a failure on any one of the parameters listed below can doom the projects, even if every other aspect of development is successful. The variables he quotes are: (1) Size and scope of project, (2) Amount of unknown technology involved, (3) Project management, (4) Support and encouragement of managers, sponsorship, (5) The urgency of the problem/opportunity addressed by the technology, (6) Norms in the organization, (7) User commitment and involvement, (8) Technical development environment, (9) Quality IT staff, (10) Strength of project team, (11) Level of expertise of participants, (12) Type of technology employed, (13) Type of application, (14) Amount of custom code written, (15)
Lucas' extensive list covers some parameters (1, 3, 10, and 14), which have been already used in the framework introduced in this paper, to determine project uncertainty \( r_P \). Some other parameters he cited (18, 19, 20, 21, and 22) determine conversion effectiveness in the sense presented here. Conversion effectiveness in this paper is described as functions of management commitment, \( MC \), and the ability of an organization to cope with transformation, \( ACT \).

\[
\sigma^2(r_{CE}) = f(MC, ACT)
\]

\( MC \) represents management’s commitment when integrating new IT Systems. The idea behind this parameter is that without management commitment to an IT project in general, and XML/EDI project in particular, conversion effectiveness will be low. “The absence as a significant productivity effect from IT therefore must be seen […] as a failure by managers to build organizations that effectively integrated IT with business strategy, human resource management, and efficient resource allocation” (Lovemenann 1994, cited in Lucas 1999: 52). The higher \( MC \), the higher conversion effectiveness, thus the lower overall project variance. \( MC \) covers every aspect of a manager’s task from awareness creation with end-users to managing the required business process redesign.

Installing Information technology involves much greater expense than just buying the hardware and software. Extensive human resources and business process redesign is required to support an IT System before it can be productive to use. Very often, the most costly elements of such support are not technological but
organizational. This fact has far reaching implications for the ways that information technology investments should be managed (Strassmann 1985: 79). The higher MC the lower the risk of conversion effectiveness, thus the lower project variance.

\[
\begin{array}{c|c|c|c}
\text{Low } ACT & & \text{High } ACT \\
0 & 0.5 & 1
\end{array}
\]

ACT is the ability of organization to cope with transformation. A company with high ACT is a flexible organization, possibly one which is used to organizational change due to frequent implementations of new Information systems. An organization with cannot easily cope with organizational change will have a very low ACT. ACT deals with the average ability of all personnel of an organization to cope with change. It might be that management commitment is very high, and even if responsible managers have good awareness creating skills, they will have a very hard time with a company with low ACT. Thus, there is a correlation between ACT and MC. A company usually has low ACT if MC for IT Projects in the past has been low. Vice versa the probability that MC is high might depend on past levels of ACT. The lower ACT, the lower conversion effectiveness, thus a higher project variance.

6.2.2 Project Returns (S)

For IT investments that offer a direct return by definition, estimates are not too difficult. When looking at a case of indirect returns, they are much more difficult to define and measure. What is the estimated return for a system that is a competitive necessity? What is the estimated return for technology when it is the only way to do the business? Some experts suggest that one should look at the cost of not investing.

Looking at the XML/EDI project’s returns S, it is known from general IT investments that three types of added value can be defined: (1) substitutional values \( U_{sub} \), (2) complementary values \( U_{comp} \), and (3) strategic values of IT investments \( U_{str} \). Furthermore the present value of the XML Projects returns are
determined by the time value of money \( r \) and the time \( t \) they will occur. Thus \( S \) can be expressed as a function of:

\[
S = f(U_{sub}, U_{comp}, U_{strat}, r)
\]

The substitutive value \( U_{sub} \) is defined as the amount of fixed costs per period saved \( s_f \), and the annual savings of variable costs, \( n \) times \( s \). Where \( n \) is the number of transaction per period, and \( s \) is the amount of money saved per transaction.

\[
U_{sub} = (n \times s) + s_f
\]

Complementary value \( U_{comp} \) is calculated by the hedonic value of the new automation system. The hedonic wage model analyzes the monetary effects of office automation. Unlike large data processing systems whose values lie in their capacity to substitute computer power for routine labor, “office automation is a complementary technology” (Sassone and Schwartz 1986: 83).

\[
U_{comp} = \text{hedonic value of IT}
\]

Office automation tools embody two kinds of benefits: (1) increased efficiency and (2) increased effectiveness. Increased efficiency is referred to shorter time needed to accomplish a given task, or allow more of a given task to be done in the same amount of time. Increased effectiveness results from increased efficiency, which allows restructuring of work. Managers and professionals can spend more time performing activities reflected in their titles and time rather that to support, clerical and non-productive (lower-productive) activities.

Quantifying this shift in working profiles uses a two-part procedure. First, a method called the Work Profile Analysis identifies the working profiles of the employees affected by the introduction of a new technology. The results thereof are then built into a computer simulation called the Hedonic Wage Model, which is basically a set of multiple linear equations.

The whole model is based on the assumption, that jobs are not monolithic but have identifiable components with different implicit values. Furthermore it is assumed that workers are worth what they cost in company: salaries, fringe
benefits and direct overheads. Their worth can be thought of as the weighted values of their activities that they perform. The weights are the percentages in the Work Profile Matrix. The values of the initial activities are unknown, but are determined through the hedonic wage model. By using the hedonic model to explain the value of the job in terms of the amounts of each component the job entails, interferences about the implicit values of each component can be drawn (Sassone and Schwartz 1986:84).

In cases where a new IT Systems (in particular an XML investment) may induce changes in the composition of a job or work profile, the hedonic values can be used to place a monetary value on that change ($U_{comp}$), and hence on the information system itself.

The strategic return $U_{strat}$ results from reducing the price elasticity of demand. In other words, a company which can provide better service quality to its customers, and/or create a better relationship with its business partners (through integration of supply-chain, or joint DSS, etc.) will enjoy higher customer satisfaction and thus a higher market share in the long run (all provided that the competitors do not succeed with a similar strategy. In this case not implementation of a novel technology would mean the loss of market share). In general, the greater the distance in business process between where the investment occurs and where benefits appear, the more likely that the benefits are indirect (or strategic) (Lucas, 1999: 82). When IT acts as a substitute for labor, a cost-benefit methodology is quite appropriate. However, when IT enhances the quality of a product by reducing cycling time or by adding value through services of information content, economic methods of analysis encounter problems (Lucas 1999: 51).

### 6.2.3 Project Costs ($X$)

The present value expenditures required to implement the XML Project are determined by the initial investment outlay $I$, the periodical fixed costs $FC$, the variable costs per transaction $VC$, the number of transactions per period $n$, the time value of money $r$, and the time $t$ they will occur.

$$X = f(I, FC, VC, n, r, t)$$
The initial investment outlay $I$ is determined by the costs of hardware $HW$, software $SW$ (also includes in-house software development), external consulting $Cons$, internal trainings $Train$, and the software platform $Platform$. 

$$I = f(HW, SW, Cons, Train, Platform)$$

Software costs depend on the costs for standardized XML/EDI software, $SW$, and the costs of development of in-house software, $Devel$. The costs of software will be lower, if a wide variety of appropriate standardized software for XML/EDI exists, as the costs of standardized software are generally lower than in-house development.

$$SW = f(existing\ standard\ SW, Devel)$$

For an IT investment, one must estimate the costs required to develop a project at the expiration date of the option, when you might undertake the investment. For an option three years in future, the analyst has to estimate how much investment would be required. Many parameters influence this cost, including the pace of technological change. In 3 years a package might come along like SAP that dramatically changes the cost of an investment (Lucas 1999: 169). The same applies to the costs of hardware. Who knows what how hardware prices will develop in the next few years?

$$HW = f(n, UHC, Platform)$$

In any case the costs of hardware is a function of the number of transactions supported by the XML/EDI project, $n$, unit hardware costs $UHC$, as well as the software platform, $Platform$. The amount of hardware capacity required very much depends on the software platform used.

### 6.2.4 Time Value of Money

When talking about the time value of money, two different interest rates have to be considered in a real option-pricing framework: (1) The rate $r$, at which the future cash flows will be discounted, which considers the risk characteristics of the investment project; and (2) The 5 year risk-free rate of interest, $r_f$, to calculate the value of the option.
6.2.5 Time to expiration ($t$)

As we are looking at a deferral option, the time to expiration $t$ is the parameter to be determined. A timeframe of 5 years is being considered, as XML experts predict that XML will penetrate the business-to-business world within the next 3 to 5 years. Furthermore it is almost impossible to make any realistic calculations for cash flows beyond a timeframe of 5 years.

6.2.6 Summary

Although waiting to invest into XML/EDI costs the firm more in terms of forgone profits, it may save money on the cost of implementing the new standard. Thus the firm must weight the costs and benefits of delaying adoption, as well as take account of the rivals’ strategic behavior. If this investment decision can be delayed, option-pricing valuation would be most appropriate to use.

In this section the option pricing formula and its parameters were analyzed in detail. This framework should help each individual firm to put the XML/EDI investment decision into real numbers and make an individual adoption decision. To help visualize certain parameters of this complex formula the next section will introduce four company profiles derived from a technology diffusion matrix, and map the parameters of the Black-Scholes formula to each single company profile.

6.3. Company Profiles

The previous section may seem confusing, due to the complexity of parameters used for the XML/EDI Investment decision framework. In order to help visualize this framework, four company profiles will be introduced, that fit into a two times two matrix, with market power defining one axis, and the timing of XML adoption the other. The underlying idea behind this matrix is the theory of technology diffusion.

6.3.1 Determining Diffusion

According to Rogers (1983:3) “the diffusion of an innovation is a process, in which the innovation spreads through certain channels in the social system (target population) in time. The diffusion (of a new technology) is the evolutionary process
of replacement of an old technology for solving similar problems or accomplishing similar objectives.”

Many researchers in the past decades have committed their work to the subject of the diffusion of innovations in general, and the diffusion of new technology in particular. On a macro level they studied the pattern of diffusion over time, trying to (a) classify the type of adopters of a new technology, and (b) identify parameters that determine the difference in diffusion patterns. One of these parameters is market structure, and consequently the individual market power of a firm.

In the diffusion matrix introduced later in this chapter, the types of adopters define one axis, and market power the other.

6.3.2 Type of Adopters

Mansfield (1961) was one of the first to investigate basic models for technology diffusion among companies. He analyzed the factors determining how rapidly the use of new techniques spread from one firm to another. He developed a model explaining differences among innovations in the rate of imitation. Analyzing the rate of imitation is particularly important, as “an innovation will not have its full economic impact until the imitation process is well under way” (1961: 762).

