Titlul referatului

SISTEME PENTRU COMERTUL ELECTRONIC

Autor

Ciprian Cândea

Coordonator de doctorat

Acad. Florin Gh. Filip

BUCUREȘTI, December 2011
<table>
<thead>
<tr>
<th>Capitolul 1</th>
<th>INTELIGENȚĂ ARTIFICIALĂ</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capitolul 1</td>
<td>e-Commerce</td>
<td>4</td>
</tr>
<tr>
<td>1.1</td>
<td>Overview</td>
<td>4</td>
</tr>
<tr>
<td>1.1.1</td>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>1.1.2</td>
<td>What is e-Commerce</td>
<td>4</td>
</tr>
<tr>
<td>1.1.3</td>
<td>e-Concept</td>
<td>6</td>
</tr>
<tr>
<td>1.1.4</td>
<td>E-Commerce supporting technologies</td>
<td>6</td>
</tr>
<tr>
<td>1.1.5</td>
<td>E-Commerce and related basic sciences</td>
<td>10</td>
</tr>
<tr>
<td>1.2</td>
<td>Categories of e-Commerce</td>
<td>13</td>
</tr>
<tr>
<td>1.2.1</td>
<td>B2B</td>
<td>14</td>
</tr>
<tr>
<td>1.2.2</td>
<td>B2C</td>
<td>15</td>
</tr>
<tr>
<td>1.2.3</td>
<td>C2C</td>
<td>16</td>
</tr>
<tr>
<td>1.2.4</td>
<td>B2G, G2G and G2B</td>
<td>17</td>
</tr>
<tr>
<td>1.3</td>
<td>Supporting tools for e-Commerce</td>
<td>18</td>
</tr>
<tr>
<td>1.3.1</td>
<td>Portals</td>
<td>18</td>
</tr>
<tr>
<td>1.3.2</td>
<td>Client Intelligence</td>
<td>19</td>
</tr>
<tr>
<td>1.3.3</td>
<td>Decision Support</td>
<td>21</td>
</tr>
<tr>
<td>1.4</td>
<td>Supporting Technologies for E-Commerce</td>
<td>24</td>
</tr>
<tr>
<td>1.4.1</td>
<td>Global Positioning System (GPS)</td>
<td>24</td>
</tr>
<tr>
<td>1.4.2</td>
<td>Geographical Information System (GIS)</td>
<td>25</td>
</tr>
<tr>
<td>1.4.3</td>
<td>Decision Supporting System (DSS)</td>
<td>26</td>
</tr>
<tr>
<td>1.4.4</td>
<td>Group Decision Supporting System (GDSS)</td>
<td>28</td>
</tr>
<tr>
<td>1.4.5</td>
<td>Intelligent Decision Supporting System (IDSS)</td>
<td>29</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capitolul 2</th>
<th>Knowledge in E-Commerce</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Ontologies</td>
<td>31</td>
</tr>
<tr>
<td>2.2</td>
<td>Ontologies Technologies</td>
<td>33</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Frame based languages</td>
<td>33</td>
</tr>
<tr>
<td>2.2.2</td>
<td>First order logic</td>
<td>34</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Description logics</td>
<td>35</td>
</tr>
<tr>
<td>2.2.4</td>
<td>The OWL language</td>
<td>35</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Knowledge representation</td>
<td>36</td>
</tr>
<tr>
<td>2.2.6</td>
<td>Reasoning mechanisms</td>
<td>40</td>
</tr>
<tr>
<td>2.3</td>
<td>Ontologies Use Cases</td>
<td>40</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Use Case – Neutral Authoring</td>
<td>42</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Use Case – Common Access to Information</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Data Access via Shared Ontology</td>
<td>45</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Data Access via Mapped Ontologies</td>
<td>46</td>
</tr>
<tr>
<td>2.3.5</td>
<td>Shared Services</td>
<td>46</td>
</tr>
<tr>
<td>2.4</td>
<td>Ontologies on E-Commerce</td>
<td>47</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Scenarios and Questions that GoodRelations ontology is addressing</td>
<td>47</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Domain concepts</td>
<td>49</td>
</tr>
<tr>
<td>2.4.3</td>
<td>URIs for E-Commerce on the Semantic Web</td>
<td>53</td>
</tr>
<tr>
<td>2.5</td>
<td>Architectures for Integrating CBR-Systems with eCommerce</td>
<td>54</td>
</tr>
</tbody>
</table>
Chapter 1  e-Commerce

