Creating a Virtual Products Market Based on Open Standards

An Investigation on Legal Aspects of the XML Web Services Market

Masters Thesis, 20 p.
in Applied Studies for Examination at the
National LL.M. Programme (jur.kand.)
Law Department
Gothenburg School of Economics and Commercial Law
University of Gothenburg
Autumn 2003

Pontus Sörlin
70 12 11 – 5098

Tutor: Ulf Petrusson

Glossary

API An application program interface (API - and sometimes spelled *application programming interface*) is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application.

B2B (business-to-business) A characterization of interaction between businesses over the Internet. In B2B interaction, applications between businesses communicate without human intervention being required during the interaction. Contrast to B2C (business-to-consumer).

B2C (business-to-consumer) A characterization of interaction between a human being and a business over the Internet. In a B2C interaction, a human being, often using a web-browser, invokes applications provided by a business. Contrast to B2B (business-to-business).

GUI (graphical user interface) A graphical (rather than purely textual) user interface to a computer.

HTTP Hypertext transfer protocol The set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

OASIS Organization for the Advancement of structured Information Standards

Protocol The special set of rules that end points in a telecommunication connection use when they communicate.

SOA (service oriented architecture) An abstract pattern that applies to a wide variety of XML Web Services situations.

SOAP (Simple Object Access Protocol) A lightweight exchange of information for exchange of information in a decentralized, distributed environment. It is an XML-based protocol.

UDDI Business registry An instance of the UDDI registry hosted at www.uddi.org

XML Extensible Markup Language A flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

WSDL (Web Services Description Language) A component of a service description that describes the interface definition of the XML Web Service, details related to binding (network protocol and data encoding requirements), and the network location of the XML Web Service.

1. Introduction

Internet or Cyberspace is characterized by that there are no geographic limitations, anonymity, ability to escape controls, structural hierarchies and zones, interactive and dynamic nature, electronic linkages.[1] An actor that intends to build a market in this environment will have some help from the state-based legal community in building the market. A market is a place where goods and services can be sold and bought in order to satisfy the need for safe trade and to minimize risks among the actors involved in the market. One way that has been used in market creation is to build markets based on standards. Standards are in essence technical specifications that have been achieved either through sole actor achievements or in cooperation among several actors. Closed standards were previously a favored concept, now giving way for open standards. The closed standards were aimed at keeping or making it more difficult for competitors to enter the market. Open standards are inclusive, which means that anyone who wants to contribute to the standards and use it is able to do that. An interesting question when you are dealing with these kinds of environment is how you construct virtual markets based on open standards? From a legal perspective several questions arise. What kinds of values are of interest of protecting in a market of that kind? What kind of legal tools are available to use on such a market? I will investigate a certain market that I believe is at the core of all these issues.

The Internets possibilities to create markets have been thoroughly discussed. Mostly the discussion has been about marketplaces or marketspaces like online auctions and online trading. The focus of interest has more often than not been how legal ties between the involved parties are created *within* the boundaries on the marketplace.[2] In this essay I will concentrate more on the outside elements that make up the foundation for markets of the previous kind. It is not however meant to be a discussion of how these earlier markets are created. I will concentrate on software markets, since I find it to be a more relevant issue in the context of the essay.

The creation of a virtual products market based on open standards involves the concept that the market and its construction cannot be viewed in isolation. It is basically a cooperation of legal, technical and business issues that decides what the market will look like. However, I will adjust this view and use Lessigs notion of code as law of how to analyze legal issues on

[1] Longworth, (2000) pp. 15-18
[2] See, for example, Ramberg (2002).

the Internet. This means that this essay will treat laws, technology, and business processes in a coherent manner. These areas are possible to view through a legal perspective altogether. But they should be discussed together and not be left out, since they all affect the construction of the market.

I will describe and analyze a specific market – the market of XML Web Services. It is a software market in essence that has been enabled through the use of open standards. In order to describe and analyze this market it is necessary to conduct an extensive investigation on its legal, technical and business parts.

2. Purpose of the essay

The purpose of the essay is: to investigate from a legal perspective how a market can be constructed on open standards. This can be done by analyzing some of the legal pitfalls that are enabled by the introduction of the market.

My more specific questions within this purpose are:

- Are parties that engage in automatic transaction in a standards-based system obligated towards each other?
- What changes may XML Web Services create in creating a solution to diminish illegal copying of software?
- Are there risks for XML Web Services to be subject to patent infringement?

I have a proactive approach, in that I believe that legal problems can be avoided or minimized by anticipating them. By discussing questions that are interrelated a better judgment can be made, I believe, when making choices on a legal strategic and tactical level. My intention is that this essay can serve as an introduction into the complex world that surrounds virtual product markets based on open standards.

I hope that I can make visible some of the strategic angels that are possible to act on for an actor in a situation of a similar market creation situation. At the very least an actor could get a view on what kind of possibilities he may have from a legal point of view in this setting.

This essay could be of benefit for software developers to develop checklists that they may use in their legal strategy in a start-up situation or to change a company's strategy in order to be prepared for developing Web Services. By creating a view of the legal considerations that can

serve as a basis for the market construction this essay is aimed at being a starting point in such a plan.

The study would also be of benefit for legal researchers in order to relate Web Services and their technical impact on the legal system and vice versa. The study would also be of benefit to studies of how legal structures are developed.

This essay is not a quantitative analysis in order to statistically determine what has or has not been done in a number of situations. Neither can I claim to make a comprehensive recollection of what has actually happened between the real actors of the XML Web Services market. I cannot even claim to have made an exhaustive list of the possible strategic choices that can be made. I can only account for the choices that I have considered to be the most relevant ones.

This essay will mainly focus on the relationships between companies, or B2B-relationships. Aspects of B2C and B2B2C will also have some place in the essay, but that is mainly for discussion purposes.

3. Method

My analysis of this area has comprised of a vast material. The material consists of literature and relevant material from the Internet. The literature and the relevant material cover legal, technical and economic matters. The multi-institutional approach had otherwise not been possible to implement.

4. Legal justification

New technological advances create new structures that will have an impact on legal structure. The legal structures are also in constant change. Even if much of legal structure is taken for granted in business, in the form of contract law, tax law and international commercial law, for instance, actors themselves are actually creating some of the legal structures. These legal structures are not easily analyzed, since the material is less public and current than that of public institutions.

Ordinarily, a legal essay will investigate the law as other scholars have interpreted it and how the legislature and its institutions have laid it out. In a field like Web Services, there are new

institutions being constructed and these institutions are part of creating the new market. Since these kinds of institutions are not part of the ordinary legislature, they lie outside the ordinary viewpoint of the business lawyer that is focusing on his role in front of the court. That is a fruitful perspective in cases that can be solved within the court system. The court view is however, not that fruitful when it comes to understanding how an internet market like Web Services is constructed. That is why I am going to take a different approach in this essay.

An alternative way of studying these structures and their creation is by using theory. There are a number of legal theorists that outline the legal underpinnings of the Internet environment. Those include David Johnson, Ethan Katsch, Lawrence Lessig, Darrell Menthe, Henry Perritt and David Post.

In this essay I have chosen to use a view on phenomena on the Internet that is proposed by Lessig: that a phenomena on the Internet can be viewed as regulated by law, market, architecture and norms.[3] These items are seen as restraints on an actor's behavior in the cyberspace environment. Law is seen by Lessig to regulate behavior through commands, markets create incentives for actors to behave in certain ways, social norms threaten non-legal sanctions for particular behaviors and architecture constraints the set of possible behaviors. One of Lessig's main points is that computer code regulates the behavior of actors on the internet since the code that is programmed by programmers decide what actions that are *possible* to live out on the net. A simple example: an unlocked door to a house may create a possibility to enter the door (architecture), but most actors will not enter simply because it's wrong (norms), some actors do not think it's wrong to enter, but do not enter by fear of breaking the law (law), a market that is in a recession may create incentives to people to enter, since there is high demand on goods on the other side of the door (market).

I will structure the essay according to these 4 dimensions, since I think that it will be a fruitful way to describe and analyze the situation I am investigating. The law part will consider legal aspects of XML Web Services, the architecture part will describe technical parts of XML Web Services, the norms section will discuss norms that are of interest for XML Web Services, and market will describe market relevant issues in conjunction to XML Web Services.

[3] Lessig (1999) pp. 88, 165

I will focus on actors in the essay. The actors in the Web Services market construction are all the parties that in some way contribute to the building of the market. Actors can be named, for example, as specific companies. They can also be a group of companies, like a consortium. Most actors have several roles at the same time. A company can be a context provider at the same time as it is developing a platform or is an end-user.

There is a danger present if the actors believe that the legal issues have been solved when a technical solution is available. Just because the business processes are embedded in technology, in this case computer code, does not mean that the legal issues goes away. The code in this case is merely one restricting factor on behavior for companies that decides to use XML Web Services for business purposes. Law is another restriction, as well as markets and norms.

An interesting actor is the developer of XML Web Services. Besides from the lawyer, the developer should be very interested in the perspectives laid out in this essay. The reason for this is that a XML Web Services developer will be interested of how to create code that will adhere to the norms, the architecture, the law and the market of XML Web Services. At the same time, a developer should be aware of his important role of creating code is a way of creating legal phenomena in what can be called cyberspace. The lawyer should be aware and work together with other actors, such as the developer, in order to create a better understanding for what creates good legal systems. Alone, no actor will be able to make it.

5. Introduction to XML Web Services

As the internet-transformation of business and society had progressed during the latter part of the 1990's, it became obvious that the Internet had a number of shortcomings. It was not prepared for the kind of use that it was receiving. There was also many uses that were wanted by the actors on the market couldn't be implemented because of how the specific technical nature of the Internet was constructed. In order to move forward, what was needed was an Internet that could be utilized in a way that made it easier for computers to communicate and exchange information.

In 2000, Sun announced XML Web Services to be the answer to all these problems. Today, in 2003, all major commercial actors around the world agree that XML Web Services is the way forward.

In Sweden, most software producers in Sweden are familiar with and are using Web Services, and 40 % of all IS (information system) projects are run using Web Services.[4]

Gartner estimates that 75 % of businesses with more than USD 100 million in sales will use Web Services in some business function. AMR Research reports in a survey that when assessing their 2003-2003 strategic IT budget 250 manufacturing firms, the top 2 or 3 most important initiatives was integration of enterprise applications with Web Services. IDC predicts that the market for XML and Web Services will reach USD 5 billion by 2005. The total market for the software market is USD 15 billion by 2005 according to Gartner.[5]

Thus, there seems to be a great potential being attributed to the XML Web Services market. This technological transformation is being described as the greatest technological advance since the break-through of the Internet.

What I am interested in this essay is to understand how a market of this kind is being created. I will make the discussion from a legal perspective, because I believe that the legal perspective is a fruitful way of discussing a market of this kind. In order to discuss and understand the XML Web Services market and how it was constructed, it is necessary to understand quite a few of the concepts of XML Web Services. I will discuss and describe the market components and at the same time make reference to a legal parallel discussion to put the technological and market components into legal perspective.

It is important to keep in mind, that from a legal perspective and in correspondence with Lessigs view of the 4 dimensions of regulation put forward earlier, that even though there will be a description of technical and market terms, these descriptions are part of a regulatory environment that is possible to envision when viewed as a totality.

6. Market

The market dimension is the first of the four dimensions that I will describe. I will discuss the XML Web services market, its components and how it fits into a market theory.

[4] Söderberg, Johan & Zenno, Patrik (2002) p. 50
[5] Deloitte & Touche (2002)

6.1 What is a market?

In order to explain how a market is constructed, it is an advantage if you have a perception what can be. Goods and services need to be sold by the suppliers to buyers. In order to simplify the exchange between sellers and buyers a market is provided. The market also fills an important function of providing a regulative framework in order for the market to be efficiently managed.[6]

The market is regulating how business is being conducted in that particular market. Most markets have in common that laws and regulations has an important part of the market functions. Laws and regulations are handled on state-level and comprises of, for example, export- and import regulations, IPR protection, and commercial code. Custom is often used in order to fill out areas where state-based regulation is not covering the area. Law of contract gives the opportunity to regulate the content of agreements at the parties' disposal. This means that the market actors are free to create rules that are governing the market. The participants are agreeing on language and definitions on the market. They agree on kinds of services and services that should be exchanged. They also decide how conflicts are going to be handled between parties on the market.