Rogers (1962) classifies adopters of innovation into 5 categories: (1) Innovators are hazardous, rash, daring, and risky. (2) Early Adopters have the role of decreasing uncertainty about an innovation. (3) Early Majority adopts new ideas just before the average member of the social system. They follow with deliberate willingness in adopting innovations, but seldom lead. (4) Late majority adopt new ideas just after the average member of a social system. Adoption may be both economic necessity and the answer of increasing network pressure. Innovations are approached with a skeptical and cautious air. Their relatively scarce resources mean that almost all of the uncertainty about a new idea must be removed before the late majority feels that it is safe to adopt. (5) Laggards are the last in the social system to adopt an innovation. When laggards finally adopt an innovation, it may already have been superseded by another more recent innovation that is being already used by innovators. They are traditionally oriented, scarce economic position, thus very precautious.
Bass (1969) developed a growth model for the timing of initial purchase of new products and empirically tested it against data of eleven consumer durables. The basic assumption of his model is that the probability of purchase at any time is related linearly to the number of previous buyers. The model implies exponential growth of initial purchases to a peak and then exponential decay. Innovative and imitative behavior is the underlying behavioral rationale for the model. According to Bass Innovators are individuals who decide to adopt an innovation independently of the decision of other individuals in a social system. Imitators are classified into (1) Early adopters, (2) Early majority, (3) Late adopters, and (4) Laggards.

Fisher and Pry (1971) developed a mathematical form of the diffusion model, which fit existing data in a wide variety of case studies. The Fisher-Pry model is based on the imitation effect, where the adoption decision is based on the experience of the early adopters. More advanced models as the Sharif-Kabir Model introduces the delay coefficient, which compares the delay of the adoption compared to the most optimistic estimate.

One idea all these theories have in common is the assumption of early adopters and late adopter of innovation. They might apply different terminology, like imitator and innovator. The terminology used in the diffusion matrix presented in this thesis is defined as follows:

Late Adopter is a company that prefers to apply new technologies later than average. “Wait and see” strategy is applied to gain more information and to avoid the risk of sunk costs. Management or owners might even be technology averse.

Early Adopters are companies that try to use new information technology before everybody else does in order to gain a competitive advantage, thus being an early adopter is the general company strategy. In some cases technology freaks sit in key positions forcing early adoption, possibly without any particular strategic motivation.

6.3.3 Parameters that Influence Diffusion

Mansfields (1961) model builds largely around one hypothesis – “that the probability that a firm will introduce a new technique is an increasing function of the proportion of firms already using it and the profitability of doing so, but a
decreasing function of the size of investment required” (1961: 762-763). His studies showed that the rate of imitation tended to be faster for innovations that were more profitable and that required relatively small investments. Several other factors showed very low statistical significance: There was some apparent tendency for the rate of imitation to be higher when the innovation did not replace very durable equipment, and when the firms’ output was growing rapidly, and when the innovation was introduced in the more recent past.

Rogers (1962) explains the different rates of adoption by introducing 5 characteristics of Innovation: (1) *Relative advantage*: is the degree to which an innovation is perceived as better than the idea it supersedes. (as opposed to objective advantage) (the greater strategic advantage the faster it will be adopted); (2) *Compatibility*: is the degree to which an innovation is perceived as being consistent with the existing values, past experiences, and needs of potential adopters. ( if not compatible, not adopted as fast); (3) *Complexity*: is the degree to which an innovation is perceived as difficult to understand and use. ( the simpler the faster the adoption); (4) *Triability*: is the degree to which an innovation may be experimented with on a limited basis (triability-> less uncertainty -> faster adoption); (5) *Observability*: is the degree to which the result of an innovation are visible to others ( the more visible, the more likely that they will adopt.

Bass 1969: 215) assumes that apart from innovators, imitators are influenced in the timing of adoption by the pressure of the social system. Although his empirical work exclusively dealt with consumer durables, Bass intended to apply his theory to the growth of initial purchase of a broad range of distinctive “new generic classes of products”.

*Fisher and Pry* (1971) developed a substitutional model of technological change based on three assumptions: (a) many technological advances can be considered as competitive substitutions of one method of satisfying a need for another. (b) If a substitution has progressed as far as a few percent, it will proceed to completion. (c) The fractional rate of fractional substitution of new for old is proportional to the remaining amount of the old left to be substituted.
6.3.4 Market Power

Market Power is a particularly important factor of the XML/EDI investment decision as the benefits of this investment project depend on the diffusion of this new standard. As opposed to other IT investment projects that only affect efficiency of internal business processes, the success of XML/EDI investment depends on the number of business partners and competitors adapting the new standard.

The first part of this section will give an insight into existing research on the impact of market structure on diffusion. Based on existing theory the definition for Market Power for the diffusion matrix presented later will be derived.

Mansfield (1961) identified inter-industry differences in the rate of imitation (although with a small sample), with the result that the rate of imitation is faster in more competitive industries.

Reignanum (1981b) finds that greater concentration in the user industry increases the diffusion rate. She concludes that diffusion is entirely a result of strategic behavior. Information is assumed to be perfect and firms identical. Since the conclusions are based on a limited example, the author suggests they be accepted only as provisional.

These results are based on a previous paper, Reignanum (1981a), where she assumes that firms are operating at Nash Equilibrium\(^1\) output levels, generating a market price and profit allocations. When a cost-reducing innovation is announced, each firm must determine when (if ever) to adopt it, based in part upon the behavior of the rival firm. If either firm adopts before the other, it can expect to make substantial profits on the expense of the other firm. On the other hand the discounted sum of purchase price and adjustment costs may decline with the lengthening of the adjustment period as various quasi-fixed factors become more easily variable. Reignanum analyzes the case of identical firms, and non-identical firms. She shows that there exist two asymmetric\(^1\) Nash equilibria in pure strategies. In both cases, at Nash equilibrium, one firm will adopt the innovation at

\(^1\) A Nash equilibrium is a situation in which each firm is doing the best it can given the behaviour of its rivals (applied to Oligopoly and Duoploy situations, where interdependence of market price and output levels between market players is big).
a relatively early date, the other relatively later. Thus even in the case of identical firms and complete certainty, there is a diffusion of innovation over time.

Quirmbach (1986) shows that the diffusion of capital embodies process innovation results from a pattern of decreasing incremental benefits and decreasing adoption costs for later adoptions. He develops a method for comparing diffusion rates for different market structures in the capital equipment market. In particular he studies the differences in cooperative and non-cooperative behavior of market players (users of new technology). For all market structures compared, he establishes that if the incremental benefits resulting from the $k^{th}$ adoption are larger under market structure A than under market structure B, then the $k^{th}$ adoption will occur sooner under A than under B. Furthermore he demonstrates that decreasing incremental benefits and decreasing adoption costs explain diffusion in both non-cooperative and centrally planed cases. In his studies he concludes that market power makes a difference in diffusion rates, and on which side of the market that power lies makes a considerable difference. Adapting the model of Reinganum (1981a, b) he concludes that contrary to Reinganums assumption, strategic behavior is inessential to the result of a diffusion process. Solely the patterns of incremental benefits and adoption costs are the key to the difference in diffusion rates. Joint ventures result in slower adoption than do other market structures because they consider the harm each adoption does to existing investments.

The explanation for the paradox between Reinganum and Quirmbachs findings is that Reinganum analyzes increases in structural concentration in the user industry while continuing to assume non-cooperative behavior of the market players, whereas Quirmbach analyzes collusive conduct of market players. Reinganum studies the increases in market share of each firm and thus the profits to a given firm’s adoption. Quirmbach (1986) criticizes that she fails to internalize the negative externalities that new adoptions visit on existing investments.

Götz (1996) analyzes the diffusion of a new technology under monopolistic competition and concludes even in this market structure new technology is adopted sequentially rather than simultaneously. He shows that the non-

---

1 “Assymetric Nash Eqilibrium” in this case indicates that firms will never adopt the new technology simultaneously.
cooperative equilibrium can be described as a simple distribution function and derives several comparative static results. He points out the effect of the degree of competitiveness on the diffusion pattern. He shows that firms in industries with higher market power adopt the new technology later, in general, than firms operating in market with lower market power. In highly competitive industries, however, diffusion is slower, although some firms adopt earlier. (i.e. increasing the degree of competitiveness in an industry leads to an earlier start of the diffusion process, but also to an expansion of the process. Opposed to Reinganums results (1981b), Götz shows that higher competitiveness in an industry does also affect the adoption dates of late adopters.

From the above mentioned research results one can conclude that Market Power of an individual firm is an important parameter in diffusion theory. Market Power of a company depends on (a) the market structure of an industry, structural concentration, and (b) whether some form of collusive conduct exist.

**Low Market Power** usually applies to small or medium sized companies. It may also apply to large companies, which are small in relation to their business partners or competitors. As an example in the automotive industry the suppliers for the car manufacturing companies are big per definition, but small compared to their customers. Low market power also applies to any company that faces perfect competition or has low market share.

**High Market Power**, vice versa, applies to companies, that are big in relation to their business partners and/or competitors (i.e. Monopsony). A monopolist has the highest MP. Any form of collusive conduct also results in higher MP. Two or more companies could decide to boycott a new technology in order to protect their return on investment in a current technology. They might as well decide to jointly adapt a new technology. In the case of XML/EDI companies could decide to jointly invest in XML/EDI in order to influence the message standards, thus minimizing the risk of sunk costs. In both cases the natural diffusion process is altered by collusive conduct.
6.4. Putting it all together

As has been discussed in the previous sections, researchers have shown that there are different factors that influence the individual adoption decision, thus the general diffusion pattern. These factors are: relative cost advantage, triability of investment, complexity and size of investment, market concentration, collusive conduct of market players, number of previous adopters, etc.

In this section a diffusion matrix is presented, with *Market Power* of an individual firm representing the horizontal axis, and the vertical axis shows how fast a company adopts XML/EDI based on all other parameters influencing the time of adoption other than *Market Power*. Underlying assumption of this diffusion matrix is that adoption costs of XML/EDI are assumed to be falling over time. Diffusion is assumed to be sequential rather than simultaneous. There are two reasons for this assumption. Potential users of a new technology may differ in a way that the expected returns from adoption are different due to firm size, market share, R&D expenditure, etc. Differences in prior believes about the ‘true’ profitability of a new technology may result in different expected benefits from adoption and therefore in distinct adoption dates.

One way to interpret this matrix is to define four company profiles based on the combination of two possible states of the vertical axis (*early adopter, late adopter*), and two possible states on the horizontal axis (*low BP, high BP*).

![Figure 18: Company Profiles in XML/EDI Diffusion Matrix.](image-url)
**Follower:** A Follower is a Late Adopter with low Market Power. A Follower is possibly a company with low market share fearing (justified or not) that investing in an XML/EDI project could induce sunk costs, since XML/EDI is not yet sufficiently diffused in the industry and competitors as well as business partners could decide not adopt XML/EDI in the future. Furthermore a Follower tends to be technology averse or generally prefers to implement new technologies later rather than early. Other reasons for late adoption might be the size and the scope of the XML/EDI investment project in case of a small company with no/ or low risk capital.

**Risk-averse Hub:** is a Late Adopter with high Market Power. The risk averse hub implements new technology only to cut costs (not to expand market share) or out of competitive necessity (to avoid loosing market share). Factors like complexity and size of XML/EDI investment, triability of investment, compatibility, observability that influence volatility of future project cash flows out weight the fact that the company has high market power (and could be opinion leader). Thus the Risk averse hub avoids the risk of sunk costs.