1.1 Overview

1.1.1 Introduction

E-commerce brought about by Internet is one of the most significant scientific accomplishments. In business, the prosperous e-commerce technology gives rise to a revolution in the circulation system. It breaks the boundary of time and space, alters the trade pattern, improves the circulation of merchandise, capital and information, and makes enterprises have an edge over others as well by reducing the cost of production effectively. In short, e-commerce has enabled the traditional business to achieve greater, faster, better and more economical results. The influence of the e-commerce will go beyond the business activity. It will make a profound impact on each aspect of human society, such as production and employment, government function, working talent, law systems and education, etc. It permeates into every profile: industries, logistics, finance, media, governments, enterprises, research organizations and even traditional agricultures. With the development of the e-commerce, it will influence and impact to a larger extent every aspect of our society with each passing day. We can say without exaggeration that the electronic commerce is the most significant industrial revolution since Industrial Revolution, with deeper influence on mankind than the former two industrial revolutions, because it not only can raise greatly productivity, efficiency of economic operations, lower the economy operation cost, and make many originally impossible things possible, but also influence people’s life styles and every social aspect and therefore change their world outlook and methodology.

1.1.2 What is e-Commerce

As described by Yan Tian and Concetta Stewart (2006) e-Commerce or electronic commerce, also known as e-business, refers to the transaction of goods and services through electronic communications. Although the general public has become familiar with e-commerce only in the last decade or so, e-commerce has actually been around for over 30 years. As the term suggests, e-commerce refers to various online commercial activities focusing on commodity exchanges by electronic means, Internet in particular, by companies, factories, enterprises, industrial undertakings and consumers. A large number of well-known organizations and corporations also have their own definitions on e-commerce. For example, ISO defines e-commerce as: it is the general term for exchange of information among enterprise and between enterprise and customers; the Global Information Infrastructure Committee defines it as the economic activities using electrical communications, with which people can purchase products, advertise goods and settle.
The following are definitions given by transnational corporations:

- Intel: E-commerce: electronic market + electronic trade + electronic service
- IBM: E-commerce: information technology + web + business
- HP: E-commerce is to accomplish commercial business by electronic means.

Since e-commerce is new in science, it is not at all surprising that there are various definitions about it. E-commerce shall be social, economic activities between social principal parts by taking advantages of computers and network.

- Network: It includes Internet, Intranet, and Extranet. Internet is the foundation of e-commerce and the carrier of commercial business information. As to Intranet, it means for enterprises to carry out internal affairs. With regard to Extranet, it is the link between enterprises and users to carry out commercial activities.
- E-commerce user. It includes personal consumers and business consumers. The business consumer scientifically manages staff, wealth, goods, production, supply and sales by Intranet, Extranet and MIS. Personal consumer has access to information and purchases goods by connecting Internet with browsers, set-top boxes, PDA (the personal digital assistance), Visual TV etc.
- Authentication Authority: The authentication Authority (CA), the authority recognized by law, is responsible for issuing, managing digital certificates and facilitating parties involved in online sales to identify each other.
- Distribution center. It is in charge of sending goods that cannot be delivered on line to consumers and keeping track of goods flow.
- Online bank. It provides the sellers and buyers the traditional bank business, such as settlement, and round-the-clock service.
- The administration of the commercial activity. It consists mainly of departments of industry, customs, tax and trade.
1.1.3 e-Concept

E-mail transforms from receiving and sending e-mails to commercial use and becomes one of the principal tools for people to communicate on line. The e-wallet and e-cash etc. basically make it possible for people transfer fund on line and accordingly makes network and real life fuse further.