Closely connected to marketplaces are marketspaces, which are electronic marketplaces, and especially Internet-based marketspaces. These kind of marketspaces changed processes previously used in trading and traditional supply chains. The process of doing business in a virtual world differs from the real world. Instead of processing raw materials electronic commerce involves gathering, selecting, synthesizing and distributing information. That is one reason for why the economics of electronic commerce is different from traditional economic models, including supply and demand, pricing and competition. [7]

In a marketspace buyers and sellers exchange goods and services for money, but it all happens electronically. The major components and actors of a marketspace are digital (or virtual) products, consumers, sellers, infrastructure companies, intermediaries, support services, and content creators.[8]

[6] Bakos (1998) p. 35
[7] Turban (2002) p. 46 f
[8] A.a. p. 47

6.2 E-business

The construction of the XML Web Services market is taking place within a large movement towards automation of business processes. It is thus a major part of what is called electronic commerce or E-commerce. Electronic commerce has a broad range of definitions. There is also the notion of E-business, which has a clear resemblance to the term e-commerce. E-business can be defined as: "… the transformation of key business processes through the use of Internet technologies.".[9] Another definition is: "… when a business has fully integrated information and communications technologies (ICTs) into its operations, potentially redesigning its business processes around ICT or completely reinventing its business model … e-business, is understood to be the integration of all these activities with the internal processes of a business through ICT.".[10] Viewed in this perspective e-commerce can be seen as a subset of e-business, as e-business has a more holistic and theoretical application when compared to e-commerce. E-business has a close relationship to the notion of the b-web, while e-commerce has a more limited scope.

The area of e-commerce can be regulated to some extent by e-commerce law. E-commerce law regulates issues such as when agreements are concluded over an electronic exchange. Some of this legislation is mandatory, but some of it isn't, which means that if an agreement is regulating a transaction, the transaction will be regulated by the agreed upon terms. It then becomes important if the contract is valid. I will discuss issues in this essay that touches on both e-commerce and e-business.

6.3 Software markets

For a technology like software, the market is sales of software or more accurately the rights for using software.

Start-ups, smaller companies or even individuals can make production of software modules and components. These have in common that they have limited investments in downstream commercialization capabilities. Therefore they choose to license their components to larger software producers.[11] By licensing software it is possible to focus on technology issues instead of becoming a supplier or a competitor to other players on the market for products.[12]

[9] http://www-3.ibm.com/e-business/index.html
[10] UK Department of Trade and Industry (DTI) (2000)
[11] Arora, Ashish, Fosfuri, Andrea & Gambardella, Alfonso (2001) p. 57
[12] A.a. pp. 58-59

Deals that include software can be classified into four groups. The first is a transfer of market rights. A company who has developed software let other companies marketing or distributing the software. The second is transactions in software components. The components are typically included into larger software systems or other technologies. The third is transactions that are transfers – "… exchange or integration of technologies or supply of software tools such as EDA tools; tools for drug research, testing, and screening; or other R&D and testing tools. In some cases, this involves integration of the two partner's software technologies or of the software technology of one partner into the technological system of the other, with further development of the technology. Design and R&D tools are the objects of the other type of transactions in this group.". The fourth group is software transactions that are licenses for transfer of rights to use a software product. This is typically how software is sold in the final market.[13]

6.4 Value-chains

The market for XML Web Services is relatively new and under constant progress. Without the connection to a theoretical framework it is hard to conceive how this market is being constructed. The reason for this is that the XML Web Services market is not working as a traditional industrial value-chain. A value-chain is a theoretical model on how actors are creating value sequentially through an industrial process. A value chain is of importance in order to build a business model. The business model is a description on how a company will generate revenues from its business.

The theoretical background for business models has changed with the advent of the "new economy" at the late 1990s. Above all the view of value has changed. The traditional industrial view on how value is created sequentially through the value chain has lost some of its significance in order to explain the emergence of value.

The older type of value chains emerged in a time with hierarchies that were supply-driven, division of labor in mass-production environments with long planning cycles in combination with stable industrial orders. This older model created crises that created process-innovation in the form of business process reengineering and total quality management, for example. A problem with these types of solutions was that they often simply led to cost-cuts. The crises

[13] A.a. p. 60

were also attempted to be solved through structural innovation like the virtual organization, outsourcing and keiretsu. Keiretsu is a semi permanent phalanx of companies bound by interlocking ownership and directorates. They were successful in creating a strong us-versus-the mindset. The Keiretsu proved to be discredited in Japan, but became a fashionable description on all kind of partnerships among Silicon Valley type firms. This was before the Internet emerged.[14] Another solution that were presented by Moore was that when the internet had been developed, a manager should see themselves as parts of organisms that are part of a bio-system or an ecosystem. This organism can be a process or a department in the company or the whole company.[15]

Moore's definition of the ecosystem is: "An economic community supported by a foundation of interacting organizations and individuals – the organisms of the business world. This economic community produces goods and services of value to customers, who are also members of the ecosystem. The member organisms also include suppliers, lead producers, competitors, and other stakeholders. Over time, they coevolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies. Those companies holding leadership roles may change over time, but the function of the ecosystem leader is valued by the community because it enables members to move toward shared visions to align their investments, and to find mutually supportive roles.".[16]

Moore pointed out that the Internet economy had big similarities with the ecosystem. The ecosystem has contrary to a traditional value-chain an inclusive strategy. The ecosystem is moreover self-organizing.[17]

The XML Web Services market is a type example on how a market has difficulties of being discussed in terms of a traditional value-chain. In the XML Web Services market there is a multitude of actors that sell and buy value from each other in unpredictable complex combinations.

This kind of value chain has also been described as a "hub" that handles relations between companies. The placement and use of electronic markets within an industry then changes the process and the structure that were present in the value chain. In this case the hub is driven by

[14] Tapscott (2000) pp. 13-17
[15] Moore, James F. (1996) p. 25f
[16] A.a. p. 26
[17] Turban (2002) p. 51

an "infomediary" that is an electronic middleman that handles the flow of information in the hub.[18]

Porter's view of the traditional value-chain seems to get a diminished importance. At the same time Porter has shown ways of adjusting the traditional value chain to an environment where Internet influences processes. Porter has attempted to show that the value chain can still remain incorporated within the traditional value-chain.[19]

After all, the Internet seems to be able to change the value-chain. The discussion has previously only revolved around marketing and sales.[20]

6.5 The b-web

Due to the fact that the XML Web services market is continually developing, there is a difficulty on how to describe it. I have therefore concluded that to use a model of markets developed by Tapscott to simplify the categorization of the XML Web Services market. Tapscott takes his starting point from the notion of a business web – a b-web. The definition of a b-web is:" … a distinct system of suppliers, distributors, commerce service providers, infrastructure providers and customers that use the Internet for their primary business communication and transactions.".[21] A b-web can behave differently, sometimes show a high level of structure, sometimes it behaves amorphically. The participants of a b-web are joining the b-web in order to create value. In the most elegant type of b-web every participant focuses on what he does best.[22]

In order to make a holistic perspective on a market of this kind, I will use Tapscott's theory on b-webs. I will not develop Tapscott's model, but I will use the model as a point of reference for the reader to make an image of what the XML Web Services market looks like.

According to Tapscott a b-web has nine features:

1. Internet infrastructure. B-webs use the Internet as their primary infrastructure for business communications and transactions.

[18] A.a. p. 61
[19] A.a. p. 64
[20] Hsu, Cheng & Somendra Pant (2000) p. 50
[21] Tapscott, Ticoli and Lowry (2000) p. 4
[22] A.a. p. 17

2. Value proposition innovation. A b-web delivers a unique, new value proposition that renders the old way of doing things obsolete.

3. Multienterprise capability machine. A b-web marshals the contributions of many participating enterprises. B-web leaders rely on partners to maximize return on invested capital.

4. Five classes of participants. A typical b-web structure includes five types – or "classes" – of value contributors.

– Customers, who do not receive but also contribute value to the b-web

– Context providers, the interface between the customer and the b-web. A context provider leads the choreography, value realization and rule making of the system.

– Content providers design, make and deliver the "intrinsic" forms of value – goods, services, or information – that satisfy customer needs.

– Commerce services providers enable the flow of business, including transactions and financial management, security and privacy, information and knowledge management, logistics and delivery and regulatory services.

– Infrastructure providers deliver communications and computing, electronic and physical records, roads, buildings, offices, and the like.

5. Co-opetition. B-web participants cooperate and compete with one another.

6. Customer-centricity. Rather than making, then selling, b-webs focus on customer value. They build mutual relationships and respond to individual customers at the point of need.

7. Context reigns. The context provider manages customers' relationships and choreographs the value-creating activities of the entire system. Such b-webs leaders get the captain's share of the spoils.

8. Rules and standards. Key participants know and adhere to the b-web's rules of engagement.

9. Bathed in knowledge. B-web participants exchange a variety of data, information and knowledge. [23]

I will discuss some of these sections more in detail in the essay. The model gives a broad view of how the XML Web services market can be envisioned to be constructed at a theoretical level. Note item 8 – rules and standards. They are in Tapscott's model a separate part of the market, but in this essay it is treated as a separate dimension.

[23] Tapscott (2000) p. 19

6.6 The actors of the market

The actors are the players on the market in different ways. I will describe them shortly, and concentrate on the context provider, which is the most interesting actor in the market section. Note that I have omitted infrastructure providers that Tapscott mentions. The reason for that is that in the XML Web Services arena the physical aspect of the market is more or less missing, since it is a virtual market.

6.6.1 Customers

The customer is the end-user of a XML Web service. The end-user rents the right to use the software that is comprised within the XML Web Service. Since the XML Web Service is not installed at the end-user, an XML Web Service can be easily and swiftly used by the end-user.

6.6.2 Content providers

A content provider is the classical provider of multimedia assets. These assets will still be very important items on the XML Web services market, since they will be the content that many XML Web Services will provide as a part of a service. However, for a content provider, the XML Web services market is not the only market. For example, a content provider that provides stock quotes, can offer them through a number of channels, XML Web Services being one of them. As the XML Web Services market progresses, it should create an increasingly interesting option also for content providers, however.

6.6.3 Commerce services providers

The commerce services providers enable the flow of business, including transactions and financial management, security and privacy, information and knowledge management, logistics and delivery and regulatory services. This group will be the most important group initially in XML Web Services, since they produce the core services that can be integrated into different systems. The commerce services providers produces the basic XML Web services that create the functions that can be utilized by a context provider or by an end-user directly.

6.6.4 Context provider

A context provider puts together one or several XML Web Services from developers and offers them in a package to the end-user. The context provider is thus packaging XML Web services. This position is an important one since it is the main connection between the developers and the end-users.

The role of the context provider is to connect developers and end-users, i.e. a traditional brokerage role. One scenario would be that an end-user would be able to rent a Web Service for a limited time from the context provider. The context provider offers services and handles the transactions with the end-user. Increasingly, the context provider will be using XML web Services to manage payments and other services. In that way the b-web is shaping around the context provider and the actors of the market are being increasingly interconnected.

XML Web Services often means that you offer database-connectivity to a Web Services interface. That means that you provide an API to your own database. An API is an opening to a computers code that enables an outside developer to access tools owned by the system that the service wants to use. Examples of this are Amazon and Google who have released developer's tools in order to connect to their API and use their databases as basis for new applications developed by third party developers.

The context provider is central in a b-web. Without a context provider the other actors cannot discover each other in an efficient manner. In order to create value it is necessary to have a context provider that has a hub-position. A context provider is in a third-party relationship to buyers and seller like a broker is. The context provider is similar to the position of the portal provider, the difference being that an XML Web Services context provider does not need a GUI for a human to access, like a portal does.

Amazon is an actor that has several of the roles embedded into its business. Amazon is here an example of provider of both content and context. Amazon is also offering XML Web Services for "sub-contractors" that may plug-in to Amazons API and set-up a web-store, or to make their own web-applications based on Amazons API. This solution would not have been possible without XML Web Services.

Amazon has a EULA that pops up when you download an SDK in order to develop XML Web Services. In the EULA there is a description that Amazon.com offers Amazon.com Web Services on the approval of the EULA. It is established that Amazon Web Services is a platform for Web Services that can be used for different uses.