**Skim the Cream:** is an Early Adopter and has high Market Power. This type of hub implements IT for strategic reasons to make use of “First mover advantage”. The Skim the Cream Hub takes advantage of its Market Power in order to influence the diffusion process and timing of XML/EDI in its industry while maximizing its own benefits. The sensibility to the risk of sunk costs is relatively low, either because the company is so big that a possible project failure would not hurt so much, or because of the fact that the company likes to take higher levels of risk.

**Niche player:** is an Early Adopter and has low Market Power. A niche player uses innovation to survive competition by differentiating itself. Market power of the niche player is low, so that it cannot influence the diffusion process and timing of XML/EDI in its market. The niche player does not fear the risk of sunk costs as much as competition. By investing in XML/EDI it hopes to be able to differentiate itself visa-a-vie its competitors. Factors like complexity and size of a project, triability of investment, compatibility, are important aspects of the investment decision. As opposed to the follower the niche player can afford (from a financial point of view) to invest into the XML/EDI project at an early diffusion stage. The borderline between niche player and follower can be very thin. If the investment project is not possible from the company’s budget point of view, then this company will be a follower because of financial restrictions.
### Mapping the Black Scholes Parameter on the Company Profiles

<table>
<thead>
<tr>
<th>Variable</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk of XML/EDI Project</td>
<td>Low Market Power</td>
<td>High Market Power</td>
<td>High Market Power</td>
<td>Low Market Power</td>
</tr>
<tr>
<td>Late Adopter</td>
<td>Late Adopter</td>
<td>Early Adopter</td>
<td>Early Adopter</td>
<td></td>
</tr>
</tbody>
</table>

**Risk of XML/EDI Project**

\[ \sigma^2 (S) = f [\sigma^2 (\text{strat}), \sigma^2 (\text{comp}), \sigma^2 (\text{PM})] \]

**Market Risk**

\[ \sigma (\text{market}) = f \{\text{BP}, \text{BP}_c, \text{MP}\} \]

- **Bargaining Power vis-a-vie Supplier BP**: Probably low
- **Bargaining Power vis-a-vie Customer BP**: Probably low
- **Market Position MP**: Probably high

**Project Risk**

\[ \sigma (\text{comp}) = f \{\text{PM}, \text{Comp.}\} \]

**Complexity of Project Comp.**

- May be lower for smaller companies if fewer transactions have to be handled and less trading partners exist (external interfaces).
- May be higher for bigger companies if more transactions have to be handled and more trading partners exist (external interfaces).
- May be lower for smaller companies if fewer transactions have to be handled and less trading partners exist (external interfaces).

**Risk of Conversion Effectiveness**

\[ \sigma (\text{comp}) = f \{\text{MC}, \text{ACT}\} \]

**Management Commitment MC**

- Low
- Low
- High
- High

**Ability to cope with transformation ACT**

- Probably low
- Probably low
- Probably high
- Probably high

**Present Value of XML/EDI Project**

\[ S = f \{U_{\text{sub}}, U_{\text{comp}}, U_{\text{low}}\} \]

**Substitute Value**

\[ U_{\text{sub}} = f \{t, s, s_b, r, \text{t}\} \]

**No. of Transactions per Period n**

Will be higher the more transactions have to be handled and the more trading partners a company has (compliance with different standards).

**Money saved per Transaction s**

Value of savings will be lower for a company, which changes from EDI to XML/EDI as opposed to a company who changes from manual to XML/EDI.

**Fix costs saved per period s_b**

Very little correlated to the company profile.

**Complementary Value = Hedonic Value**

\[ U_{\text{comp}} = f \{t, s, s_b, r, \text{t}\} \]

**Strategic Value**

\[ U_{\text{low}} = f \{t, s, s_b, r, \text{t}\} \]

- Low
- * Low vis-a-vie competitors
- * Low vis-a-vie customers (if not monopoly)
- * High vis-a-vie Business Partners
- High
- High

**Cost of XML/EDI Project**

\[ X = f \{t, FC_p, VC, n, r, \text{t}\} \]

**Initial Investment Outlay I**

- Probably lower for smaller companies, but not always proportionally. Assume A and D are smaller companies, then the FC are lower than FC of B and C.

**Periodical Fixed Costs FC_p**

Periodical Fixed costs depend on infrastructure technology fixed costs, HR FC (maintenance of system, etc.). Fixed costs probably lower for smaller companies, but not proportionally. Assume A and D are smaller companies then the FC are smaller than FC of B and C.

**Variable Costs per Transaction VC**

- Very little correlated with company profile.

**Deferral Time of Project T**

- 0-5
- 3-5
- 0-2 (3)
- 0-2 (3)

*Figure 19: Mapping the Black Scholes Parameter on the Company Profiles*
6.5. Criticism

It is beyond the scope of this dissertation to establish an in depth model for XML/EDI diffusion. Every company thinking about investing in XML/EDI should try to make a qualitative analysis about diffusion of this technology in their industry, by Delphi method, forecasting by analogy, growth curves, proportional to trend analysis, or trend analysis and extrapolation.

Nevertheless, the XML/EDI diffusion matrix introduced in this chapter shall help visualize this investment decision. Mapping the Black-Scholes parameters on the 4 company profiles makes the many parameters less theoretic, and more intuitively understandable.

Unfortunately this matrix has not been validated against empirical data. The same applies to the specifications of the Black-Scholes Formula. Data from a real case study would help validating the parameters with sensitivity analysis. Such a case study could help identify which parameters significantly influence the outcome of the formula.

7. Conclusion

The first chapter analyzed current technologies supporting electronic market transactions, and derived the need for XML as a ubiquitous data exchange format as the underlying technology for the semantic web, since the fundamental business-to-business problem in e-commerce has so far been the lack of shared semantics. Momentum is clearly with XML as a portable data mechanism. The immediate benefit of XML-based e-commerce is that it will allow new and smaller businesses to participate in e-commerce. XML promises simple implementation, wide availability, and lower costs. Even enterprises using traditional batch EDI will be able to use XML data interchange to include more partners and profit from resulting efficiencies.

Though EDI has been standardized through international EDI-standards bodies, high implementation and operational costs, per-partner customization requirements, and the need for value-added network providers have halted its widespread adoption. Extensible Markup Language is providing Internet
developers with new tools for implementing e-commerce and EDI Solutions. Electronic Data Interchange has been used for over 30 Years and represents a particular class of inter-organizational systems that automate routine tasks in the context of trading partner communication by replacing physical mailing or transportation with electronic transportation. Due to cost and complexity, small and medium sized businesses find it difficult to implement and maintain traditional e-commerce systems such as EDI. For these reasons, most small businesses today do not enjoy the operating efficiencies that an automated electronic information routing system proposes. But that is changing with the advent of XML.

The main challenge in using XML is not with document structure and syntax, but with semantics. Each application domain needs to have both specific document structure and agreed-upon semantics to make effective use of XML-defined documents. Various vertical industry segments have to agree on the meaning of their domain-specific XML/EDI tags. The key question is when will the industries agree on industry specific XML/EDI tags and how fast will diffusion of the new standard reach a critical mass. Thus, the question companies face today is not whether to invest in XML/EDI, but rather when and to what extent.

Historically, computing has been seen as essential foundation and infrastructure investment in the same way as buildings, and other office and production facilities. The purpose has been to automate the office, reduce expenses and increase efficiency and output. Today, new IT is changing the very nature of many industries. Companies are experimenting with strategic application in order to make the company more flexible and more responsive to customer needs in a rapidly changing and competitive environment. Such highly strategic investment projects are hard to assess.

Thus, potential participants in electronic markets based on XML/EDI standards face substantial uncertainties regarding the actual benefits of implementing such a system. This uncertainty can affect the strategic behavior of buyers, sellers and potential intermediaries, by inducing them to adopt a wait-and-see strategy where they delay introducing or joining a system in the hope that they will learn from the experience of other organizations.

It was concluded that traditional capital budgeting techniques for investment decisions under certainty fail to show the real value of strategic IT investments. On
the other hand, probabilistic methods and real option valuation methods that would be more appropriate for investment decisions under uncertainty are not widely adopted in the business world. Managers find them too mathematical, or find the values for the parameters too hard to obtain.

Therefore I tried to explain the idea behind the real option valuation method in a simplified way, first in general, and in the last chapter by breaking the single parameters down into multiple factors of influence of the XML/EDI investment decision.

I supposed that every company today facing the XML/EDI investment decision is the holder of an American deferral option on a dividend-paying asset. The asset underlying this option is the potential stream of revenues from an investment opportunity that will materialize only when a company decides to invest in an XML/EDI project at any given future date, where the dividends are the revenues lost during deferral time.

Without a real life case study, this concept still seemed too theoretic. Therefore I tried to map this specific option valuation framework on different company profiles. These company profiles were derived from a diffusion matrix, which is based on the theory of technology diffusion and the case of XML/EDI. The final table shows the company profiles in the columns and the single parameters of the option-pricing framework in the rows. The result shows whether a parameter depends on a specific company profile, and if yes in which range the value of each parameter can lie.
Appendix 1: XML Tutorial

This is a short tutorial into the syntax of XML. The online version can be found under: http://indi.wu-wien.ac.at/~student/xml/index.htm

This chapter intends to provide an easy to understand technical introduction to the Extensible Markup Language (XML). It does not give complete information about the XML Syntax and rules, nor does it discuss Extensible Style sheet Language (XSL) or the Extensible Linking Language (XLL). The complete specifications of XML v 1.0 can be found in the W3C (2000) XML Specifications.

The first section defines the important terms. In the following sections the XML Syntax will be analyzed in detail, to give an insight into well-formed and valid XML documents, how to write Document Type Definitions, and other important rules of the XML syntax. In order to make the syntax easier to understand, a sample XML-file will be introduced and be discussed in detail throughout the whole introduction in the relevant chapters.

Markup

Markup gives meaning to a document, something we do every day like highlighting text, check marks on a bank statement, etc. If we want others to understand what markup means, we need a set of rules to declare what constitutes markup and declaring exactly what it means.

A markup language is just a set of rules. In SGML for example anything in angle brackets is considered markup. <...>

The makers of HTML used SGML to make a set of rules declaring what markup in HTML means. This set of rules is contained in a separate document called the HTML DTD (Document Type Definition). HTML DTD for example says that when you come across a <P> in a document start a new paragraph. In HTML there are three different types of markup:

- **Structural Markup**
  
  Tells how the document should be structured; i.e. which position within a document an element takes. E.g.: <P>, <DIV>, <H*>, ...
Stylistic Markup
Tells how the document is to be styled. These are procedural tags, which give formatting instructions. E.g. `<I>`, `<U>`, `<B>`, ...

Semantic Markup
Tells us something about the content of the document. `<TITLE>`, `<CODE>`...

HTML mainly focuses on structural markup and stylistic markup while the markup of content has not been a focus. In XML you can define semantic markup tags.

Sentence in HTML

```html
<P>Requiem is a song composed by Mozart</P>
```

Semantics of the sentence expressed in XML

```xml
<song>
<title>Requiem</title>
<composer>Mozart</composer>
</song>
```

Tags

XML is very similar to HTML. It has tags, which identify elements. These tags also contain attributes about these elements. In XML a tag is what is written between angled brackets, i.e. XML tags open with the `<` symbol and end with the `>` symbol. They always come in and matched pairism, with the defined element between the open and close tag. `<composer>` is an example for an opening tag. In XML all opening tags must have closing tags, in this case the closing tag would look like this: `</composer>`.