E-mail, attracts people to pay attention to the existence of the network, and sparks the rush of network gradually.

The e-commerce has on a large scale began to impact the traditional pattern of trade and brought about intense changes of life style.

In future we can see that new concepts will spring up and the combination of electronic technology’s and other ideas will give rise to new things, because electronic technology based on Internet is advancing continuously. Already a major impact of social media o the e-Commerce is seen, as well at the decision level – how people obtain information and cooperate to take decisions at individual level or group level.

E-concept is a new thing taking electronic and network technology as foundation, other technologies and ideas as platform of upper level. After double integration with other technologies and ideas, e-concept will exert a great impact on social life style. In this sense, e-commerce is just a subset of e-concept. E-commerce gives impetus to the development of society and provides broad space for future, though it is merely a subset of e-concept. The development of e-commerce may transform the whole social operation regulations, so the e-concept will exert much more influence on the society, no matter in depth or in scope. The significance of e-concept lies in that it makes people grasp not only radical changes of new things but also the root of all changes brought by e-concept in e-times so as to promote greater social changes.

1.1.4 E-Commerce supporting technologies

The presence of e-commerce not only influences the traditional trade process but changes the components of market to a large extent. Traditionally, market transaction chain is formed in the exchange process of commodity, service and currency. At present, owing to e-commerce, information is reinforced and brings about information of commodity, information service, electronic money etc. Though the essence of e-commerce remains unchanged, some links in the transaction process have changed because of the changes of e-commerce carriers. From the perspective of individual enterprise, trade method has taken some changes, while from the perspective of the whole business environment, the whole business has taken on a new look, for some business losing opportunities while some new one grasping the opportunities, and some industries decline while some flourishing.

In next figure briefly shows the framework and principal factors in the environment. As the figure shows that, e-commerce consists of four levels (Internet infrastructure, multimedia and online publishing infrastructure, newspaper and media infrastructure,
newspaper and information communication infrastructure and business service infrastructure) and two mainstays (public policies, laws, privacy, and other standards of various technologies).

- **Infrastructure**

  This level consists of hardware infrastructure to realize e-commerce. The infrastructures are systems for information transmission, including remote communication network, cable TV network, telecommunication network and Internet etc. The above-mentioned networks provide the transmission line for e-commerce, while at present most applications of e-commerce are based on Internet consisted of telephones, modems, hubs, routers, etc.

- **Multimedia and online publication**

  The increasing bandwidth of Internet infrastructure makes the transmission of information through network possible, such as texts, sounds and images, video, etc. At present new media is present and new standards like HTML5 are in use, by which the multimedia contents can be easy to search and expressive.

- **Information communication infrastructure**

  This level mainly supplies tools and methods to transmit information.

  - Non-formatted data communication. Example: information is transmitted to people by making use of fax and e-mails, social media, in which man shall take part.
  - Formatted data communication. For example, information is transmitted to machines automatically by EDI, in which man shall not take part.
- **Service infrastructure**

Standard online commercial business services are managed on this level. The above service will be used by all enterprises and persons in doing business, including establishment of commodity catalog/price list, electronic payment, secure transmission of commercial information, authentication of legality of both sellers and buyers. For the e-commerce, the transmission of information shall meet the requirement of e-commerce and provide a secure authentication mechanism to guarantee the data are reliable and undeniable. Presently, many technologies are for this purpose, such as password, digital certificate, and SET protocol.

- **Public policies, laws and privacy**

Public policies comprise the following policies centering on e-commerce, which shall be worked out by government. They are taxation regulations, information pricing, information access fee, information transmission cost, and privacy etc. Laws maintain the regular operation of e-commerce. From the perspective of law, authentication of e-commerce refers to the reliability and security of data and products involved in e-commerce. Because e-commerce is same as the traditional trade, a serious social activity, two parties involved in e-commerce shall join in the market with true identity, and provide real information, that is the real e-commerce. The security of e-commerce refers to the parties involved are in duty bound to keep the documents/products of the other party secret when not authorized to give publicity to them, for these documents and products are all real. The first step of establishing e-commerce law is to establish the system of e-commerce authentication.