The user is given a limited revocable, nonexclusive license. The license is issued for all data, content, tools or other information that can be delivered through the Amazon Web Service.

There is also detailed instructions on what you are allowed to do with the information that is received through Amazon Web Services. For instance, the developer is allowed to develop an application that is connected to Amazon Web Services.

Amazon makes possible through Web Services that other actors can connect to their databases. This possibility was not an option before XML Web Services, unless if you focus on small-scale connection through pre-negotiated agreements by a small number of actors. In their role as content provider, Amazon is now offering access to their databases consisting of titles, text and images and other information to a very large number of users. Other context providers can offer this material directly by connecting the Amazon database. Amazon is also providing their databases via their own homepages. By providing others to connect to their databases Amazon are allowing other actors to create their own value that can be added to the actors own applications and to the b-web as a whole.

7. Architecture

As we have now discussed some of the more basic notions of the XML Web Services market, I will now get into the legal/technical architecture of the XML Web Services market.

7.1 Internet

The Internet that the XML Web Services market is built on, consists of two layers – a physical and a virtual. The physical layer consists of hardware like computers, teleoptic fibers, etc, which would correspond to Tapscott's position of an infrastructure provider, which is not discussed in this essay. The physical layer creates the basic possibilities for information to travel through the network. The virtual layer consists of different kinds of software. Software are applications like Microsoft Word, or any other program you use on a computer. Software is also operating systems like Windows or MacOS. The operating system is needed in order to run the programs on a computer. The virtual part of the virtual layer is making use of so called virtual products. In its simplest form a virtual product could be a database. Digital products have different cost-curves than regular products, as in digitization most of the costs are fixed and the variable cost is very small. Profit will therefore increase very rapidly as volume increases once the fixed costs are paid for.[24] For our purposes it is enough to say that software are a part of a number of products that can be categorized as virtual products.

[24] Turban (2002) p. 47

In its simplest understanding an XML Web Service is a virtual product. That means that the XML Web Services market is a virtual product market. A digital or virtual product is dependent of a number of prerequisites: digitalization, medial convergence, interactivity, international computer networks, automatisation of information, the disconnection between information from the bearer, flexible informational structures, simplicity in process and change.[25] Databases, news and information, books, magazines, movies, games, music and software are delivered as virtual products over a digital infrastructure, worldwide. Financial transactions are carried out through digital currencies or on smart cards via networked computers and mobile devices. Physical goods like home appliances are enabled with microprocessors and networking capabilities. The market for virtual products is therefore quite large and diverse.

The protection of virtual products has been thoroughly discussed. Copyrights and trademarks can protect them, but they also take their protection in company secrets, computer patents and complex licensing contracts. The combination of different resources into new products is difficult to handle in legislation, since the virtual products do not match the law requirements entirely. This means that new ways of protection are emerging to make up for this lack of coherence. Sometimes this takes the form of virtual techniques in the form of copy-protection software for example that can assist legislation in protecting the intellectual property rights assigned by legislation. This is an example of how law is not enough, but the need for regulation of behavior by means of architecture is also used.

In different virtual products some of the prerequisites listed above may be more important than others. The combination of different content on for example a website can be called a multimedia product. The multimedia product as such consists of a number of copyrighted materials, for example text, music or pictures. There might also be software that is a part of the product that is protected by copyright. Often there are also several brands involved as several companies are involved in the production process. The brands themselves are protected through the brand legislation. Today the term multimedia seems rather outdated. It also seems to hint at only the aspect of media convergence.

The virtual products can be seen as a totality of relations, IPRs, and values that are not related to IPRs in a b-web. It is therefore difficult to separate the virtual product from its market.

[25] Det IT-rättsliga observatoriet. Rapport 9/98. p. 9–12

This essay will therefore treat the virtual product and its market as two aspects of the same phenomena, which limits are difficult to define. Since the purpose of the essay is not to determine where the limits are between the virtual product and the market it is sold on, this will not be a problem. This view is accentuated by the fact that the market actors have difficulties in defining what XML Web Services is. That can be interpreted as most of the actors are tracking a phenomenon that for the sake of simplicity can be called XML Web Services.

7.2 XML

The XML Web Services market is built on a computer language called Extensible markup Language (XML). A markup language is a method of adding information to structural elements indicating the logical components of a document. Structural elements can be content, like text or images. A common markup language is HTML, which is the language that is used in order to create homepages. HTML tells the web browser how to display text or images on the homepage. HTML is a simple markup language that does not allow for much dynamic use, because it only give you one way to structure content – it is defined within HTML. XML is used to identify structures that define data within a document. XML is therefore a meta-language that defines rules for other markup languages.

Compared to a law, HTML can be compared to a fairly specific law like the traffic legislation. It tells what you can do and what you cannot do in fairly specific terms. There are not many unclear terms in the law and what behavior is allowed or not is easy to determine. There are some situations where such a markup language or a law is too coarse to regulate the behavior of content or people, because the situation may be unclear, badly defined, and there is a need to analyze a lot of data that may change from time to time. Lawyers handle this situation by arguing that the issue at hand has to be resolved based on merit. One way to resolve this issue is to use words that can be interpreted in ways that can be applied in different situations. The lawyer's ability to use language in this way, to handle complexity in situations is one of the cornerstones of the lawyer's craft. A lawyer uses his skills to create definitions and structure in a law to adapt to the situation that the law should cover. The lawyer has knowledge in how laws are constructed and how they are used to regulate behavior. A lawyer could not use the same type of law elements in every situation. For example, it would be impractical to use jail as a remedy in all situations. The lawyer may need to create a new kind of remedy if it is not already available to him. This gives the lawyer the freedom to innovate within his field of profession. To return to the markup

language, HTML is using jail as a remedy for everything. This means that you have to innovate around this problem. A more efficient solution would be to be able to create other remedies or tags, as elements are called in the markup language. XML allows you to do that, which means that XML is a more flexible tool in an environment that needs flexibility.

XML is a meta-language for creating new markup languages. That means that with XML you may construct your own rules for how data of a certain kind is going to be processed by an application. This is very useful for example if you use a lot of data that cannot be translated to another database's categories.

XML can be used by any application that can read XML in the same way that a HTML document can be read by an application that can read HTML-"web-pages". That means that XML is more flexible than HTML, and can be used for more specialized things than HTML, which is not more than a way of displaying computer content to a human.

XML is more suited for machine-machine communication since it doesn't need a GUI in order to work. To put it simply, a GUI is a graphical representation of what goes on inside a computer. In early PC's there were only a command-line, which is a screen with text where you typed in commands. When using a command-line, there is no need to use the mouse or folders on the screen, which is common in today's computer environment. The GUI makes it easier for humans to use a computer. Two computers that communicate do not need a GUI, however. Since XML is more suited for computer-computer interaction there is more efficiency of communication, since there is no human interference in the form of interpretation. If you use this mechanism to make agreements between two computers, something that is desirable between businesses for example, a number of problems arise. The main problem is that there may not a clear understanding on who is making the decisions when the computers are making the agreements. This leads into discussion of electronic agents and the relation of these questions to traditional middleman law. The notion of middleman agent questions are not at the core of this essay, but can be an interesting way to further research issues in the area of XML Web Services. It all depends on how the XML Web Services market will develop. If there will be produced XML Web Services that are complex enough to be autonomous agents, which is a clear possibility, then these questions will be of great interest.[26] Until then this essay will stick to what could be a foreseeable development in this area.

[26] Geurts, Rik (2003) p. 232

Since XML allows for the exchange of both content delivery as well as structure delivery, there are even bigger concerns. If an agent is allowing another agent to change its structure, that could mean that an entire databases structure could change, in essence the entire database could become useless.

The area of XML is starting to attract attention from legal scholars. The main focus has been XML:s significance in relation to personal data regarding issues of privacy and regulation of content within the framework of XML-based documents.[27] The services dimension of Internet based business has also been recognized to some extent.[28] This essay takes a different stance, in that it accepts the role of XML as the base for XML Web Services and instead focuses on market building on open standards.

7.3 What is an XML Web Service?

It is difficult to give a short description of what the XML Web Services market is. The reason is that XML Web Services takes on many traits of the Internet in that it is so versatile and can be utilized in so many ways. For starters, XML Web Services are a sub-set of Webservices. Webservices is a collective term for all kinds of services that are available on the Internet. Even though there are difficulties in getting a clear definition of what XML Web Services are, they are clearly distinct from the notion of an Internet banking service for example. The Internet banking service might use XML Web Services as a part of its webservice, however.

I will now provide a couple of definitions of XML Web Services. An IBM definition of Web Services is:

"A Web service is an interface that describes a collection of operations that are networkaccessible through standardized XML messaging. A Web service is described using a standard, formal XML notion, called its service description. It covers all the details necessary to interact with the service, including message formats (that detail the operations), transport protocols and location. The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written. This allows and encourages Web Services-based applications to be loosely coupled, component-oriented, cross-technology

[27] Magnusson Sjöberg, Cecilia, 2002:112, pp.195-206; Lundblad, Nicklas and Magnusson Sjöberg, Cecilia, (2003), pp. 2
[28] SOU 2003:35 Digitala tjänster – hur då? En IT-politik för resultat och nytta., SOU 2002:51 Breddtjänster – ett nytt skede i IT-politiken.

implementations. Web Services fulfill a specific task or a set of tasks. They can be used alone or with other Web Services to carry out a complex aggregation or a business transaction.".[29]

Microsoft has several descriptions for web services:

"A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web Service is implemented. Web Services combine the best aspects of component-based development and the Web, and are a cornerstone of the Microsoft .NET programming model."[30]; "A Web Service is programmable application logic accessible using standard Internet protocols. Web services combine the best aspects of component-based development and the Web. Like components, Web services represent black-box functionality that can be reused without worrying about how the service is implemented. Unlike current component technologies, Web services are not accessed via object-model-specific protocols, such as the distributed Component Object Model (DCOM), Remote Method Invocation (RMI), or Internet Inter-ORB Protocol (IIOP). Instead, Web services are accessed via ubiquitous Web protocols and data formats, such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). Furthermore, a Web Service interface is defined strictly in terms of the messages the Web Service accepts and generates. Consumers of the Web Service can be implemented on any platform in any programming language, as long as they can create and consume the messages defined for the Web Service interface.".[31]

From a technical point of view a definition of web services can be:

"a platform and implementation independent software component that can be: described using a service description language, published to a registry of services, discovered through a standard mechanism (at runtime or design time), invoked through a declared API, usually over a network, composed with other services.".[32]

Ethan Cerami's minimum definition of a Web Service:

"… any piece of software that makes itself available over the Internet and uses a standardized XML messaging system.".[33]

Phil Wainewright suggests that the unifying theme is:

"…a general move towards adoption of dynamic coupling of software-based service components at all levels of the online infrastructure.".[34]

---

[29] http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf
[30] http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000442
[31] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnWebsrv/html/Websvcs_platform.asp
[32] Graham, Steve et. al (2001), p. 9.
[33] http://www.oreillynet.com/pub/a/webservices/2002/02/12/webservicefaqs.html
[34] Wainewright, Phil (2002), p. 8

I have shown a number of definitions of XML Web Service to illustrate the difficulty of making a relevant definition. Even though the problem of defining an XML Web Service, I think the best definition of XML Web Services in order to understand what XML Web services means is: a platform- and implementation-independent software component that can be *described* using a service description language; *published* to a registry of services; *discovered* through a standard mechanism (at runtime) or design time); *invoked* through a declared API, usually over a network; and *composed* with other services.[35]

The definitions are important, but it is also important to see that the actors are making different definitions of the same technological phenomena. You can also make the observation that the market-leading actors like IBM and Microsoft are submitting their own definitions. They have a vast influence over the language, which could include some actors and exclude others. The definitions in language will eventually be translated into computer code, which means that the language definitions can be very important. Especially this can be true for smaller actors, that obviously will have difficulties in influencing the definitions as made by the larger actors. However, by using an open standards organization as the base for the specifications of the underlying standards, there is also influence for smaller actors.

7.4 Applications

The virtual layer, discussed earlier, is the one layer that supports web-based applications like XML Web Services. The Internet has previously been used mainly for sending content, like bits of information that were digital products like software, music and text. With XML Web Services it will be natural for corporations to use the Internet for business processes and web-based applications.