Element

```xml
<composer>Mozart</composer>
```

Start Tag

The beginning of every non-empty XML element is marked by a start-tag. An example of a start-tag: `<composer>`

End Tag

The end of every non-empty XML element is marked by a end-tag. An example of an end-tag: `</composer>`

Element Content

The text between the start-tag and end-tag is called the element's content. The element content in this case would be: Mozart
Empty-element tags may be used for any element that has no content, whether or not it is declared using the keyword **EMPTY** (see chapter on Element Type Declaration). For interoperability, the empty-element tag must be used, and can only be used, for elements that are declared **EMPTY**.

**Empty Element Tag**

If an element is empty, it must be represented either by a start-tag immediately followed by an end-tag or by an empty-element tag. An empty-element tag takes a special form:

```
<IMG align="center" src="logo.gif"/>
<br/>
```

As opposed to HTML where a line break is declared with:

```
<br/>
```

The name in the start-tags and end-tags gives the element's type. In our example it would be: composer.

**Element Type**

The Name-AttValue pairs are referred to as the attribute specifications of the element, with the Name in each pair referred to as the attribute name and the content of the AttValue as the attribute value. In the following example we have an empty element 'IMG' and two attributes 'align' and 'src'.

```
<IMG align="center" src="logo.gif"/>
```

XML and HTML markup may be used in the same document. By convention HTML tags are put in upper case and XML tags in lower case. Furthermore, XML is case sensitive. Therefore `<COMPOSER>`, `<Composer>` and `<composer>` are different kinds of tags in XML.

Tags should begin either with a letter, an underscore (_), or a colon (:) followed by some combination of letters, numbers, periods (.), colons, underscores, or hyphens (-) but no white space, with the exception that no tags should begin with any form of "xml". It is also a good idea to not use colons as the first character in a tag name even if it is legal. Using a colon first could be confusing. The XML 1.0 standard does not limit the length of tag names, but XML processors may limit the length of markup names. To visualize XML syntax, a sample XML file will be
introduced in the next section, which will be analyzed in detail throughout the next sections.

### Character Data

Text consists of *character data* and *markup*. XML defines the text between the start and end tags to be "character data" and the text within the tags to be "markup". Data is typically "character data" (letters, numbers, punctuation... anything within the boundaries of valid Unicode) but it can also be binary data. Markup includes tags, comments, processing instructions, DTDs, references, etc.

*Markup* takes the form of (1) Start tags, (2) End tags, (3) Empty element tags, (4) Entity references, (5) Character references, (6) Comments, and (7) CDATA section delimiters. *Character Data* represents all text of a document that is not markup.

### XML Documents

XML documents are similar to HTML documents. They contain information and markup tags that define the information and are saved as ASCII text. The name of the XML document has an XML extension *xyz.xml*. A data object is an XML document if it is well formed. A well-formed XML document may in addition be valid if it meets certain further constraints.

- **Well formed** XML documents contain text and XML tags which conform with the XML syntax.
- **Valid** XML documents must be well formed and are additionally error checked against a Document Type Definition (DTD). A DTD is a set of rules outlining which tags are allowed, what values those tags may contain and how the tags relate to each other. Typically a valid document is used when documents require error checking, and enforced structure, or are working within a company- or industry wide environment in which many documents need to follow the same guidelines.

The following XML document, film.xml, will be the guiding example throughout this technical introduction. Those familiar with SGML and HTML, will find it quite easy to understand the following example. At the end of this technical introduction, understanding this source code should not be problem any more.
<?xml version="1.0"?>
<!DOCTYPE videocollection [ 
<!ENTITY R "Romance"> 
<!ENTITY WAR "War"> 
<!ENTITY COM "Comedy"> 
<!ENTITY SF "Science Fiction"> 
<!ENTITY ACT "Action"> 
<!ELEMENT videocollection (title,genre,year,language*,crew?)> 
<!ELEMENT title (#PCDATA)> 
<!ELEMENT genre (#PCDATA)> 
<!ELEMENT year (#PCDATA)> 
<!ELEMENT language (#PCDATA)> 
<!ELEMENT crew (cast*, director*, screenwriter*, cinematog*, editor*)> 
<!ELEMENT cast (#PCDATA)> 
<!ELEMENT director (#PCDATA)> 
<!ELEMENT screenwriter (#PCDATA)> 
<!ELEMENT cinematog (#PCDATA)> 
<!ELEMENT editor (#PCDATA)> 
]> 
<!ELEMENT title (#PCDATA)> 
<!ATTLIST title xml:lang NMTOKEN "EN" id ID #IMPLIED> 
<!ELEMENT genre (#PCDATA)> 
<!ELEMENT year (#PCDATA)> 
<!ELEMENT language (#PCDATA)> 
<!ELEMENT crew (cast*, director*, screenwriter*, cinematog*, editor*)> 
<!ELEMENT cast (#PCDATA)> 
<!ELEMENT director (#PCDATA)> 
<!ELEMENT screenwriter (#PCDATA)> 
<!ELEMENT cinematog (#PCDATA)> 
<!ELEMENT editor (#PCDATA)> 
</videocollection> 
<title>Tootsie</title> 
<genre>&COM;</genre> 
<year>1982</year> 
<language>English</language> 
<crew> 
<cast gender="unknown" role="helen">Dustin Hoffman</cast> 
<cast>Jessica Lang</cast> 
<cast>Teri Gar</cast> 
<cast>Sydney Pollak</cast> 
<director>Sydney Pollak</director> 
</crew> 
</videocollection>
1. Well Formed XML Documents

Well-formed XML documents simply markup pages with descriptive tags, without the need of describing or explaining what these tags mean. In other words a well-formed XML document does not need a DTD, but is must conform to the XML syntax rules. If all tags in a document are correctly formed and follow XML guidelines, then a document is considered as well-formed. Syntax is the Grammar of a language. For a document in XML to be well formed, it must obey the following most important rules:

XML documents must contain at least one element. In this example "Tootsie" is not well formed, because it is not marked up as an element within angle brackets.

<table>
<thead>
<tr>
<th>Well Formed</th>
<th>Not Well Formed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;title&gt;Tootsie&lt;/title&gt;</code></td>
<td>&quot;Tootsie&quot;</td>
</tr>
</tbody>
</table>

XML documents must contain a unique opening and closing tag that contains the whole document, forming what is called a root element. In this example, the second column is not well formed because it lacks a root element as in the first column:

<table>
<thead>
<tr>
<th>Well Formed</th>
<th>Not Well Formed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;videocollection&gt;</code></td>
<td><code>&lt;title&gt;Tootsie&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;title&gt;Tootsie&lt;/title&gt;</code></td>
<td><code>&lt;title&gt;Jurassic Park&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;title&gt;Jurassic Park&lt;/title&gt;</code></td>
<td><code>&lt;title&gt;Mission Impossible&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/videocollection&gt;</code></td>
<td><code>&lt;/title&gt;...</code></td>
</tr>
</tbody>
</table>

All other tags must be nested properly, i.e. there must be an opening and a closing tag and the tags cannot overlap. The tags that in HTML would normally stand alone, such as `<img>` or `<br>` Tag are called "empty Tags" when used in an XML document In XML empty Tags look like this: e.g.: `<BR/>`.

`</title>...` has no closing angle bracket, therefore the tag is not complete!

`</title>)...` has a wrong closing bracket, therefore the tag is not complete!
The example in the right column shows the tags are not properly nested.

<table>
<thead>
<tr>
<th>Well Formed</th>
<th>Not Well Formed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;videocollection&gt;</code></td>
<td><code>&lt;videocollection&gt;</code></td>
</tr>
<tr>
<td><code>&lt;title&gt;Tootsie&lt;/title&gt;</code></td>
<td><code>&lt;title&gt;Tootsie&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;title&gt;Jurassic Park&lt;/title&gt;</code></td>
<td><code>&lt;title&gt;Jurassic Park&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;title&gt;Mission Impossible&lt;/title&gt;</code></td>
<td><code>&lt;title&gt;Mission Impossible&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/videocollection&gt;</code></td>
<td><code>&lt;/videocollection&gt;</code></td>
</tr>
</tbody>
</table>

Tags in XML are case sensitive, that means that `<CREW>`, `<Crew>` and `<crew>` are not the same. The XML processing instruction must all be lowercase. But keywords in DTDs must be all UPPERCASE, such as `ELEMENT`, `ATTLIST`, `#REQUIRED`, `#IMPLIED`, `NMTOKEN`, `ID`, etc. However, your own elements and attributes may be any case you choose, as long as you are consistent.

<table>
<thead>
<tr>
<th>Well Formed</th>
<th>Not Well Formed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;crew&gt;Sydney Pollak&lt;/crew&gt;</code></td>
<td><code>&lt;CREW&gt;Sydney Pollak&lt;/crew&gt;</code></td>
</tr>
<tr>
<td><code>&lt;crew&gt;Sydney Pollak&lt;/Crew&gt;</code></td>
<td><code>&lt;crew&gt;Sydney Pollak&lt;/Crew&gt;</code></td>
</tr>
</tbody>
</table>

Attribute values must always be quoted (as opposed to HTML).

<table>
<thead>
<tr>
<th>Well Formed</th>
<th>Not Well Formed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;title id=&quot;1&quot;&gt;Tootsie&lt;/title&gt;</code></td>
<td><code>&lt;title id=&quot;1&quot;&gt;Tootsie&lt;/title&gt;</code></td>
</tr>
<tr>
<td><code>&lt;title id=1&gt;Tootsie&lt;/title&gt;</code></td>
<td><code>&lt;title id=1&gt;Tootsie&lt;/title&gt;</code></td>
</tr>
</tbody>
</table>

These are just some examples for the above-mentioned well formdness constraints, which themselves are only the most important, but by far not complete.

In our example the following section of XML data is Well Formed. It does not contain a DTD, but it meets all XML syntax rules.
<?xml version="1.0"?>
<videocollection>
<title id="1">Tootsie</title>
<genre>comedy</genre>
<year>1982</year>
<language>English</language>
<cast>Dustin Hoffman</cast>
<cast>Jessica Lang</cast>
<cast>Teri Gar</cast>
<cast>Sydney Pollak</cast>
<crew>
<director>Sydney Pollak</director>
</crew>
<title id="2">Jurassic Park</title>
<genre>science fiction</genre>
<year>1993</year>
<language>English</language>
<cast>Sam Neil</cast>
<cast>Laura Dern</cast>
<cast>Jeff Goldblum</cast>
<crew>
<director>Steven Spielberg</director>
</crew>
<title id="3">Mission Impossible</title>
<genre>action</genre>
<year>1996</year>
<language>English</language>
<cast>Tom Cruise</cast>
<cast>Jon Voight</cast>
<cast>Emmanuelle Beart</cast>
<cast>Jean Reno</cast>
<crew>
<director>Brian de Palma</director>
</crew>
</videocollection>

A Parser is used to check whether a file is well formed or not.