- **Technology standards**

Technology standard is the basis of information releasing and transmission and the guarantee of agreement of online information. It defines the technology details of customer interface, transport protocol, releasing information standard, security protocol and so forth. From the perspective of the whole network environment, standard is of significance for compatibility and universality. Similar problems have arisen in e-commerce, such as EDI standard, and ebXML standard. Some international credit organizations as VISA, MasterCard with all social circles have worked out SET protocol for secure e-payment.

1.1.4.1 **EDI**

EDI is an e-technology that supports business-to-business transactions and is principally used by large organizations that have a large volume of standardized transactions. There are a number of definitions around, but (unlike definitions of e-commerce and e-business) they are all different ways of saying the same thing. The definition described by Parfett, 1992, adopted by the UK National Computing Centre is the “transfer of structured data,
by agreed message standards, from one computer system to another, by electronic means.”

Definition keywords are detailed:

- “structured data,” brings most trade exchanges within the potential scope of EDI. A simple business-to-business trade exchange consists of an order, delivery note, invoice, and payment, and these are all structured data. The utility and preciseness of the trade exchange data is enhanced by the use of codes in place of addresses and product descriptions. The Article Numbering Associations (ANA) is one set of organizations that have devised appropriate coding systems; EAN (European Article Number) and UPC (Universal Product Code) are examples of this. Lauden and Traver, 2004, explain that the scope of EDI has also been extended from basic trade documents to requirements such as customs clearance, container information, and continuous replenishment. EDI is also used for other applications, for example, in banking (electronic funds transfer, EFT), education, medicine, and weather forecasting.

- “agreed message standards” implies a message coding system that has wider application than just the two (or a small group of) organizations involved in trade exchanges. EDI standards were initially devised by industry-sector organizations and often on a national basis. A difficulty with the early message standards was that they were trade-sector or nationally based. As the use of electronic trade exchanges expanded to a wider range of products and across international borders, suppliers were increasingly being asked to implement more than one standard. The answer to this problem was to devise a standard that was more widely accepted and catered for a greater range of requirements. This resulted in the ANSI X12 standard in the United States, and EDIFACT (EDI for Administration, Commerce and Transport) in Europe. EDIFACT was developed under the auspices of the United Nations (UN) Economic Commission for Europe but was subsequently adopted as the international standard at a meeting of the International Organization for Standards in September 1987. The standard is vast, but that does not mean that an individual transaction has to be complex.

- “from one computer system to another” that distinguishes the application of EDI from most other e-technologies. The concept is of an exchange that directly links the computer application of the customer to that of the supplier. This can be, for example, the stock control or replenishment system of a supermarket chain “talking” to the order-processing system of the supplier of breakfast cereals, or the material and requirements planning system of an automobile assembler issuing just in-time orders for the delivery of components and subassemblies by the supplier of those products. EDI is used for business-to-business transactions that are computer generated, frequently repeated, and computer processed. To gain the full benefit of EDI, there should be no requirement for manual (administrative) intervention at either end. EDI is the e-technology that most effectively automates and integrates the logistics supply chain. It facilitates the creation of an interorganization information system (IOS) integrating the business systems of customers and suppliers, and it can include other players such as shippers, customs authorities, and banks.
“by electronic means” implies the transaction takes place over a network. Traditionally, the network has been a commercial value-added data service (VADS), also known as a value-added network (VAN). A feature common to all these networks is the provision of post and-forward facility so senders can transmit their EDI exchanges at times that suit them, and the addressees can pick them up in accordance with their own operating schedules.

New technologies changed the old style implementation by adopting the XML (extensible markup language) for business to business electronic data exchanges. The advantages of XML are next:

- The standards are simple; you can devise your own (and so can everyone else).
- XML software are readily available and in many cases they are free (although there will still be the need to integrate the XML processing software with the business application).
- EDI is old and, in comparison, XML is new, versatile, and exciting.