Applications are different from content in that an application is performing a process. Business processes like the handling of customer information, for example, can be automated for automatisation and cost-saving purposes. The process is put into a computer application, which is conducting the process. The process is dependent on a number of attributes in order for the process to work.

[35] Graham, Steve et. al (2001), p. 10

Development of an application demands a lot of time and resources. The reason is that the application more often than not has to be specifically tailored for the specific company whose processes needs to be automated.

One of the advantages of using XML Web Services to execute business processes is that XML Web Services allows combination of automated process components. Web Services allows that services are developed without the need for construction of a new application from scratch. The functionality is embedded in the technical infrastructure that XML Web services are a part of. It is fair to say that you don't need to construct an application, but instead focus on building the business process. The processes can then be added together without focusing on application building. From a construction perspective you are working with services instead of building software. The construction of applications will therefore be of a lesser importance in the XML Web Services-environment for the end-user.

The transformation from content to process will mean that the network and the business will have more open boundaries. You can therefore talk about a common infrastructure for XML Web Services.[36] This common infrastructure will have several issues, some of them that I will discuss below.

The process that is run through the use of software is moving from the desktop to the Internet. Today when you run a software application on your computer, the applications connects to the computer's code libraries via gateways called API:s. These API:s are the connections to the computer's "brain" that will execute the functions that the application's processes are calling. The functions and code libraries that previously has been called upon on the local computer, is with XML Web Services provided through an internet-connection. In XML Web Services, functions can be provided from hundreds of different function providers at the same time. On your local computer, you can only call on functions that are provided by your operating system or by other installed programs. When you use API:s over the Internet new problems will emerge since the net has built-in problems that has not been present on the local machine. Performance problems are the main concern here, but also security issues.

XML Web Services will certainly demand strong security and robust networks. Compared to a web-browser: if a web-browser is going to display a newspaper-page the worst thing that

[36] Wainewright, Phil (2002), pp. 5-6

can happen is that some text or graphics are missing. If a XML Web Services component fails the worst that can happen is that a company's business-critical processes seizes to function altogether. It is hard to imagine that you would not want to control the central development of the components make up that XML Web Services.

In order to convince the parties involved that a network at this level is going to work, standards are needed. Standards that are not developed, means that the parties will settle with solutions that are not adjusted to the full potential of XML Web Services. Therefore, at present, most companies have focused on vertical integration with XML Web Services within the company's firewall. With time, standards will be finalized for Web Services full potential, which will mean that applications can be integrated over the Internet even outside the firewall.[37]

The areas of use for Web services can be found within three main groups, plug-in functionality, remote infrastructure and EAI (enterprise applications integration). Plug-in-functionality are functions like news feeds, search boxes and other smaller functions that can be added to portals and web pages. A remote infrastructure service is a third party that offers Web Services technology that enables services like authentication and payment services for commercial web sites. EAI means integration of applications within a company/group of companies or a value-chain.[38]

7.5 Functionality

Building an application you need functions that are needed by the user of the applications. An "off-the-shelf"-application is mainly offered, which means that it is being licensed in a standardized version. There are also types of software that can be tailored to some extent to end-users needs. A newer type of software that can be growing with the advent of XML Web Services is a combination of the two earlier types. The software is assembled is small modules that are tailored to the customers needs.[39]

In off-the shelf-software, the licensee does not need many of the functions in the package but they are nevertheless included. XML Web Services enables the buyer to only pay for the components that he really needs. Today this is not possible for a number of reasons. License

[37] A.a. pp. 7-8
[38] A.a. pp. 8-9
[39] Warsta, Juhani (2001), p. 34

agreements will not allow a user to alter the source code. Even if the user had access to the source code, it wouldn't be possible to alter the code without great investments in time and money. The application also had to be tested to ensure security of the application in order for the application to be installed on all needed computers in the company. In essence this means setting up your own programming department, which is not practical for most companies.

An application in the XML Web Services environment can be made like this: I will visit a web page that offers application functions. I choose the functions I need to rent and for how long. The application is assembled instantly and you can use the application at your company instantly.

From the end-user's view the difference compared to ordinary construction of applications is evident. No development time is necessary, even is you need specific functions. You don't need to hire programmers or employ a computer consultant. There is no software to download and install. There are no lump-sum payments, since you pay for the time you use the application. You only pay for the functions that you are using.

An XML Web Services application is used over the Internet, which does not mean that it is has to be used through your web-browser. There will be "shell"-applications that is adjusted for use of the XML-applications.[40] The shell-applications will be like the web-browser is today: a shell for the content they will be filled with. In the world of web services the shell will be filled with applications instead of images, text and sound.

XML Web Services applications can also be executed locally on a hard drive without Internet access. XML Web Services can therefore be run within the firewall in a company or a group of companies. This usage is called integration within the firewall.

The usage of small components that can be combined opens up a large market for small developers. Especially the possibilities that enable the smaller developer to create revenue streams from time-limited use of applications will be an important market for smaller developers.

[40] One example of a simple XML Web Services shell-application is Watson, by Karelia software. http://www.karelia.com

7.6 EDI

There has been similar functionality like XML Web Services earlier, in the form of EDI.[41] EDI couples two companies together enabling the sending of large quantities of standardized information between them. Mostly the parties are handling orders and stock information. EDI – agreements were and still are difficult to negotiate and to implement. However, the benefits of EDI-agreements for the most part outweighed the drawbacks. EDI is still an important solution for many companies.

During the 1990s a solution came in the EDI-environment that is similar to Web services. That development was called Open-EDI. Open-EDI was envisioned to make it possible for EDI to be used between multiple companies.[42]

To use EDI for a larger audience wasn't possible, since large companies were the targets for this kind of data communication. There were also contracts negotiated – EDI contracts – that were complicated paper-documents that took a long time to settle.

EDI is communication between two participants. XML Web Services opens for a flexible connection between multiple actors.

7.7 SOA

The goal of XML Web Services is to enable a SOA - a service-oriented architecture. The SOA is an abstract pattern that applies to a wide variety of Web Services situations. SOA defines an architecture consisting of three roles (service provider, service registry, and service requestor) that can be fulfilled or implemented by a variety of techniques. SOA also defines the contracts between these roles in terms of three operations: publish, find and bind.[43] The SOA environment is the key to XML Web services. It enables three-part relationships to be created inside the XML Web Services framework.

I will describe one practical situation where SOA is not present and the problems it produces and one situation where SOA is put in effect where the problems in the first situation are solved.

---

[41] For a basic work-through of some of the issues of EDI, please refer to Tornéus, M, (1999)
[42] Mitrakas, A. (1997)
[43] Graham, Steve et. al (2001), p. 21.

A company – company "A" has around 50 employees and is managing its payrolls internally. The payroll system is today run on software that is run from an internal server at the company. The company has a site license for the software and it was paid for at a one-time payment. They pay regularly for service and maintenance of the software via a service license from the company that distributes the software.

The company that manages the software – company B – is thinking of ways to increase its productivity and lessen its costs. The costs come mainly from marketing and distribution of the software. There are also big costs in testing and adapting the software to new environments. Company A is also looking to minimize their costs and get a more flexible operation. The problem for company A is that they cannot change software as a change in software will provide company A with even bigger problems and bigger costs. There will be difficulty in finding a new partner, to negotiate a deal with the new partner. The problems are thus due to transactional costs. These costs are going to limit company A to find a new partner to work with.

A solution for company A is to utilize the SOA in Web Services. Company B or some other company provides a Web Service. The webservice is *published* in a UDDI registry. Company A sets the preferences for the service that they want to use at the registry. The SOA enables the possibility to *search* for a service that matches the preferences set by company A. After a finished search the registry provides the best-suited services according to company A's preferences. Company A suddenly has a multitude of different Web Services to choose from. That is different from before, when they only had one service from one company to choose from. Company A can now choose a new partner but can also change partners easily again.

The SOA also takes care of the negotiation process, by comparing XML contracts between the provided services and the preferences set by Company A. Every Company that provides a Web Service also provides a XML contract, which is a computerized contract. The XML contract can be used to negotiate with other contracts.[44] When a successful negotiation has been made the SOA *binds* the companies together by providing Company A with the service and telling Company B who has connected to the service. In this example, this is done

[44] It is important to understand that contracts and negotiations in this context are not the same as the legal definitions of the words. This mixing of meaning can create specific problems, which I have hinted in the question of obligation between the parties that uses this kind of service.

without any human intervention when the search process has been started. You can also set the preferences to automatically search for a new partner if certain parameters have been met.

Company A uses an environment that enables it to make a flexible search for other companies. The search is done behind a veil, so that the best service can be able to match the contract. This means that the importance of brands might be diminished since the SOA are looking for the best service. There might on the other hand be a description of some kind of brand index function that is needed. But the SOA does not need any human attributes to find the best service – it gives you the best service according to your linguistic and quantitative preferences.

Developers of applications list their services in UDDI, which is a public catalog for XML Web Services. UDDI, which is short for Universal Description, Discovery and Integration is a standards-based approach to very sophisticated service registry implementation.[45] UDDI provides catalog of services that can be accessed through XML Web Services. UDDI is meant to contain all available services on the XML Web Services market. UDDI will provide the backbone for UDDI operators.

UDDI Operators are companies that run public instances of a UDDI Business Registry. Operators have signed an operator agreement that commits them among other things to support the UDDI API and to replicate registrations among themselves on a periodic basis.[46] An UDDI operator provides the means for a company to connect to the XML Web Services SOA. A company can connect to an XML Web Service without the need for a UDDI Operator, but then he can only connect to a XML Web Service that he has fund, negotiated and agreed with on a personal level, not through the XML Web Services SOA. There is a lack of automation in this later process and the difference is not great compared to a simple web-portal solution, where a person searches for information and acts on that information. The problem with this later process is that the actor is acting on more incomplete information than he would have done in the XML Web services SOA environment.

When you want to order an application you submit the search-criteria for the services you want. The preferences are sent to UDDI, where the preferences are matched to the applications and services in the database. When UDDI has found a service that is closest to

[45] Graham, Steve et. al (2001), p. 22
[46] A.a. p. 23

the service that you have specified, the service is selected and is delivered. UDDI is acting as a depository for a broker.

The advantages with the system of XML Web Services are obvious. Since no people have to be involved in the search, the search and the best options are swiftly discovered. If the requestor is not satisfied with the service, or finds a better partner, it is easy to change service partner, by adding another preference. You can therefore quickly escape from situations where the service provider is not up to the current specified preferences. This function in the XML Web Services system is an advantage from the requestor of services, since it makes it easy to find a better service provider. Previously it was hard to get out of a bad relationship since the data formats were provided in a manner that couldn't use other suppliers' applications.

Rosettanet is an example of a standards organization that has gone beyond the core standards and has developed a number of business process standards that can be used by XML Web Services. The model in Fig. 1 shows how an agreement can be made through electronic messaging.