2. Valid XML Documents

Valid XML is a more rigid, or formal, form of XML. All XML documents are well-formed documents (otherwise they would not be XML documents). Some XML documents are additionally valid. Valid documents must conform not only to the syntax, but also to the DTD (Document Type Definition). DTD is a set of rules that defines what tags appear in a XML document, so that viewers of an XML document know what all the tags mean. DTDs also describe the structure of a
document. To help you get a feel for the difference between well-formed XML and valid XML, consider the following well-formed English:

```
went we. never school to
```

As you can see, all the words and punctuation represent well-formed elements of English. However, unless you are into absurdist poetry, the words and punctuation are virtually meaningless, and difficult to interpret. To be valid English, the words must conform to a standard grammatical structure. For example:

```
We never went to school.
```

In the case of the markup languages defined by XML, the DTD provides the grammatical structure to bring order to the elements of the language. The main difference between valid and well formed is that Valid XML requires a DTD and whereas well formed XML does not. Still it is advisable for an XML document to have a DTD, because if several people are authoring the document, the DTD will set out the ground rules that they can all work by, and more importantly they can use a parser (the validity checker) to make sure that they are not violating the rules. The XML DTD can either be in the prolog of the document, or it can be in a separate file that is referred to in the prolog.

---

**Prolog**

Documents are made up of a prolog and a body. The document prolog contains the XML Declaration and the document body contains the actual marked up document. According to the official XML specifications, every well-formed XML document must contain a prolog. A prolog can consist of nothing else that white space. The prolog must follow two simple rules:

- It must come before the opening root tag (first element in the document).
- It can contain a *version declaration*, a *document type declaration*, *comments*, and *processing instructions*, but it need not contain any of them at all.
1. Comments

Comments are information for the user and author. A well-commented DTD is particularly important in order to avoid confusion. Comments may appear anywhere in a document outside other markup. In addition, they may appear within the document type declaration at places allowed by the grammar. They are not part of the document's character data. An XML Processor is not required to pass this information to the user agent. The comments we used in our example in the beginning where following:

```xml
<!-- ///////////// PROLOG \\\\\\ -->
<!-- ///////////// PROLOG \\\\\\ -->
<!--This DTD is a modified version of FlixML DTD -->
<!--This starts the DTD, and addresses to the document structure-->

<!-- ///////////// ENTITIES \\\\\\ -->
<!-- ///////////// ENTITIES \\\\\\ -->
<!-- ///////////// END OF ENTITIES \\\\\\ -->
<!-- ///////////// END OF ENTITIES \\\\\\ -->

<!-- ///////////// DOCUMENT CONTENT MODEL \\\\\\ -->
<!-- ///////////// DOCUMENT CONTENT MODEL \\\\\\ -->

<!-- ///////////// END OF DOCUMENT CONTENT MODEL \\\\\\ -->
<!-- ///////////// END OF DOCUMENT CONTENT MODEL \\\\\\ -->

<!-- ///////////// END OF PROLOG \\\\\\ -->
<!-- ///////////// END OF PROLOG \\\\\\ -->

<!-- ///////////// BODY \\\\\\ -->
<!-- ///////////// BODY \\\\\\ -->

<!-- ///////////// END OF BODY \\\\\\ -->
<!-- ///////////// END OF BODY \\\\\\ -->
<!-- ///////////// END OF BODY \\\\\\ -->
<!-- ///////////// END OF BODY \\\\\\ -->
```
For compatibility, the double hyphen `--` must not occur within comments as it might be confusing to the XML processor.

A comment must not be placed: (1) within a tag, the code would otherwise be poorly-formed XML. (2) inside of an entity declaration. (3) before the XML declaration.

2. Processing Instructions

Processing Instructions (PI's) are information for the application. They are not really of interest to the XML parser. Instead, the instructions are passed to the application using the parser, because the purpose of processing instructions is to represent special instructions for the application. Like comments, they are not textually part of the XML document.

All processing instructions, including the XML declaration, begin with `<?` and end with `?>`. The name of the processing instruction follows the initial. The PI begins with the PITarget used to identify the application to which the instruction is directed.

```xml
<?name pidata?>
<?xml version="1.0"? encoding="UTF-8" standalone="yes"?>
```

A particular document may have processing instructions for several different applications, so the processing instruction target (PITarget), right after the opening `<?`, identifies the target application for this processing instruction.

3. Version declaration

Version declaration, as a type of Processing Instruction, it is information for the application. XML documents start with an XML version declaration (XML declaration), which specifies the version of XML being used. For the time being there exists only version 1.0 of XML. Although the XML version declaration is optional, it is suggested by the W3C specification. The XML declaration is a processing instruction that notifies the processing agent that the following document has been marked up as an XML document. It will look something like the following:
The version declaration can also contain other information such as an *encoding declaration* or *standalone declarations*.

*Encoding declarations* inform the processor what kind of code the document uses (e.g. UTF8, which is the same character Set as ASCII). All XML parsers must support 8-bit and 16-bit Unicode encoding corresponding to ASCII. However, XML parsers may support a larger set. For a list of encoding types go to the XML Specification.

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

*Standalone declarations* tell the processor whether the document can be read as standalone document, or whether it needs to look outside the document for the rules. Thus, if standalone is set to "yes", there will be no markup declarations in external DTDs. Setting it to "no" leaves the issue open. The document may or may not access external DTDs.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

4. **Document Type Declaration**

Document Type Declarations are information for the parser, upon which the validity of XML documents is checked. Document Type Declaration is a XML mechanism that defines the constraints of the logical structure and supports the use of predefined storage units. A large percentage of XML specifications deal with various sorts of declarations that are allowed in XML. One of the greatest strengths of XML is that it allows the user to create own tag names. But for any given application, it is probably not meaningful for tags to occur in a completely arbitrary order. If the document is to have meaning, there must be some constraint on the sequence and nesting of tags. Declarations are where these constraints can be expressed.
More generally, declarations allow a document to communicate meta-information to the parser about its content. Meta-information includes the allowed sequence and nesting of tags, attribute values and their types and defaults, the names of external files that may be referenced and whether or not they contain XML, the formats of some external (non-XML) data that may be included, and entities that may be encountered. The document Type declaration can contain the following:

- Document name.
- Reference to an external DTD (Document Type Definition).
- Markup declaration (internal DTD).
- Parameter entity references.

Document Type Definition (DTD) and Document Type Declarations are often mixed up, so remember that DTD is part of the Document Type Declaration.

**Document Type Definition (DTD)**

A DTD is a set of rules that defines what tags appear in a XML document, what attributes the tags may have and what a relationship the tags have with each other. When an XML document is processed, it is compared within the DTD to be sure it is structured correctly and all tags are used in the proper manner. This comparison process is called validation and is performed by a tool called the parser. A DTD is only needed for a valid XML document. A DTD is also useful for checking a XML document for structural errors.

```
film.dtd
<!DOCTYPE film [ 
<!ENTITY COM "Comedy"> 
<!ENTITY SF "Science Fiction"> 
<!ELEMENT film (title+,genre,year)> 
<!ELEMENT title (#PCDATA)> 
<!ATTLIST title xml:lang NMTOKEN "EN" id ID IMPLIED> 
<!ELEMENT genre (#PCDATA)> 
<!ELEMENT year (#PCDATA)> ]>
```
1. Internal DTDs

Internal DTD (markup declaration) are inserted within the document type declaration.

```xml
<?xml version="1.0"?>
<!DOCTYPE film [ 
  <!ENTITY COM "Comedy"> 
  <!ENTITY SF "Science Fiction"> 
  <!ELEMENT film (title+,genre,year)> 
  <!ELEMENT title (#PCDATA)> 
  <!ATTLIST title xml:lang NMTOKEN "EN" id ID #IMPLIED> 
  <!ELEMENT genre (#PCDATA)> 
  <!ELEMENT year (#PCDATA)> ]>

<film> 
  <title id="1">Tootsie</title> 
  <genre>&COM;</genre> 
  <year>1982</year> 
  <title id="2">Jurassic Park</title> 
  <genre>&SF;</genre> 
  <year>1993</year> 
</film>
```

DTDs inserted this way are only used in that specific document. This might be the approach to take for the use of a small number of tags in a single document, as the above example has shown.

2. External DTD

DTDs can be very complex and creating a DTD requires a certain amount of work. DTDs are stored as ASCII text files with the extension .dtd. In the following example it is assumed, that the previously internal DTD was saved as a separate file (under the name film.dtd), and is therefore now referred to as external definition (external DTD):
<?xml version="1.0"?>
<!DOCTYPE film SYSTEM "film.dtd">

<film>
  <title id="1">Tootsie</title>
  <genre>&COM;</genre>
  <year>1982</year>

  <title id="2">Jurassic Park</title>
  <genre>&SF;</genre>
  <year>1993</year>
</film>

For making the creation of XML documents easier, there are published DTD that define tags for commonly used elements. This avoids the recreation of existing DTD by merely pointing to them in the doctype tag of the XML file. Furthermore these XML rules pointed to must be followed when creating the XML document.

3. Shared DTDs

The document type declaration can point to an external subset (a special kind of external entity) containing markup declarations, or can contain the markup declarations directly in an internal subset, or can do both. The DTD for a document consists of both subsets taken together.

When both the external and internal subsets are used, the internal subset is considered to occur before the external subset. This has the effect that entity and attribute-list declarations in the internal subset take precedence over those in the external subset.

There are four kinds of markup declarations in XML within the DTD:

- Element declarations.
- Attribute list declarations.
- Entity declarations.
- Notation declarations.
It is recommended to be as conservative as the DTD allows being, when declaring elements, attributes, entities, and notations. The rest of this section will analyze these types of markup declaration in detail.

### Element Type Declarations

Element declarations identify the names of elements and the nature of their content. As in HTML, elements are the basic building blocks of XML. Element type declarations constrain which element types can appear as children of the element.