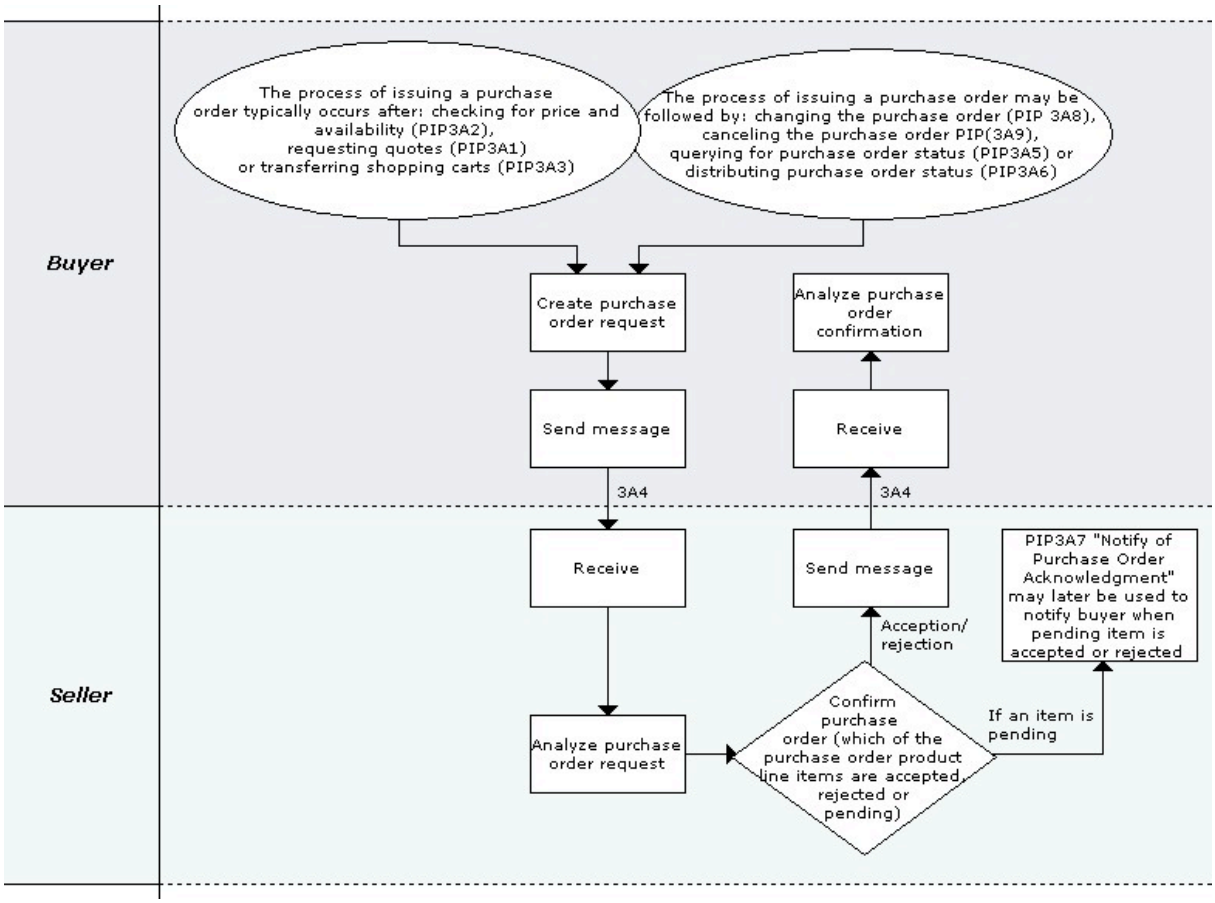

Fig. 1. Standard business model contract procedure. This example regards PIP 3A4: Request Purchase Order. Source: http://www.rosettanet.org/PIP3A4

These models are similar to the interchange standards that were provided in EDI. Interchange agreements has been shown to reveal that legal obligations can be hidden. These obligations could in EDI, if they were not acknowledged, affect the original contracts between the parties.[47] In EDI there were two parties that often hade lengthy pre-contractual discussions that emanated into an EDI agreement. In spite of these agreements in a fairly controlled environment that makes up EDI, there could be legal difficulties. This suggests caution when discussing an environment that has no such period, but is created instantly. Issues like cancellation and the reasons for canceling when making incorrect order are only a few examples. Many paragraphs in the act of Contract will be enabled.

7.8 Open standards

What you are doing in development is that you take an open standard and build your applications in compliance to the demands and specifications within those open standards.

The difference between using open standards and other standards can be important. There are three main types of standards: proprietary standards, de facto standards and open standards. The type of standard that you choose to build your technology around depends on the purpose of your business. A strategy that would try to keep competitors out of the market could try to build the market on a proprietary standard. Proprietary standards emerge when a company invents a technology, but want to keep the control over how the technology is developed. A de facto standard is a standard that can be both proprietary and open, but that most of the companies on the relevant market is actually using. An open standard is developed on the basis that no single actor should own the rights to the standard, but the intellectual property rights that are a part of the standard should be open for all. The advantage of an open standard is that all actors can use it in order to build new technologies on top of it.

In standards development four types of standard building groups can be categorized. Scientific and professional societies, which beside their duties as developers of professional and educational goals, also produces standards. There are trade associations, who promote their industry and solve problems specifically to that industry. They also produce standards for manufactured products or for products used by their industry. Standards developing

[47] Korah, V. 1990, pp 128, 129

organizations often called SDOs, are bodies that are designed specifically to develop standards for specific purposes. Developers of informal standards distinguish themselves from the other three groups, because they operate outside traditional frameworks for standards development. The standards they produce are ad hoc, de facto or consortia standards.[48]

Private sector initiatives or informal standards development was previously easy to identify. Through openness and the adoption of due process it was also easy to communicate with these bodies. Today industry demand for expediency has made it more difficult to identify these standards bodies. Standards are not goals in themselves, but means to a goal, namely to be an intermediary and medium for communication. The pace of the markets in telecommunication sectors for example has made the formal standards bodies more or less redundant in certain fields of technology, due to the lengthy period of developing traditional standards. This has created a need for consortia that can move faster by de-emphasizing formal due process procedures.[49]

When a company's products become the "standard" within the industry, a proprietary informal standard has been created. If a company strategically positions its products to increase market share and collect royalty fees from licensing IPR, then this is a proprietary de facto standard. If the market chooses one company's products instead of other companies' equally effective products, as an end-result a de facto standard is created. In the arena of consortia standards development, groups of supporters to the technology are working together to form a standard. In that case that producers tries to achieve critical mass in a technological field in order to establish a standard. These types of standards constitute 95 % of the standards today. There are also ad hoc variety-reduction standards that seek to reduce the number of standards in a given technological field.[50]

In the U.S. standards developers are viewed as service providers, and corporate managers allocate resources to the SDO that provide the best return on investment.[51]

A set of reasons for why standards have been increasingly important is that: 1. the stakes has risen extremely due to the ICT promise of delivering an infrastructure to the global

[48] Spivak, Steven M. & Brenner, F. Cecil (2001), p. 135
[49] A.a. p. 135-136
[50] A.a. p. 136-137
[51] A.a. p. 145

information economy, 2. conflict of interest has become greater, while companies and regions are dependent for a specific standard for their survival.[52]

The benefit of an open technology product system is that a single provider can create a product or service without being forced to create a complete integrated system. Users can correspondingly utilize the open system by custom-design his system by choosing components from different manufacturers, resulting in an optimized system at minimum cost.[53]

Creating and controlling de facto standards can be strategically managed by licensing IPR. The benefit of standard control through active licensing is the reward of receiving payment for the use of the standard.[54]

Standards are targeting technical components issues rather than actors, and standards are not generally seen as rules for a system of actors.[55] This is though due to a very strict interpretation of what a standard is. Taken into consideration that a standard is actually and evidently shaping the actors behavior, I would say that a standard is really rules for actors, even though they are not entirely that. When a standard is providing information to actors within a system the standard is beneficial since it provides information to all actors alike in theory and it does provide possibilities for greater competition between the actors on the market. But the agents have different asset ownership entering such a market. Therefore some actors may have a greater advantage in design and production components in a given market. These strong actors may use their advantage to affect the standardization process. A group of actors may then gain a disproportional possibility to implement the standards internally and thus have a bigger chance of dominating the market using the standard as a strategic tool. This is common in sponsored standardization, where a certain standard is picked for strategic reasons.[56] Risks involved in such attempts to dominate a market is of course antitrust legislation. Strategic pricing can be used in order to establish a standard. The dominant players have an incentive to manipulate interfaces between components in the system they control.[57] Anticipatory standards development is contrary to de facto standards a less risk-taking way of creating standards. Anticipatory standards are created before products are

[52] Lundvall, Bengt-Åke (1995), p. 8
[53] Tassey, Gregory (1995), p. 162
[54] Arora, Ashish, Fosfuri, Andrea & Gambardella, Alfonso (2001), p. 177
[55] Lindmark, Sven (2002) p. 87
[56] A.a. p. 83
[57] A.a. p.84

made, and are based on a collective, consensus-based effort to research engineer and design products. A reference model is made to create the context of the standard, later the standard is created, while identifying its service and putting the code in a relevant protocol. Strategic choices for firms include assessing the costs of participation (e.g. personnel cost, resource commitment) against possibilities to influence the outcome, learning what information to withhold or provide (including property rights), agreement to proposals, stalling and compromising.[58] In fast-developing techno-economic systems the rate of technological progress is rapid. A consequence is that state of the art technology has to be included in standards and these have increasingly come in conflict with property rights. The rapid pace of technology development is also pushing forward consortia for developing standards. A consortium has less strict requirements for openness and consensus than regular standards bodies.[59]

7.8.1 The standards actors

When working on developing standards there are several important aspects to consider. In order to have a working standards development, you need to consider how the standards that are set in place will affect values and how they will affect legal institutions of different kinds. To set a standard you may choose to make it a standard. OASIS and the W3C are the contributors of the XML standard. When XML was founded it was not developed until there was interest in it.

OASIS is the main developer of standards for XML Web Services. OASIS is also developing standards in neighboring technologies and areas. The work is organized in technical committees where the members of OASIS are working together. OASIS is a fully open standards body where all members are allowed to contribute in the organizations work. Membership fees are ranging from 250 USD/year up to sponsorship status of 13,500 USD / year. A mission statement from OASIS: “The mission of OASIS is to create interoperable industry standards based on public standards such as XML and SGML, as well as others that are related to structured information processing. While this could be read to say that OASIS could be anything having to do with XML, in reality OASIS is most interested in creating works related to interoperability, whether vertical industry applications of XML, conformance specifications, horizontal e-business platforms, web services, etc.”.[60] On the

[58] A.a. p. 84
[59] A.a. p. 85
[60] http://www.oasis-open.org/committees/guidelines.php

OASIS board of directors are representatives from BEA Systems, Intel, OASIS, Hewlett Packard, Microsoft, Sun Microsystems, The Federal Reserve System and IBM.

OASIS has a section on IPR, where a contributor of IPR does not have to claim more than a reasonable and personal knowledge of any IPR that he contributes. In section OASIS.IPR.3.1.6: ”The contributor does not represent that he personally knows of all potentially pertinent proprietary and intellectual property rights owned or claimed by the organization he represents (if any) or third parties.”.[61]

No less than three members of OASIS form a technical committee. There is a requirement for an electronic mailing list to be held public. The minutes of all TC meeting are also published to the mailing list.

The discussions in the technical committees create an arena where the technical specifications are being molded. It is not dissimilar to contacts between political minor officials in the government bodies, where the practical policy is being created (albeit from political decisions decided by senior levels). Solutions and specifications are discussed. The committees are discussing from already defined definitions.

The standard is neutral at the same time that it is connected to market leading actors. Still the standard is open for all actors to utilize. If it weren’t it would be hard for the standards body to get accepted by most market actors. Still it seem to be a problem that there seem to be no discussions on the societal aspects or ethical and legal consequences of the standards process in OASIS. To refer to societal phenomena and future scenarios is on the other hand a time-consuming and energy-demanding task. In an environment where swift results are demanded activities like that might be seen as non-relevant.

It is difficult to see a standards development without market leading actors. A standard will have difficulties with interoperability if the influential actors are missing. When a standard is completed and is due for implementation, a business can then be built upon the standard. That is hard to do if there isn’t any market acceptance for the solution. It is easier to implement a standard if there are a will and capital to build on it. An open standard could be seen as being built for that purpose – to limit the risks at the building of the market.

[61] http://www.oasis-open.org/who/intellectualproperty.php

The aspect of a market that takes the societal aspect seriously might be a thing of the future. The more connected the market will be with societal interests and issues of content, the more companies will be forced to take real responsibility for these questions. To be unaware of legislation or value conflicts could give diminishing profits due to the buyers on the market may find the product unacceptable.

7.8.2 UDDI

UDDI, which was discussed earlier, is a registry for XML Web Services. In this registry a company can submit its XML Web Service and have the possibility to access the SOA. Today the registration to UDDI is made through sites hosted by IBM, Microsoft, SAP and NTT-Com. There is a choice between these sites. In order to register, there is a requirement to register at one of these operators' private databases.

Example of the terms of use for the UDDI service is displayed on the SAP site. The service is explained to be a registry to give companies a possibility to describe their services and offer them to customers. The information that is left to UDDI is handled by "operators" that have pledged to share the information between each other and the users of the service.

There are descriptions on how to fill out the information and what should be requirements for the authorization to be able to fill out the information.

It is described how the information is allowed to be used. Information that is entered into fields that are marked "private" is available only in the services registry that received the information. Information that is marked "public" will be available to other operators and also available in the public registry.

An actor can be active in multiple roles. An advantage with the standards committees is that an actor can be a small developer and still have influence in development of standards. It is however hard to imagine that a small developer in a single-employee business will be able to assert itself in the competition with a team of full-time employees from a transnational company, even is the possibility is there.

A large corporation has possibilities to coordinate its activities. This is an advantage if a company wants to play multiple roles. That means that the company's influence in a b-web will have larger chance to influence if it can coordinate its resources in several roles.

8. Law

In this section law is considered to be law as it is traditionally viewed – as a state-based legislation produced from within the legal system.

8.1 Contract

The starting point for agreement in Sweden is the law of Contract (1915:218). § 1 of the law of Contract establishes that there are no formal rules for offers and accepts. A contract can be made orally or in writing. It can therefore be created by electronic means, for example via XML Web Services. The Law of Contract is however not mandatory, which means that the parties can use another way of establishing a contract between each other.

For example, we presume that nothing has been said between the parties of excluding the Law of Contract. How do we establish a contract? There has to be an offer and an accept. The acceptance should be given within a reasonable time. A mere technical confirmation on that an offer has been received is not considered to be an acceptance. When a human is involved it is complicated enough to establish that an acceptance has been reached, for instance, by pressing an ”OK”-button. In that case there has to be high demands on a clear GUI so that there can be no ambiguity on that the user understand that he has produced an action with legal effect.[62]

There is also the possibility of being forced into an agreement. The SOA may present problems in this area. When using the SOA, you may be presented with a service deliverer that you do not want to use. If it will be that a process will decide for you which of deliverers that will provide your process, then will you be satisfied with the choice? Even if you get the terms you want, the provider may be a less desirable deliverer. If you are being forced into such an agreement, could this be seen as the requestor being forced into the agreement? There are rules in the third chapter of the Law of Contract that can invalidate a contract on the basis of coercion and betrayal. According to § 36 of the Swedish law of contract there is the possibility of making a contract invalid if certain conditions, like if the terms are unreasonable based on the facts at the creation of the agreement, later occurred circumstances and surrounding circumstances.

[62] Lindberg, Agne & Daniel Westman (2001) pp. 70-71

The creator of a system for electronic trade needs to focus on several issues in order for his system to be valid to a court. The key is for each item to be proved in the court. In Sweden there is a free test of evidence, which means that the evidenced material can be judged on its own merits. The court has the last word on whether the evidence is accepted or rejected. Other legal systems have other types of tests of evidence, which can be a problem when using a system that is constructed outside Sweden. The parties need to be aware of these questions in order to make sure that the material that has emanated from the transactions are accepted as evidence.

By using contracts, the actors that are tied to the construction of the market can create trust between each other and in the end for the market itself. The contracts are constructed by the implementation of different processes that are legal, technical and economic in nature. The analysis of several areas is necessary in order to create an understanding for what it takes for market construction.

The Internet enabled an increased globalization in that even small actors can participate in a worldwide market. A contract is generally created using the previously mentioned principles and clauses in law that is similar throughout the world due to international conventions.

Most commonly, contracts that are used in trade over the Internet are complete contracts. When only the price is bargained over for standard products, then this constitutes a small problem. It is when an end-user wants to negotiate more instances of an agreement over the Internet, the drawbacks of Internet trade becomes visible. The marginal cost of being able to re-negotiate certain clauses in a standard agreement would be quite large, since it would be possible to maintain a flexibility in contracting, while preserving the benefits of low-cost internet transactions.

One of the reasons for the difficulty in using flexible contracts is that the architecture of the Internet has not allowed this kind of use. With the advent of XML and XML Web Services it will be possible to complete these kinds of transactions in a fully automated manner. Since XML is based on semantic rules it is possible to change terms that are not only considering quantity and price but also content related items that are not quantifiable.

Even though an agreement can be negotiated and agreed upon, there can still be uncertainties on whether the parties are really obliged to the contents of the agreement. When contracting

through XML Web Services, the parties' two negotiators can be considered to be two "agents" that are negotiating on behalf of the parties. By using WSDL, the two agents are connecting the services. The problem with WSDL contracts and other kinds of electronic agreements like TPA are that they are not legal documents in themselves. Their construction is based on pure architectural criteria, like if the technical requirements are in place, the services can be connected. Does it really mean that the parties can be sure that both parties are obliged to follow this kind of agreement?

In Sweden, evidence is based on its own merits, which means that this kind of technical arrangement at least could be taken on as evidence to prove that an agreement has being been set up.

Today, you could question the validity of such agreements that are not even refers to a separate legal agreement. You could say that the parties' intentions prior to the conclusion of the agreement can be determining the obligations of the parties. If an agreement has been concluded, could one of the parties claim that the circumstances has suddenly been changed and wants to withdraw from the agreement, can the other party claim that the contract should still be concluded? One way to do that is to refer to general terms or law that is applicable for electronic trade.

When electronic agents are being used to create contracts, a "hypothetical intention" has been attributed to a party that uses a system that enters into agreements on the behalf of that party. The party is then deemed to agree to the contracts being made once the system has been put into use.[63] Another view is that obligation is created for the parties as an effect of certain external circumstances that in combination would work as directly contractual. Such a situation would be where practical needs forces effect of agreement, in spite of the fact that a declaration of intention in a traditional sense is not at hand.[64] This is in line with article 13 item 2 b of the model agreement of UNCITRAL where the risk for declarations of intentions that the system generates is put on the party that chooses to use such a system.[65]

For the actors, a better solution might be to be able to re-negotiate the contract instead of terminating the contract or to create doubts on that the contracts validity.

---

[63] A.a. p. 73
[64] SOU 1996:40 s 121 f.
[65] UNCITRAL Model Law on Electronic Commerce

When two computers are communicating without human intervention, there is an area of law that is called agent law. This is usable under circumstances where a third party is conducting tasks for a principal. It is unclear whether a computer that acts on behalf of principal is an independent actor or if the principal should be seen as identical with the computer. Computerized systems are being increasingly independent, which makes it increasingly difficult to draw the line. The literature suggests that it would be inappropriate to view software program to be an agent.[66] Possibly, with XML Web Services we might see computer software that are being more and more independent, where the results of the negotiations being carried out by the agent is not predictable beforehand. The area for determining whether an agreement could be deemed unreasonable of unusual and surprising in such an area could be relatively big.[67]

Another question that may arise is what will happen if a XML Web service will destroy some of your data? Consider if company A is using an XML Web Service that is provided by company B via the UDDI registry and are bound together. While using the service, A's database that is processed with the help of B's service is altered. A want B to pay for the damage. The first thing to decide is whether there has been an agreement made. A is using the service, and could therefore be seen as having entered an agreement. What if there is no standard agreement and the service is just provided as is? There has to be a legal specification for if there is an agreement created. By relying on a business process, it is not certain that the process in itself produces the items that are needed in order to prove that an agreement has been made.

Can your network provider be liable if the connection fails? Can for example UDDI be liable if UDDI provides an XML Web Service that does not match the criteria of such a service? According to the agreements that UDDI provides UDDI is merely a registry. The middleman between the service provider and the end-user will be a context provider. Will the context provider be responsible? That depends on the deal that is provided by the context-provider. In the case that the context provider excludes himself from responsibility, there is a problem. If the context-provider merely provides services and takes no responsibility for them, which might be a reasonable position for the context provider, the parties are left to each other to work out legal problems. How do company A know that company B has read the terms that

[66] Lindberg, Agne & Daniel Westman (2001) p. 69
[67] Hultmark, Christina (1998) p. 73

have been constructed? These questions are important to address in a system that shall work as an independent system.

One possibility to create a legal certainty in this area would be to refer to agreements that are not present in the technical system. The problem with that approach is that the possibility of flexible contracts may not be able to integrate with such a solution.

8.2 Licenses and services

When it comes to computer components, software license agreements are necessary between the application developers before offering the service through a context provider. The context provider can provide services with license agreements with a different content. The service agreement provided by the context provider demands more terms on reliability of the service etc than restriction on copying of software, since the software is not distributed to the end-user. Reliability of service becomes very important since the software is executed over a network that can be disturbed. The software components that are licensed could be parts of a bigger product that is made by several services by the context provider. The end-user will license the services at the same time as he might be licensing text, sound, or images as a part of the service provided.

The offering of services is presumed to be run by XML Web Services brokers. By creating three-part relationships between the application developers and the context provider a bigger product can be created.

An application developer can of course offer his application directly to a customer. There is an interesting difference from previously though, because when the application developer offers his service to an end-user, the developer is a context provider providing a service. The license terms are thus different, as discussed above. This is a new situation for the applications developer who previously has been distributing software in order for the end-user to install the software locally. The delivery of software in a service form might not be a relevant business for the software developer. In order for the application developer to avoid the new situation, he would probably be considering to offer his software through a third-party context offer instead. The context provider will then handle the situation with the end-user.

The licensing of immaterial rights can have both pros and cons. Fig. 2 illustrates some of these pros and cons for the licensor and the licensee.

For the licensor **For**	**Against**
Assets	
Local volume is not worth own investment	**Profit goes mostly to licensee**
The product cannot be protected against copying	**Lack of control of quality, brand, name, sales effort**
Competence	
The activity lies outside core competence	
Perspective for learning from licensee	
Positional advantage	
Entry into market is blocked or entails too many risks or investments	**Lack of control; danger of creating a competitor**
For the licensee **For**	**Against**
No R&D expense and risk	**Arrow's paradox of information for assessing value**
Profit from knowledge, expertise, contacts of licensor	**Contractual limits on operations**
	Risk of obsolescence
	Ongoing royalty payments

Fig. 2. Pros and cons regarding liceningfor licensor and licensee. Source: Nooteboom (1999), p.61

To license immaterial rights through XML Web Services will mean new challenges for lawyers and engineers. The technology is similar to the one technology that created big challenges for the intellectual property rights dimension – the Internet. XML Web Services is a faster, more advanced technology that will mean new challenges.

Licensing can in its simplest form be seen as a right to use a patent or a copyright for exchange of a fee. The licensing most often mean a factual transfer of technology for inclusion into another product for example.

To license technology was previously a business that was regarded as a side-business. Lately, licensing has become an increasingly important factor to reach new markets and to achieve co-operation between organizations. Important binding methods in a licensing process are for example, legal, technical, knowledge-based, organisatorical and financial means.[68] Licensing is mainly used in order to increase competitive advantages in products, but the side effects of licensing such as sales or to secure a market position can be other possible goals in a licensing situation.[69]

Licensing is capable of increase competition within an area of business through the fact that a technology can be offered and distributed by multiple actors. There is a risk that the licensing leads to diminished competition when, as cross licensing between competitors in a technology that is state-of-the-art my lead to limited incentives to develop products.[70]

License agreements that are satisfying both parties in an agreement is based on: care when other party is chosen, realistic expectations on products or methods that are covered by the license, to hold on to items that are reasonable and to clearly display argument s in negotiation. The agreement also has o be practically useful and that it doesn't lead to conflicts between the parties.[71] The area of licensing lacks custom or optional rules. Moreover, the area of licensing demands thorough knowledge of highly specialized areas of law like patent law, copyright law, tax- and currency legislation, means that most companies needs highly trained specialists when dealing with licensing issues.[72]

A sole license gives the right for the licensee to use the license object at the same time that the licensor has the right to license the license object to other licensees. An exclusive license means an exclusive right for the licensee to use the license object.

It is important that the object of licensing – the license object – has to be clearly defined in the license agreement. Sometimes this can be very hard to do, since immaterial objects often change.

[68] Thunman, Carl, G. (1992) pp. 61-74
[69] Thunman, Carl, G. (1994) pp. 173-190
[70] Tritton, Guy (1996) p. 375
[71] Karnell, Gunnar (1985) pp. 19-20
[72] Karnell, Gunnar (1985) p. 20

8.3 Piracy

Licensing has been abused by the fact that a license on software easily can be broken, but the infringed license is not easily detected.

Software can be distributed in several ways – by downloading through network protocols like http or ftp, in a client-server environment. There are also new areas for distribution of software and other content; peer-to-peer networks (Gnutella, JXTA), where the distribution is transacted through equals (client-client) rather than through client-server relationships.[73] An XML Web Service can be seen as an alternative way of distributing software.