1. **Element Content**

   An element type has element content when elements of that type must contain only child elements (no character data), optionally separated by white space. The "content model" follows the element name. The content model defines what an element may contain. An element type declaration takes the following form:

   ```xml
   <!ELEMENT [element_name] (names of allowed elements)>  
   ```

   Another example for element declaration:

   ```xml
   <!ELEMENT customer (name,company?,(fax|email|phone)*, address+)>  
   <!ELEMENT name (#PCDATA)>  
   <!ELEMENT ...>  
   ```

   **PCDATA** is short for parsed character data and means any text or other characters that is not markup or ",", "&", or "[]". Another example for element declaration:

   ```xml
   <!ELEMENT customer (name,company?,(fax|email|phone)*, address+)>  
   <!ELEMENT name (#PCDATA)>  
   <!ELEMENT ...>  
   ```

   In this case, a customer must contain name, address and optionally fax, email or phone. The commas between element names indicate that they must occur in succession. The pipe character | is used to specify an "OR" operation. It indicates an option (either 'fax', 'email' or 'phone'). The plus after 'address' indicates that it
may be repeated more than once but must occur at least once. The * symbol indicates zero or more possibilities for either 'fax', 'email' or 'phone'. The question mark after 'company' indicates that it is optional (zero or one). A name with no punctuation, such as 'name', must occur exactly once.

```
cast*, crew*
Asterix indicates zero or more, we might have between zero and more cast and
crew members.

language?
Zero or one time, the question mark after the element name indicates that we may
include one language of the original version or none, but not several.

gender+
One or more, the plus indicates that we have to include at least one genre, since a
movie can also be classified into two genres like romance and action.

( )
Parentheses indicates a group of expressions to be matched together.

| The pipestern is used to indicate an option. In this case there is either fax, email or
phone.
<!ELEMENT customer (name,company?, (fax|email|phone)*, address+)>

and it is also compulsory for mixed content
<!ELEMENT contact (#PCDATA| date)*> ,

Each of these elements, separated by commas, must occur once, and in the exact
order they are declared.

<!ELEMENT film (title,genre+,year)>
```

Two other content models are possible: EMPTY indicates that the element has no
content (and consequently no end-tag), and ANY indicates that any content is
allowed. The ANY content model is sometimes useful during document conversion,
but should be avoided at almost any cost in a production environment because it
disables all content checking in that element. It is bad form to use the ANY
keyword for any element other than the root element.
Children can be grouped using parentheses. Within a grouping, you may use only one connector (such as the comma or the pipestern). Thus, it is invalid to use

```xml
<!ELEMENT customer (name,fax|email)>
```

Instead, a subgroup like the following must be created:

```xml
<!ELEMENT customer (name,(fax|email))>
```

### 2. Mixed Content

An element type has mixed content when elements of that type may contain character data, optionally interspersed with child elements. In some circumstances, you will wish to include such parsed character data as a valid element. In this case, the types of the child elements may be constrained, but not their order or their number of occurrences. Examples of mixed content declarations:

```xml
<?xml version = "1.0" standalone = "yes"?>
<!DOCTYPE FILM [ 
  <!ELEMENT FILM (#PCDATA|TITLE|GENRE|YEAR)*>  
  <!ELEMENT TITLE (#PCDATA)> 
  <!ELEMENT GENRE (#PCDATA)> 
  <!ELEMENT YEAR (#PCDATA)> ]>

<FILM>
<TITLE>Tootsie</TITLE>
<GENRE>comedy</GENRE>
<YEAR>1982</YEAR>
Critic: very funny movie!
</FILM>
```
PCDATA is short for parsed character data and means any text or other characters that is not markup or ",& or []. The asterix * as well as the pipestern | are compulsory.

### Attribute List Declarations

As discussed in the last section, the element type declaration constrains the element's content. This section will explain how attribute list declaration constrains the attribute's content.

Attributes provide additional information about elements. Elements as well as attributes also exist in HTML. `<H1 align="center">` includes an element H1, and an attribute value center. In XML, attributes specify additional data about an element, but it is never formatting related as in HTML. It is, instead, additional data about that particular element.

Attributes are created in the DTD, when the elements are specified. They are specified through an attribute list as stated in the example below.

```
<!--This defines a listing of books-->
<!DOCTYPE books [  
<!ELEMENT booklist (title, author)>  
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title  
edition (CDATA) #REQUIRED  
type (paper|cloth|hard)"paper">  
<!ELEMENT author (#PCDATA)> ]>
```

Attribute declarations identify which elements may have attributes, what types of attributes they may have, what values the attributes may hold, and what default value each attribute has. Attribute specifications may appear only within start-tags and empty-element tags.

Attribute-list declarations specify the name, data type, and default value (if any) of each attribute associated with a given element type. **Attribute Default value** provides default values for attributes. **Attribute Type** establishes type constraints
for these attributes. *Attribute Name* defines the set of attributes pertaining to a given element type.

1. **Attribute Default Values**

   - **#REQUIRED**
     
     means that the attribute must have a value every time this element is listed. Although there is no default value provided by the DTD, the attribute when actually implemented in an XML document must define a value. In our example the default value tells the validator that the ID must be present and a value must be specified.

     ```xml
     <!ATTLIST title id ID #REQUIRED>
     ```

   - **#FIXED**
     
     Sometimes one will want to provide a default value that the document author may not modify. In that case, **#FIXED** will be used. Attribute where only one value is allowed, that must be included. An attribute declaration may specify that an attribute has a fixed value. In this case, the attribute is not required, but if it occurs, it must have the specified value. One use for fixed attributes is to associate semantics with an element.

   - **#IMPLIED**
     
     The attribute value is not required, and no default value is provided. If a value is not specified, the XML processor must proceed without one. The processor ignores this attribute unless it is used as part of this element. It does not assume any default value. When you use the IMPLIED default, you will provide a default value for the document author. If the document author does not override your default, your default will be used. Unless otherwise specified, attributes are implied.

   - **"VALUE"**
     
     An attribute can be given any legal value as a default. The attribute value is not required on each element in the document, but if it is not present, it will appear to be the specified default. Value (default values) provides a default
value for that attribute. If the attribute is not included in the element, the processing program assumes that this is the attributes value.

```
<attitude interest="unknown">
  e.g. <attitude interest="warm">

If the document just contained
<interest/>

Then the attribute "unknown" would be presumed by the validating software.
```

Attribute values in XML must always be contained in quotes.

2. Attribute Types

There are three kinds of XML attribute types: a string type, a set of tokenized types, and enumerated types.

(a) String Type

String Type may take any literal string as a value. CDATA attributes are strings where any text is allowed. Don't confuse CDATA attributes with CDATA sections. In CDATA attributes, markup is recognized.

```
<attlist actor role CDATA #IMPLIED>
  e.g.: <actor role="hamlet">
```

(b) Set of Tokenized Types

A name must not appear more than once in an XML document as a value of this type. The tokenized types have varying lexical and semantic constraints. In other words, a tokenized attribute type represents a fixed set of keyword types with special meanings.

ID Attribute Default: ID represents a unique ID name for the attribute that identifies the element within the context of the document. IDs are much like internal links in plain HTML. For the most part, ID is used primarily by programs or scripting languages that process the document. The value for ID
must be a valid XML name beginning with a letter and containing alphanumeric characters or the underscore character without any whitespace. ID is incompatible with the #FIXED default but it must have a declared default of #IMPLIED or #REQUIRED. Also, take care that your ID values are unique within a document! All ID values used in a document must be different. IDs uniquely identify individual elements in a document. Elements can have only one single ID attribute

\[
\text{id ID #REQUIRED}
\]

\[
<!ATTLIST title id ID #REQUIRED>
\]

e.g. <title id="3">

- **IDREF**: Values of type IDREF must match the Name production, and values of type IDREFS must match Names; each Name must match the value of an ID attribute on some element in the XML document; i.e. The IDREF type allows the value of one attribute to be an element elsewhere in the document provided that the value of the IDREF is the ID value of the referenced element.

- **Entity Name**: Values of type ENTITY must match the Name production, values of type ENTRIES must match Names; each Name must match the name of an unparsed entity declared in the DTD.

- **Name Token**: The NMTOKEN and NMTOKENS types are another example of those types that are useful primarily to processing applications. The types are used to specify valid names. You might use them when you are associating some other component with the element. Values of type NMTOKEN must match the NMTOKEN production; values of type NMTOKENS must match NMTOKENS.

(c) Enumerated Types

You can specify the value of an attribute that must be taken from a specific list of names. There are two kinds of enumerated types
**Enumeration**: Attributes can also be defined by a list of acceptable pipe-delimited values from which the document author must choose. In this case, each of the values is explicitly enumerated in the declaration.

```xml
<!ATTLIST cast
gender (male|female|unknown)"unknown">#
```

**Notation Type**: specifies that the names must match a particular notation name. Values of this type must match one of the `NMTOKEN` tokens in the declaration. For interoperability, the same `NMTOKEN` should not occur more than once in the enumerated attribute types of a single element type.

### Entities

XML documents can be made of information drawn from different files. These pieces of information are called entities. It might be easier to think of entities as a macro for programmers, or as aliases for more complex functions. A single entity name can take the place of a whole lot of text. Entity references cut down on the amount of typing one has to do because anytime it is necessary to reference that bunch of text, the alias name is simply used and the processor will expand the contents of the alias.

Entities allow referring to other data and pages as shortcuts, so that declaring the same information in a document or DTD is not necessary. Entity declarations allow you to associate a name with some other fragments of the document. That construct can be a chunk of regular text, a chunk of the document type declaration, or a reference to an external file containing either text or binary data. Entities are declared in the DTD, similar to elements and attributes.

### 1. Parsed vs. Unparsed Entities

Entities may be either parsed or unparsed. A parsed entity's content is referred to as its replacement text; this text is considered an integral part of the document. An unparsed entity is a resource whose contents may or may not be text, and if text, may not be XML. Each unparsed entity has an associated notation, identified by
name. Beyond a requirement that an XML processor makes the identifiers for the entity and notation available to the application, XML places no constraints on the contents of unparsed entities. Parsed entities are invoked by name using entity references; unparsed entities by name, given in the value of ENTITY or ENTITIES attributes.

2. General Entities vs. Parameter Entities

General entities (or simply entities) are entities for use within the document content. Parameter entities are parsed entities for use within the DTD. These two types of entities use different forms of reference and are recognized in different contexts. Furthermore, they occupy different namespaces; a parameter entity and a general entity with the same name are two distinct entities.

The Name identifies the entity in an entity reference or, in the case of an unparsed entity, in the value of an ENTITY or ENTITIES attribute. If the same entity is declared more than once, the first declaration encountered is binding; at user option, an XML processor may issue a warning if entities are declared multiple times. A general entity declaration could look like this:

```
<!DOCTYPE videocollection [ 
<!ENTITY R "Romance">  
<!ENTITY WAR "War">  
<!ENTITY COM "Comedy">  
<!ENTITY SF "Science Fiction">  
<!ENTITY ACT "Action"> ]>
```

These entities are then used (referred to) in a XML document like this: An (general) entity reference refers to the content of a named entity. References to parsed general entities use ampersand (&) and semicolon (;) as delimiters.

```
<videocollection>  
<title id="1">Tootsie</title>  
<genre>&COM;</genre>  
<year>1982</year>  
<title id="2">Jurassic Park</title>  
<genre>&SF;</genre>  
<year>1993</year>  
<title id="3">Mission Impossible</title>  
<genre>&ACT;</genre>  
<year>1996</year>  
</videocollection>
```
As in HTML, the name of the entity is preceded with an ampersand (&) and followed by a semicolon (;).

Parameter entity declaration is used for shortcuts within the DTD. An example for a parameter entity declaration:

```
<!ENTITY % NAME "text that you want to be represented by the entity">
<!ENTITY % pub "\xc9;ditions Gallimard" >
<!ENTITY rights "All rights reserved" >
<!ENTITY book "La Peste: Albert Camus, \xA9; 1947 %pub;.&rights;">
```

Parameter-entity references use percent-sign (%) and semicolon (;) as delimiters. The Parameter entity reference then is:

```
<!ENTITY book "La Peste: Albert Camus, \xA9; 1947 %pub;.&rights;">
```

The replacement text for the entity `book` is:

```
La Peste: Albert Camus, © 1947 Éditions Gallimard.
```

XML expands the power of entities in a big way. There are three kinds of entities.

(a) Internal Entities

If the entity definition is an entity value, the defined entity is called an internal entity. There is no separate physical storage object, and the content of the entity is given in the declaration.