Within this field there is an interesting discussion on "grid computing" and distributed computing. The grid is a concept that is on a higher level than web Services. Grid computing could utilize Web Services protocols, but also pure ASP and SSP, EAI and peer-to-peer computing.[74] The concept of "virtual organizations are a possible development here.[75]

What is the difference between regular software that you receive on a CD-rom or downloads from the Internet and a Web Service? You don't download any software when you use a Web Service. The content that is distributed through the Web Service, however, will be licensed in some way, probably by means of a EULA or a Click-wrap-type of agreement. There are Web Services that comes with software to be installed, because if you don't have specific software installed, the machine cannot read the information. This software will probably be offered by third-party software manufacturers or as a part of the operating system. If you however access the Web Service through a portal in a web browser, you don't have to install anything else than the web browser, as the provider of the portal handles the implementation of the Web Service.

The problem with the EULA will be that their content will need to be changed.

§ 1 of the Swedish Copyright Act establishes protection for literary and artistic works created by a creator. Protection is granted for works that are expressed in text, music or images for example. Computer software is protected according to the copyright act § 1. A great deal of the content of an XML Web Service will consist of database content that is licensed. Databases are protected according to § 49 of the Copyright Act, which protects databases,

[73] Holmgren, David & Johansson, Tomas (2002) pp. 3-4
[74] Foster, Ian, Kesselman, Carl, & Tuecke, Steven, (2001) pp. 15-16
[75] A.a. pp. 3-5

charts and similar items. Copyright protection gives the creator the right to make copies of the work and to make he work publicly available. There are no legal demands for the type of media the copy would have been expressed in for the copy to be viewed as a copy, which means that also a digital copy may be a copy according to the Copyright Act.[76] The production of copies is an exclusive right for the creator, which means that other copying has to be approved by the creator. Exceptions from this rule is placed in chapter 2 of the Copyright Act, one of them being copying for private use according to § 12. The copying of software is excluded from the exception according to § 12 para. 2 p. 2, although there exists a mandatory right to make security copies in § 26 h.

Piracy is a major problem for the licensors of software today. There have been several solutions to this problem, including copyright protection gadgets of different kinds. One of the major problems with piracy is that the pirates will have the software locally so they can investigate the source code. They have copies of the software because software needs to be distributed in order to be installed and run. When it is distributed, there are few ways that the licensor can control that the user of the software actually is the rightful licensee.

In a sense the XML Web Service is a way of distributing software. Software can be distributed in several ways – by downloading through network protocols like http or ftp, in a client-server environment. There are also new areas for distribution of software and other content; peer-to-peer networks (Gnutella, JXTA), where the distribution is transacted through equals (client-client) rather than through client-server relationships.[77]

Within this field there is an interesting discussion on "grid computing" and distributed computing. The grid is a concept that is on a higher level than web Services. Grid computing could utilize Web Services protocols, but also pure ASP and SSP, EAI and peer-to-peer computing.[78]

XML Web Services might be considered to be many things, but what it boils down to is a way of charging for software and distributing it. That might create a situation where piracy or illegal copying of software will be less of a problem. Since a service is run somewhere else than locally on a computer, there is less need for illegal copying since the software isn't available for distribution. Access to the service will be available through other means, for

[76] Rosén, Jan (1998), p. 88
[77] Holmgren, David & Johansson, Tomas (2002) pp. 3-4
[78] Foster, Ian, Kesselman, Carl, & Tuecke, Steven, (2001) pp. 15-16

example by codes that are connected to certain computers that can utilize the service. This will put the control over the use of the service at the context provider, instead of locally at the user as with locally installed software. The possibilities for reverse engineering might also be lessened, which could be a problem. Will creators and context providers be forced to hand out copies of the source code in order for the needs for reverse engineering to be handled? In the Swedish copyright act the right to reverse engineering is a mandatory right.

One thing that Web Services solves is that software can be paid for as it is used. The possibility of integrating a XML Web Services infrastructure that enables micropayments will make it possible to rent software and content for a small sum. This could also reduce the incentive of copying software and content because of the possibility of unauthorized copying. Copying of software mainly is a problem since there is no payment going to the rights owner. The XML Web Services infrastructure could provide a solution to that question.

8.4 Patent infringement

Computer software has until recently solely been protected by copyright. Patenting of software is becoming a more common demand for protection of software from industry and business. In the U.S. there exists possibilities of patenting software as is. Also in Europe there are proposals for patents of software as is.[79] It is therefore not entirely uncontroversial to claim that software can be patented under certain circumstances. In Swedish law computer software can already today be patented, given that the program has a technical effect, is of a technical character, that it is new and that is substantially differs from what is previously known. Today computer software cannot not be patented in itself in Sweden, but only as a part of a physical technical system.[80]

To patent software can be seen as a positive development, considering the perspective of creating economic possibilities for some inventors. Patent may get negative effects on a young business like the software industry, since a relatively few number of patents may limit the pace of innovation. For smaller developers the entering on a market that is covered by patents may be difficult, because there is the need for licensing a great number of patents. Patenting software has received a lot of negative attention from the software community. The main argument against patenting of software is that a relatively young industry could be

[79] EPO Boards of appeal, IBM/Computer programs (T 935/97), IBM/Computer program product (T 1173/97)
[80] Wallström, Patrik & Pawlo, Mikael (2002) p. 4

hampered by patents that run for 20 years. It is clear to see that software that comprises of an entire product area like spreadsheet software should be vary of a product like Microsoft Excel to patent this area.

The licensing of patented XML Web Services can create difficulties at the context offerer. A XML Web Services can be offered as a service to an end-user who has residence in a jurisdiction where the software component would be covered by a patent. Possibly the software could be deemed to be infringing such a patent even though it is not being offered within the specific jurisdiction, since a service through the internet often is possible to connect to in some way.[81]

9. Norms

Norms are implicit rules and expectations that dictate behavior. Norms in this essay will be viewed in context of the actors. What norms would be important for the actors in the XML Web Services market?

I will also touch upon some of the norms that are of interest in the areas of piracy, patent and contract.

9.1 Interoperability

An important norm that could be fostered in a XML Web Services environment is interoperability, i.e. standards adherence. The standards enable the interoperability of the different components of the market. But this is only true if the standards are implemented as they should. A standard that is not implemented will disappear and be replaced by another standard. Therefore it is important for an actor to keep in touch which standards are being adopted by the XML Web Services actors.

9.2 Co-opetition

Co-opetition is another norm that could be fostered in this environment, following Tapscott. Since the actors are dependent on each other in building the market, they have to help each other in different ways. It could be sharing data and plans, for example. But the actors have to be careful not to interfere with competition law rules. At the same time that the actors have to

[81] "Web services opens door to patent infringement problems" Infoconomy September 6th 2002

cooperate, they have to compete. It is common that an actor is active in two roles at the same time, for example a developer and a context provider.

9.3 Customer-centricity

The customer should be put in the center of the applications. In an area as competitive as the XML Web Services market – considering that a XML Web Service can be exchanged for another deliverer almost instantly – a provider of XML Web Services has to stay on the edge of what the customer deserves. This means also that it is important to keep in touch with the end-users and communicate with them and incorporate features that they want to have. Otherwise, the customers might vote with their feet.

9.4 Piracy

In order to guard of piracy, an important norm is to guard the software developers' rights. Counter to this norm is that software and information should be free.

9.5 Patent

The norm for using patents is to create monopoly markets for the inventors in order to create incentives to foster more innovation. Strangely enough the software market seems not to be driven by patents, judging from the strong resistance to it. If there were shorter patent times, the resistance might be easier to bear.

9.6 Contract

Trust is an important element in transacting in any market. If the actors cannot be sure of that there is a secure connection between the purchaser and the seller, or that the methods of payment are working properly, there will be few actors that will use that particular market. To make sure that a contract is established will secure the possibilities of enforcing the contract with the aid of state-based contract law. As the XML Web Services market get these systems up an running, especially the system of micropayments, and that these systems are running on standards that are trusted by the actors, then there should be an emerging trust in the market itself.

10. Discussion

I will now make a brief discussion on each subject and put together a summary.

10.1 Piracy

The XML Web services market might affect the illegal copying of software.

10.1.1 Law

The laws that would counter piracy, i.e. IPR protection are comprehensive. Even so, legislation is bound to fall behind. There are still no efficient mechanisms to monitor adherence to the laws that prohibit software piracy. Legislation can be effective in stopping industrial-type copying and selling of software, but it seems to be less effective in the widespread copying among private citizens.

10.1.2 Norms

The norms that guard the software developers' rights can be important tools to limit piracy. By focusing on the actual individual developers situation there might be a possibility to create an understanding of that piracy is really about infringing on creators individual rights.

10.1.3 Market

The market need to have a structure that can create possibilities that can create value from R&D. Piracy is limiting the possibilities for this. A solution would be to change the cost-structure in cutting prices on software to create incentives for buyers of software to buy software instead of pirating them.

10.1.4 Technology

XML Web services could limit piracy by offering rental of software. The technical possibilities are present, considering that the systems for micropayments are adjusted for the XML Web Services structure. By creating an environment of renting software, the need for surveillance of software use can be limited. This also limits the problems with issues of privacy that arise when surveillance of software use is coming closer to the desktop.

10.1.5 Conclusion piracy

XML Web Services seems to offer some solutions to the difficult to solve problem of software piracy. By offering solutions like renting of software on a per-use basis, the norms among copiers of software may change. If these norms change there might be a lessening of the ideas that raises privacy concerns in order to stop piracy among some of the actors.

10.2 Patent

The institute of patents can create problems for the XML Web Services market.

10.2.1 Law

Patents for software are a reality in some countries and it's an issue that is being vigorously discussed. A software patent made real may create new incentives for some software developers, but mostly, patents for software are treated with skepticism among the software community. The argument would be that only the larger actors would have benefit of such an

institute. The idea that XML Web Services would be infringing patents in countries that the XML Web Services are not even offered in is not a pleasant idea, and it is somewhat destroying the idea of the patent institute, since there is no such thing as a world patent.

If a XML Web Service would be considered to infringe on patents issued in a country where the products market is not considered to be, this could damage the entire XML Web Services industry. It would be hard not to be accused of infringing a patent issued under these circumstances. The situation has not emerged yet, so there should be a discussion when and if such a situation would arise in a court case. A tentative argument would be that if a XML Web Service were offered in a country where it has no market presence, and an end-user uses the product unbeknownst to the context provider, then the intention of the context provider should be of importance. The reason is that if the context provider is making it as difficult as possible to connect to the service from another country, then it might not be infringing a patent in another country. There should be a reasonable way to protect a business from infringing a software patent, since the patent institute is not allowing world patents in other areas of patenting.

10.2.2 Norms

Most actors are comfortable in using patents as a tool for protecting technical inventions. When business models and software patents are being discussed, one might wonder whether the patent institute in itself would be damaged. The reason for this being that it would be an easy task to infringe on these patents. If that would be too easy, it may not create the trust in the patent system that is needed. If a system does not have trust from its actors, a patent would become an empty tool for protecting the developed IPR. It may also be difficult in changing a markets use of copyright as protection for software. Such a change could take a long time to implement.

10.2.3 Market

Software markets seem not to focus on patents in the majority of cases at present. Considering business models and software patents in conjunction, then there would be an incentive for creating more products that are not only focusing on software development.

10.2.4 Technology

Since there would be a possibility for infringement even though the software product is not offered in the country in question. A reasonably simple solution to this problem would be to create a XML Web Service component that could detect if a XML Web Service was used from a country that has not been considered to be a market for the product.

10.2.5 Conclusion patent

The idea of patents being infringed in another country, and that a context provider could not in practice escape from doing it is a problem that should be solved. It is doubtful that something other than a solution on the legislative level could solve this problem, since the patent institute is based on a legislative cooperation on a multi-state level. Another solution that could make the patent a more viable solution is to make the patents duration shorter. That would probably create better incentives for developers in fast moving market such as the XML Web Services market to compete for a short-term monopoly situation in order to monetize from its R&D, but still allowing competitors to stay in business and develop competing products.

10.3 Contract

The issue of contracting is of special importance in the XML Web Services market.

10.3.1 Law

The notion of the principle pacta sunt servanda is prevalent in contract law. But there is not much use of that principle if there is no contract established. The law of contract can be a good basis for interpreting law. In the area of contract law, I believe that there should be a conservative stance to the change of contract law in relation to contracting in cyberspace.

10.3.2 Norms

It is important for the actors to develop norms for entering contracts in a market that is new. These norms are creating the basis for holding the market together.