Internal Entities allow for entities to be defined in DTDs so they can be used throughout the rest of the document. If, for instance, a phrase such as Science Fiction occurs frequently in a document, following could be put in the DTD to avoid typing the whole phrase each time. Internal entities allow you to define shortcuts for frequently typed text or text that is expected to change, such as the revision status of a document. Internal entities help avoiding misspellings and retyping of the same information.
An internal entity is a parsed entity. Example of an internal entity declaration:

```xml
<!ENTITY SF "Science Fiction">
Whenever the full term needs to be used in the document, it sufficient to type &SF;
```

Internal entities can include references to other internal entities, but it is an error for them to be recursive.

**(b) External Entities**

If the entity is not internal, it is an external entity. External entity references are used for replacement text that is really long. The information is then kept in another file.

External entities allow an XML document to refer to an external file. External entities contain either text or binary data. If they contain text, the content of the external file is inserted at the point of reference and parsed as part of the referring document. Binary data is not parsed and may only be referenced in an attribute. Binary data is used to reference figures and other non-XML content in the document.

The entity declaration in this example refers to documents that are located in different sections. They are placed into the XML file by using the entities, rather than cutting and pasting the contents of separate files together. An entity that has text defined external to the document can be used by using the `SYSTEM` keyword.

```xml
<!ENTITY LIagreement SYSTEM "http://www.mydomain.com/license.xml">
<!ENTITY LOGO SYSTEM "http://www.mydomain.com/logo.gif" NDATA GIF87A>
```

In this case, the XML processor will parse the content of that file as if its content had been typed at the location of the entity reference.

The entity is also an external entity, but its content is binary. The `LOGO` entity can only be used as the value of an `ENTITY` (or `ENTITIES`) attribute (on a graphic element, perhaps). The XML processor will pass this information along to an
application, but it does not attempt to process the content of
/standard/logo.gif.

(c) Predefined Entities

There are five predefined XML entities, most of which should be well known to
HTML coders:

```
&lt; produces the left angle bracket <
&gt; produces the right angle bracket >
&amp; produces the ampersand &
&amp;apos; produces a single quote character '
&amp;quot; produces a double quote character "
```

Entity references can also be used within tag attributes.

```
<INVOICE CLIENT = "&IBM;" product = "&product_id_8762;"
quantity ="5">
```

An external entity must not be referenced from within element attributes. The
referenced text may not contain the < character because it would cause a parsing
error in the element when replaced.

Some more rules concerning Document Type Definitions

- No attribute name may appear more than once in the same start-tag or
  empty-element tag.
- The attribute must have been declared and the value must be of the type
  declared for it.
- Attribute values cannot contain direct or indirect entity references to external
  entities.
- The replacement text of any entity referred to directly or indirectly in an
  attribute value (other than &lt;) must not contain a <.
- There may not be any whitespace embedded in an entity reference. & SF;
  or &SF ; will cause errors.
- Entities must be declared in an XML document before they are referenced.
Other Details

As stated in the beginning of this chapter, this technical introduction does not
discuss all the details of the XML specifications. The following section will analyze
some of the other most important features of XML that have not yet been
discussed.

1. CDATA Sections

CDATA Sections are used to escape blocks of text containing characters, which
would otherwise be recognized as markup. An XML processor treats CDATA
Sections just like any character data, thus ignores all tags and entity references
within the CDATA Section. CDATA blocks have been provided as a convenience
measure when you want to include large blocks of special characters as character
data, but it is not desirable having to use entity references all the time. For
example writing a tutorial about XML would contain:

"In XML you need elements which have a starting tag <song>
and end tag </song>"

The markup for this sentence would be:

```html
<div> In XML you need elements which have a starting tag &lt;song&gt; and end tag &lt;/song&gt;</div>
```

In order to avoid this inconvenience, XML has a method to treat markup as text (or
CDATA). This is done by simply enclosing the text with markup, which we want to
displayed (not interpreted) in a CDATA element.

"<![CDATA[In XML you need elements which have a starting tag
<song> and end tag </song>]]>"

Between the start of the section, <![CDATA[ and the end of the section, ]]> , all
character data is passed directly to the application. The only string that cannot
occur in a CDATA section is ]]> . Comments are not recognized in a CDATA
section. If present, the literal text <!--comment--> will be passed directly to the
application. The character string ]]> is not allowed within a CDATA block as it would signal the end of the CDATA block. CDATA does not work in HTML.

2. White Space

White space is often used when editing XML documents to set apart markup for greater readability (spaces, blank lines and tabs). Such white space is typically not intended for inclusion in the delivery of the document. On the other hand, "significant" white space that should be preserved in the delivery is common (poetry, source code). An XML processor must always pass all characters that are not markup through to the application. How this is dealt with is up to the agent. XML provides a built in attribute xml:space that instructs the agent to preserve all white space.

xml:space is one of the few attributes that XML reserves for itself. This attribute is inherited by child elements from their root element where white space is declared. When declared, it must be given as an enumerated type whose only possible values are default and preserve.

```xml
<!ATTLIST poem xml:space (default|preserve) 'preserve'>
```

default signals that the application applies the white-space only for this element. preserve indicates the intent that the application preserves all the white space.

The rule for XML processors is that in the absence of a declaration that identifies the content model of an element, all white space is significant.

3. Language Identification

In document processing, it is often useful to identify the natural or formal language in which the content is written. A special attribute named xml:lang may be inserted in documents to specify the language used in the contents and attribute values of any element in an XML document. In valid documents, this attribute, like any other, must be declared if it is used. The values of the attribute are language identifiers as defined by [IETF RFC 1766], "Tags for the Identification of
Languages”. XML’s country and language identifiers can give the application important information that it needs for tasks like case conversion, because two countries that speak the same language may have different case conversion rules. (For example, an upper-case "è" is "È" in Quebec but "E" in France.) The Langcode may be any of the following:

- A two-letter language code as defined by [ISO 639], "Codes for the representation of names of languages".

- A language identifier registered with the Internet Assigned Numbers Authority [IANA]; these begin with the prefix i- (or I-).

- A language identifier assigned by the user, or agreed on between parties in private use; these must begin with the prefix x- or X- in order to ensure that they do not conflict with names later standard-ized or registered with IANA.

The term "tag" in the title of the Internet Engineering Task Force’s (IETF) Request for Comment (RFC) 1766 has nothing to do with the XML sense of the term. This RFC defines a standard for identifying a language using one or more words: the first identifies the language and the optional second one identifies the country in which the language is being spoken and optional additional information. The language "tag" should be the two-letter code specified in ISO 639, "Codes for the representation of names of languages". For example, fr is French, en is English, and sa is Sanskrit.

There may be any number of Subcode segments; if the first subcode segment exists and the Subcode consists of two letters, then it must be a country code from ISO 3166, "Codes for the representation of names of countries". If the first subcode consists of more than two letters, it must be a subcode for the language in question registered with IANA, unless the Langcode begins with the prefix x- or X-.

The optional second "tag" specifies the country using either an abbreviation from ISO 3166, "Codes for the representation of names of countries" (for example, BE for Belgium or US for the United States) or some other code registered with the Internet Assigned Numbers Authority (the group responsible for first-level domain
names like com, edu, and org). According to IETF RFC 1766, subsequent "tags" can be anything you like.

For example:

```xml
<p xml:lang="en">The quick brown fox jumps over the lazy dog.</p>
<p xml:lang="en-GB">What colour is it?</p>
<p xml:lang="en-US">What color is it?</p>
<sp who="Faust" desc='leise' xml:lang="de">
<l>Habe nun, ach! Philosophie,</l>
<l>Juristerei, und Medizin</l>
<l>und leider auch Theologie</l>
<l>durchaus studiert mit heißem Bemüh'n.</l>
</sp>
```

The intent declared with xml:lang is considered to apply to all attributes and content of the element where it is specified, unless overridden with an instance of xml:lang on another element within that content. In the specification's example above, the four <l> elements written in German are children of the <sp> element. Because they have no xml:lang attribute of their own to specify their language, an application must treat them as if they had the xml:lang value of de, as their parent does.

A simple declaration for xml:lang might take the form xml:lang NMTOKEN #IMPLIED but specific default values may also be given, if appropriate. In a collection of French poems for English students, with glosses and notes in English, the xml:lang attribute might be declared this way:

```xml
<!ATTLIST poem xml:lang NMTOKEN 'fr'>
<!ATTLIST gloss xml:lang NMTOKEN 'en'>
<!ATTLIST note xml:lang NMTOKEN 'en'>
```

Note that the attribute declared default of #IMPLIED in that first example makes that xml:lang attribute optional. For the poem, gloss, and note examples, including this attribute in element start-tags is also optional, but for a different reason: because defaults are supplied. If no xml:lang value is specified for any poem, gloss, or note elements in the document, they will each have the default value shown in their declarations.
XML Software

XML Software can be divided into XML editing software, XML parsers, and XML servers.

1. **Editing Software**

Several authoring tools are currently available for XML. It is clear that many previously existing editors for HTML and SGML will become capable of dealing with XML in the near future (some of the are already).

2. **Parser**

An XML parser is a processor that reads an XML document and determines the structure and properties of the data. If the parser goes beyond the XML rules for well-formedness and validates the document against an XML DTD, the parser is said to be a **validating parser**. In this case the parser checks the XML syntax and reports errors. A **Non Validating Parser** does not check a document against any DTD, but checks whether the document is properly marked up according to XML syntax rules. It reads XML files and generates a hierarchically structured tree, passing the data to viewers and other applications for processing.

3. **XML Server**

The XML server is emerging as an important integration platform for developing applications that create and send, as well as receive and process XML data. An XML server is similar to a Web server in that one could use a browser to retrieve an XML document from it. But there are two things that make an XML server different: First, it provides a standard application interface for processing XML-based information and second, it provides a variety of communication alternatives for passing the XML documents to and from the XML server. An XML server, which can receive, interpret, and generate XML documents, is really designed to communicate with other applications or other servers, not with users. The XML server can be used to automate the interchange of information between different applications, or between different organizations. For instance, a purchasing
application might send information in a standard XML format to another application, which could use it to update a product database.

An XML server can be a small, specialized application with abilities equivalent to a Perl script, or it can be a complete application development framework for general-purpose use. The major components of an XML server are: (i) Client, (ii) Communication Services, (iii) Document Handlers, (iv) Data Object Access, (v) XML Core Services.

**Client:** This can be any application that sends or receives XML documents. Clients can range from a device like a Palm Pilot, to a browser or Web server to another XML server application.

**Communications Services:** An XML server must be able to support a variety of communication protocols. Email and HTTP are the most useful, but some applications might need to multicast or broadcast XML documents to a large number of clients, or may need the guaranteed delivery of a message queuing product. Ideally, an XML server should provide a framework for any communication protocol to be used as the transport layer for an application. Multiple communication handlers can be plugged in to the communications service without changes to the XML server, the application code, or the interface. This means that a single application can be written that can handle multiple communications protocols without any changes. An XML server might support communications handlers such as HTTP, SMTP, FTP, EIB, RMI, IIOP, COM, MQSeries, and others. Java provides an excellent framework for integrating new services and reusing objects across applications. Java also leverages work from many vendors that provide core XML services as well as base-server functionality. For example, IBM, Microsoft, and Sun all provide Java-based XML parsers. The W3C has specified the Java API for the Document Object Model (DOM), one of the core interfaces that define how a program accesses an XML document.

**Document Handlers** are the key functional unit in an XML server. A document handler is a particular piece of business logic that can operate on a certain class of documents. Each document handler will usually have multiple methods-functions that can be performed on a particular type of document. A purchase-order document handler might have submit-, acknowledge-, and cancel methods, etc.
Each of these methods would incorporate application specific logic and they might call different back-end data objects such as a database.

**Data Object Access:** The document handler calls the appropriate back-end data object and passes the appropriate input fields based on the method processing. For example, it may invoke a SQL statement that inserts a new customer into the database. An XML server should have an implementation with production-level features such as connection pooling, secure database login, JDBC support, error handling, and transactioning. Access to other non-JDBC or XML data sources, including SAP, PeopleSoft, mainframes, middleware, and "new-age” objects like Java, EJB, CORBA, and COM will also be important for many implementations. When the data or objects are returned from the back end, they are then used to generate an XML document within the document handler. The returned results are put into a callback structure or into an XML DOM for manipulation. Page generation can proceed in a number of ways via a mapping or binding service, template processing, content-generation objects, or even Extensible Style Language (XSL). The key idea is to bind the data fields coming from the database to the appropriate elements in the XML document. If multiple rows of data are returned, then multiple items must be generated in the XML document in the appropriate fashion. As with the input side of this process, it may also be appropriate to filter and translate some of the data. This final assembly of the XML document is called content generation, and can involve use of XSL or reusable Java classes, called content-generation objects.

**XML Core Services:** The XML server uses core XML services such as a parser and DOM, as well as XSL and XML Query Language (XQL). These provide the core services for parsing the incoming document to determine the appropriate class and method to call, as well as getting the data out of the

A dynamic XML server is very powerful tool. Just a few lines of Java code can perform a complete query to the database and generated a well-formed XML document, with automatically generates DTDs. The XML server takes care of the parsing using a standard XML parser. The XML server also takes advantage of reusable Java components like the content generator.
Appendix 2: XML/EDI Working Groups

The initiative body XML/EDI group was founded in July 1997 to specifically work on the broad use of XML/EDI in the future e-business. Further XML/EDI groups were formed in order to work on specific topics of interest.

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997 July</td>
<td>XML/EDI Group was founded</td>
</tr>
<tr>
<td>1997 October</td>
<td>XML/EDI Guidelines released</td>
</tr>
<tr>
<td>1997 December</td>
<td>W3C Working Group finalized v1.0 of XML</td>
</tr>
<tr>
<td>1998 March</td>
<td>BizServ provide Web hosting</td>
</tr>
<tr>
<td>1998 July</td>
<td>CEN Pilot with EDIFACT started</td>
</tr>
<tr>
<td>1998 September</td>
<td>Repository Working Group launched</td>
</tr>
<tr>
<td>1999 December</td>
<td>ebXML initiative established by UN/CEFACT &amp; OASIS</td>
</tr>
<tr>
<td>2000 December</td>
<td>ebXML technical architecture specified</td>
</tr>
<tr>
<td>2001 February</td>
<td>Integrates SOAP into ebXML Messaging Services Specification</td>
</tr>
<tr>
<td>2001 May</td>
<td>Final ebXML meeting</td>
</tr>
</tbody>
</table>

Figure 20: XML/EDI Historic Highlights

The idea of the XML/EDI working groups is to develop open standards, so no major player will establish proprietary standards. The aim is to document techniques that will allow existing EDI messages to be converted to XML for display on standard Internet document browsers, and for converting XML messages based on EDI-compatible data components into appropriate EDI messages. At the same time the group is looking at how data from XML messages can be integrated with the databases and internal business processes used at a client (data generator) site, something which is currently difficult for EDI users. The main thrust of the XML/EDI Group approach is that the translation process should be as automated as possible and that the processes developed should be sufficiently simple to be implementable by an SME who does not have in-house IT expertise (Bryan 1998a). The XML/EDI Group is working with Data Interchange Standards Association and CommerceNet to map the X12 standard used for EDI in North America to XML. The XML/EDI Group also cooperates with the Unified
Modeling Language Group, which will enable people to model their business objects, map them to XML DTDs, and exchange complex objects via HTTP.

**The European XML/EDI Pilot Project** is one of a number of projects being undertaken by the Electronic Commerce Workshop of the Information Society Standardization System within the European Committee for Standardization (CEN/ISSS). The project is open to all interested parties (Bryan 1998b). A number of European EDI projects are currently reviewing the relevance of XML to their work, especially in the following areas:

- Statistical data interchange, as being employed at EuroStat and in a proposed pilot project in the Norwegian public sector.
- Transport services data interchange for the interchange of messages between transport companies and their clients. This project is of big relevance throughout Europe.
- Healthcare informatics which is responsible for the standardization of healthcare EDIFACT messages.
- Electronic forms, as being developed by EDIFRANCE
- Simple EDI as developed by UK/CEFACT

The XML/EDI has not developed anything new lately, but the people behind this movement (Bruce Peat, Martin Byran) are the driving forces behind the ebXML initiative. New Consortia like OASIS and RosettaNet are under way that have established a variety of XML Frameworks like biztalk, ebXML, cXML lately.

**ebXML** is an international initiative established by United Nations (UN/CEFACT) and OASIS in late 1999 with a mandate to undertake an 18-month program of work to research and identify the technical basis upon which the global implementation of XML can be standardized. The vision of ebXML is to enable a global electronic marketplace where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML based messages. ebXML aims to be a set of specifications that together enable a modular electronic business framework (ebXML 2001).

Business Processes (expressed in XML) will define the core models which are stored in a repository, containing business messages also expressed in XML.
Trading Partner Agreements (expressed in XML) will specify parameters for businesses to interface with each other. The Business Service Interface (expressed in XML) implements Trading Partner Agreements. Transport and routing layer move the actual XML data between trading partners. Registry and Repository will provide a container for process model vocabularies and partner profiles. The ebXML specifications are stored in these common repositories and can be downloaded from any company, which then can modify the ebXML specifications to its own business needs and register its own company profile with the repository. Any other company interested in conducting business with this company will be able to download the other company's profile and agree (ebXML 2001).
1. Vertical XML Applications

Vertical market vocabularies serve a defined industry. This group best illustrates the challenge of interoperability facing the XML standards community. Firstly, some vocabularies cover only a single transaction, while others are collections of functions for industries, sometimes even within the same industry. For example, one of the BizTalk schemas is an Asset Financing Credit Application, while more general financial services languages also have this function.

Second, industry groups need to reconcile conflicting or overlapping specifications. At least three separate specifications claim to serve the human resources community. Any hope for achieving interoperability needs to begin within these industry groups.

---

Examples for Vertical Industry Applications (Kotok 2000)

- XML for the Insurance Industry
- Chemical Markup Language
- Electronic Health Record
- Financial Products Markup Language
- Hotel Electronic Distribution Network Association
- Hospitality Industry Technology Integration Standards
- XML Schemas for Insurance
- Instrument Markup Language
- NewsML
- Open Financial Exchange
- Weather Observation Markup Format
- Real Estate DTD Design
- Open Software Description Format (OSD)
- Open Settlement Protocol
- OpenTravel Alliance
- XML for the Automotive Industry
- Schools Interoperability Framework
- 'SmartCard' Markup Language
- Telecommunications Interchange Markup
- Tutorial Markup Language
- UML Exchange Format

---

Some industry groups are beginning to resolve these differences. In the travel industry, the Open Travel Alliance (OTA), decided to incorporate much of the Hotel Electronic Distribution Network Association specification in the first version of its
standards. OTA is also negotiating with the Hospitality Industry Technology Integration Standards group to include its guidelines.

The Open Trading Protocol (OTP), while claiming to be an interoperable framework for Internet commerce, OTP covers XML for Web retailing, rather than presenting a general approach for business. As a result it falls under vertical markets, although it has more general specifications than most entries in this group.

2. XML Application Functions

The Functions category covers a range of common business operations. Many of these guidelines resemble EDI transaction sets or messages (XML/EDI Repositories), such as purchase orders, order acknowledgments, and invoices. Others in the group perform message routing and management functions such as Information and Content Exchange, and the Directory Services Markup Language. Security and privacy functions, such as the W3C’s Digital Signature and P3P, the Platform for Privacy Preferences (W3C 2000c), fall into this category as well.

- Business Rules Markup Language
- Common Business Library
- Signed Document Markup Language
- Simple Object Access Protocol
- Trading Partner Agreement Markup Language
- Universal Commerce Language and Protocol
- Universal Standard for Data Synchronization
- UN XML EDI/EC Work Group - XML/EDI
- Visa Global XML Invoice
- WAP Wireless Markup Language Specification (WML)
- XEDI
- XML-Based Process Management Standard
- XML Digital Signature
- XML/EDI Repositories

Examples for XML Application Functions (Kotok 2000)

Common Business Library (CBL) provides the syntax and semantics for interoperability. While certainly beneficial to the development of XML business messages, CBL does not provide an overall message structure like BizTalk, eCo Framework, or that which ebXML has on the drawing board.
Another entry in this group that straddles the categories is the Extensible Financial Reporting Markup Language, which falls under financial services but covers common reporting functions required by most businesses.

3. XML Application Frameworks

XML frameworks have probably received the most public attention, since they offer the most potential for interoperability. Examples for such frameworks are: BizTalk (Microsoft) and eCo Framework (CommerceNet), who have both announced their specifications, as well as ebXML (UN/CEFACT and OASIS), which is still in the planning stages. This group also includes collections of related document type definitions (DTDs) and schemas, such as those offered by the Open Applications Group and Commerce XML (cXML). The XML/EDI Group's guidelines are likewise included in this category, as well as the related XML/EDI workshop under the aegis of the European Standardization Bureau.

- BizTalk Framework
- Commerce XML
- Distributed Management Task Force
- eCo Framework
- Electronic Business XML Initiative
- European XML/EDI Pilot Project
- An Open Standard for Small to Medium sized Businesses
- Open Applications Group
- RosettaNet

List of for XML Application Frameworks (Kotok 2000)

As with any categorization scheme, some entries do not fit cleanly in the boxes. RosettaNet (2001) is a good example. While RosettaNet is an undertaking of the computer technology industry, and therefore a candidate for the Verticals group, its approach of defining and modeling business processes, and then developing specifications, offers a framework for most industries to follow.
Appendix 4: Black-Scholes Option Pricing Table

This table shows call option values as a percent of share price based on Black-Scholes model. To obtain corresponding European put values, add present value of exercise price and subtract share price.
References


Accessed: April 12, 2001

Accessed: May 2001


http://www.xml.com/pub/a/2000/08/02/ebiz/extensible.html
Accessed: January11, 2001


http://www.unece.org/trade/untdid/welcom1.htm


Available: http://www.wiwi.uni-frankfurt.de/~tweitzel/XMLEDI.doc


Available: http://www.geocities.com/WallStreet/Floor/5815

Available: http://www.xmledi.com/repository
Accessed: March 16.