10.3.3 Market

A market without aid from other areas will have difficulties in managing itself. The market needs help in order to build and sustain trust, especially when it will provide goods and services in exchange for money or is equivalents.

10.3.4 Technology

The technology involved in XML Web Services is creating the practical means to create contracts. There is some question if these kinds of means are enough to ensure the actors that contracts can be handled in the XML Web Services markets in an adequate way. Unless there are ways of making the contract parties the contents of the legal obligations, and to make sure that the parties has recognized these obligations are essential elements that possibly could be wrapped into technology.

10.3.5 Conclusion contract

There are ways of combining technological and legal means in order to secure contracts. It has been done before, but the contracts have been based on standards contracts that could not be negotiated. To be able to negotiate contracts automatically in the XML Web Services environment, this should be based on standard methods of interaction that can be trusted by all parties. Awaiting those methods are being constructed, there should be a caution on relying on purely "technical" contracts that are offered. The legal significance of contracts of that nature is questionable as to the rules that a court would apply on these kinds of contracts.

11. Discussion: to build markets and legal issues

I will now discuss some of the issues previously touched upon and try to summarize the creation of a virtual products market.

11.1 Standards

In order to create a market with big acceptance among other actors a creating of a standards body is important. The standards-body is created by other actors. The standards-body will serve as a place for co-operation around the work with the technology. The standards-body will simplify the co-operation between competitors because competitors need to co-operate in order to act in a b-web. The standards-body will provide advantages because actors that are normally competitors can co-operate on IPR that is licensed to the standards-body. Conflicts concerning the standard can be handled by the standards-body. It also gives the actors a possibility to take part in the development of the standards. This is true even for smaller actors. By laying a foundation for the co-operation with a standards-body the other actors can focus on creating products that will be placed on the created market.

A standards-body is also important since it doesn't need any national ties. The actors are coming from different countries with different legislation. It would be extremely difficult with unilateral negotiations between these actors to create something new that would have been of use for all actors.

When the foundation for the market has been made by the standards the actors can move on. They have developed their own products that are covered by IPR, which now can be made into virtual products. The IPR can now be commercialized due to the developed infrastructure. The commercialization is created without the need for buying or selling of IPR, but is instead enabled through licensing and service agreements. By licensing the actors keep the ownership of the IPR at the same time that they can increase revenue.

By licensing its content a content provider can license text, images, sound as before. The developers that deliver content in the form of software can now license their software to context providers instead of directly to the end-users. By doing that they can limit the risks of illegitimate copying and instead be given the opportunity to sell time for the use of software via a third party.

The context provider can as a provider of services have a possibility to generate revenue by renting software to end-users. The context provider could offer the virtual products that have not been able to offer without the complicated net of actors that created the market and the products that were created for the market. The possibilities were created by creating the needed actors. The actors could then act towards each other in order to create needed prerequisites to make possible the selling of the IPR-based products that the actors had created.

12. Conclusion

It is now possible for me to reconnect to my previously stated research questions.

- Are parties that engage in automatic transaction in a standards-based system obligated towards each other?

Clearly there are uncertainties built in to the XML Web Services market. This can be resolved by a more close cooperation between technicians and lawyers to make sure that the agreements that these kinds of systems are producing are legally binding.

- What changes may XML Web Services create in creating a solution to diminish illegal copying of software?

The XML Web Services market seems to offer a somewhat hopeful solution to the problem of software piracy. By licensing the software through rental and limiting the movement of physical distribution of software the possibilities for piracy are dramatically decreased.

- Are there risks for XML Web Services to be subject to patent infringement?

The risks with patent infringement in the XML Web Services market are real. However, the development of software patents are not entirely clear, since the patent institute itself in this area is presently being discussed. In this area much uncertainty still exists.

The XML Web Services vision is an exciting environment that promises to create new business. But there is a danger that the efficiency of technology brings blindness to the legal issues. The legal issues are always important, especially in areas where there are no apparent legal institutions. Therefore these issues need to be discussed.

It is important that companies make sure that their technical processes that touch upon XML Web Services are covering the legal areas that they are supposed to cover. It is therefore important to include lawyers already in the development stage of such technical systems. To be using a lawyer when legal problems has already arisen is not recommended. A lawyer needs to be present during the entire process of development.

In this essay I have tentatively investigated issues that is of importance for the development of the XML Web Services market. There are several issues that are not solved, and probably a number of problems that has not emerged yet. No doubt there will be a great deal of transactions enabled through the XML Web Services market. Undoubtedly, this market will provide a continuously rich area for research in the legal field for many years to come.

References

Andersson, Kenneth (2001) XML for B2B, Chalmers University of Technology: Masters Thesis

Arora, Ashish, Fosfuri, Andrea & Gambardella, Alfonso (2001) Markets for Technology. The Economics of Innovation and Corporate Strategy. London: The MIT Press

Chaffey, Dave (2002) E-Business and E-Commerce Management. Essex: Pearson Education Limited

Christou, Richard (1996) International Agency, Distribution and Licensing Agreements. London: FT Law & Tax

Edqvist, C. & Johnson, B (1997) "Institution and Organizations in systems of innovation", in Edqvist (ed.) Systems of innovation: Technologies, Institutions and Organizations, Pinter, London

Foster, Ian, Kesselman, Carl, & Tuecke, Steven, "The Anatomy of the Grid. Enabling Scalable Virtual Organizations." pp. 15-16. International Journal of Supercomputers (2001)

Graham, Steve et. al (2001) Building Web Services with Java. Making Sense of XML, SOAP, WSDL, and UDDI. Sams Publishing, Indianapolis

Geurts, Rik (2002) "Legal Aspects of Software Agents" in Prins, J.E.J., Ribbers, P.M.A., van Tilborg, H.C.A., Veth, A.F.L. & J.G.L. van der Wees (eds.) Trust in Electronic Commerce. The Role of Trust from a Legal, an Organizational and a Technical Point of View. Dordrecht: Kluwer Law International

Holmgren, David & Johansson, Tomas (2002) Secure Distribution of Files Using Peer-To-Peer, Master's thesis, Luleå University of Technology

Hultmark, Christina (1998) Elektronisk handel och avtalsrätt. Stockholm: Norstedts Juridik

Hsu, Cheng & Somendra Pant (2000) Innovative Planning for Electronic Commerce and Enterprises. A Reference Model. Norwell: Kluwer Academic Publishers

Karnell, Gunnar (1985) Introduktion till den Internationella Licensavtalsrätten. Stockholm: Norstedts

Lessig, Lawrence (1999) Code and other Laws of Cyberspace. New York: Basic Books

Lindmark, Sven (2002) Evolution of techno-economic systems. An investigation of the history of mobile communications. Gothenburg: Department of Industrial Management and Economics

Longworth, Elizabeth, "The possibilities for a legal framework for cyberspace – including a New Zealand perspective", in The International dimensions of Cyberspace Law, (2000) Aldershot: Ashgate Publishing

Lundblad, Nicklas and Magnusson Sjöberg, Cecilia, Making money from information standards: Legal issues and cost structures in XML-related projects. XML Europe 2003, 5-8 May, 2003, London, Conference Proceedings. 22 pp

Magnusson Sjöberg, Cecilia, The Melting Pot Paradox of Structured Documents. In: Law and Information Technology, Swedish Views: An anthology produced by the IT Law Observatory of the Swedish ICT Commission, SOU 2002:112, pp.195-206

Mitrakas, Andreas (1997) Open EDI and Law in Europe. A regulatory framework. Dordrecht: Kluwer Law International

Moore, James F. (1996) The Death of Competition. Leadership and Strategy in the Age of Business Ecosystems. Chichester: John Wiley & Sons

Nooteboom, Bart (1999) Inter-firm alliances; Analysis and design. London: Routledge

Karnell, Gunnar (1985) Introduktion till den Internationella Licensavtalsrätten. Stockholm: Norstedts

Korah, Valentine. (1990) An introductory guide to EEC competition law and practice. Oxford: ESC

Lindberg, Agne, Westman, Daniel (2001) Praktisk IT-rätt. Stockholm: Norstedts Juridik

Lundvall, Bengt-Åke "Standards in an innovative world" in Standards, innovation and competitiveness: the politics and economics of standards in natural and technical environments (1995) (Ed.) Hawkins, Richard, W., Mansell, Robin & Skea, Jim, Aldershot, UK: Edward Elgar

Ramberg, Christina (2002) Internet marketplaces. The law of auctions and exchanges online. New York: Oxford University Press

Rosén, Jan (1998) Upphovsrättens avtal. Stockholm: Norstedts Juridik

Spivak, Steven M. & Brenner, F. Cecil (2001) Standardization Essentials. Principles and Practice. New York: Marcel Dekker

Söderberg, Johan & Zenno, Patrik (2002) Web Services. Degree of Establishment in Swedish Consultancy and Software Companies. Bachelor's Thesis, Informatics, Jönköping International Business School

Tassey, Gregory "The roles of standards as technology infrastructure" in Standards, innovation and competitiveness: the politics and economics of standards in natural and technical environments (1995) (Ed.) Hawkins, Richard, W., Mansell, Robin & Skea, Jim, Aldershot, UK: Edward Elgar

Tapscott, Don, Ticoll, David & Alex Lowry (2000) Digital Capital. Harnessing the power of business webs. London: Nicholas Brealey Publishing

Thunman, Carl G "Managing international licensing relationships". In: Forsgren, M & Johansson, J. (eds.), (1992) Managing Networks in International Business. Philadelphia: Gordon & breach, pp. 61-74

Thunman, Carl G. “Planning and controlling the licensing process” In: Brooke, M. Z. & Skilbeck, J.M. (eds.), (1994) Licensing – The International Sale of Patents and Technical Knowhow. Aldershot: Gower, pp. 173-190

Tritton, Guy (1996) Intellectual Property in Europe. London: Sweet & Maxwell

Tornéus, Mats, “EDI och EDI-avtal i praktiken” in Wahlgren, Peter (ed.) 1999 Promemorior IT-rätt, Stockholm: Jure

Turban, Efraim, King, David, Lee, Jae, Warkentin, Merril, Chung & H. Michael Chung (2002) Electronic Commerce 2002. A managerial perspective. New Jersey: Pearson Education Inc.

Wallström, Patrik & Pawlo, Mikael (2002) Något om patenterbarhet av datorprogram i svensk rätt. Diskussionsunderlag. Det svenska IT-rättsliga observatoriet. Rapport 2003:61

Warsta, Juhani (2001) Contracting in Software Business. Analysis of evolving contract process and relationships. Oulu: Oulu University Press, p. 34

Westman, Daniel (1998) *Rättspolitik på IT-området – ett diskussionsunderlag*. Det IT-rättsliga observatoriets rapport 9/98

Law sources

Act (1960:729) on Copyright in Literary and Artistic works
UNCITRAL Model Law on Electronic Commerce:
http://www.uncitral.org/english/texts/electcom/ml-ecomm.htm

Case law

EPO Boards of Appeal:
IBM/Computer programs (T 935/97)
IBM/Compueter program product (T 1173/97)

Public publications

SOU 1996:40 *Elektronisk dokumenthantering*.
SOU 2002:51 *Breddtjänster – ett nytt skede i IT-politiken*.
SOU 2003:35 *Digitala tjänster – hur då? En IT-politik för resultat och nytta*.

Web-based material

Bakos, Yannis (1998) The Emerging Role of Electronic Marketplaces on the Internet. http://pages.stern.nyu.edu/~bakos/retail-ecommerce.pdf (Downloaded 20030420)

Deloitte & Touche (2002) Web Services: The Next Evolution in Software. Available through: http://www.looselycoupled.com/blog/2003_03_02_lc.htm#90402167 (Downloaded 20030309)

IBM. Web Services Conceptual Architecture
http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf
(Downloaded 20020604)

Web Services Infrastructure: The global utility for real-time business, http://www.philwainewright.com/pubs/wp/WSIpaper.pdf (Downloaded 20020613)

Web-pages

http://www-3.ibm.com/e-business/index.html
http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf
http://www.infoconomy.com/
http://www.karelia.com/
http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000442
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnWebsrv/html/Websvcs_platform.asp
http://www.oasis-open.org/committees/guidelines.php
http://www.oasis-open.org/who/intellectualproperty.php
http://www.oreillynet.com/pub/a/webservices/2002/02/12/webservicefaqs.html
http://www.rosettanet.org/