Many EDI software packages now includes the XML encoding and here is also an advantage that by using XML, a smaller trading partner can use XML style sheet and free XML software to access the transaction and to integrate them into their open source software for e-Commerce.

### 1.1.5 E-Commerce and related basic sciences

E-Commerce development is based on the development of many other sciences and technologies. Sciences of mathematics, computer, communications, and management have great influence on the development of e-commerce soft environment. The development of logistics distribution, postal service, communications, and electronic technology exert great influence on the development of e-commerce hard environment.

Because e-Commerce is based on evolution of all these disciplines can be considered that e-Commerce is impacting all of them in order to respond to new challenges and vice-versa.

#### 1.1.5.1 Computer science

E-commerce is based on electronic information technology, in which information storage, information exchange and information processing are all carried out by computers and Internet. The developments of computer science and computer technology have a decisive effect on the development of e-commerce.

The computer network has already become principal platform of e-commerce. The universality of TCP/IP greatly facilitates the transmission of information between the networks conveniently, lowers the cost of information acquisition; the emergence of routing technology and the continuous creative innovation of the routing algorithms give full play to network, thus comes the global outburst of Internet. At the same time, the
rapid development of the intranet and extranet, based on IP, has turned into the foundation for enterprises to carry out e-commerce.

Computer languages have developed, from machine language, to assembly language, advanced language and even to the intelligence language, and many operating system software’s, business software’s, application software’s, specialized application software’s and integrated development software’s. All the above changes speed up software development, free people from the tiresome mental work of source code compiling to satisfy the basic need to assemble many basic software modules of excellent functions. The data base technology, the key technology of e-commerce, cannot only store all kinds of information (e.g. articles, transactions, and consumers) but also provide market information and decision-making basis by rearranging and analyzing data. The key factor to ensure the popularity of e-commerce is the security of information, for which various encryption technologies such as symmetric key encryption and public key encryption are developed to provide the basis of information privacy. In addition, technologies of firewall and intrusion detection keep information from being destroyed and stolen illegally.

1.1.5.2 Communication science

In e-commerce, communication system is the carrier of information flow and capital flow, and exchanges and transmission of information are both carried out through communication network consisting of transmission, switch, terminal equipment’s, signaling, protocol and corresponding operation support systems. It is necessary to build up wide communication channels for information transmission and exchange of e-commerce network. Rapid development of contemporary communication technologies influence e-commerce, because is one of the foundation.

The development of e-commerce demands for higher and higher communication. Safe, high-speed network transmission and multimedia communication have been the inevitable factors to carry out e-commerce.

The mobile communication (the information transmission process in which both parties or at least one party involved carry out information exchange while moving) is the determining technology for information exchange, usage, processing, and releasing between any communication subjects all over the world anytime and anyplace and the core technology to realize Mobile-commerce.

Internet is the technical foundation and indispensable for e-commerce. The development of e-commerce will definitely make greater demands on the communication technology and boost the development of communication technologies; the development of communication technologies provides technical support for new applications of e-commerce. Vice versa, along with the extension of e-commerce applications, the communication infrastructure network can carry on larger-scale information flow and this requires continuous increasing of infrastructure network bandwidth.
1.1.5.3 Management science

At the classical management phase, the focus of management is to raise labor productivity by managing equipment’s and staff manually with strict regulations. Because this phase falls into the category of the seller’s market, a tight market, the main task of governors is to increase labor productivity and supply.

In the behavior management phase, the focus of governors is to transform the volume-produce mode into small-lot versatile production mode. The competition has changed to the service of how to meet the customer’s requirement from pure price competition. In this phase, supply has been guaranteed; customers began to push forward their claims for high-quality products. Products of low-quality are difficult to sell. Since the main task of the managers is to sell high-quality products, small lot and diverse types are manufactured to satisfy various requirements of customers.

![Management phases](image)

Figure 3 Management phases

In the phase of strategic management, administrators have begun to make and carry out decisions dynamically, continuously and adapt themselves quickly to changing internal and external conditions to balance stability, continuity, suitability, and innovation. In this phase, enterprises made use of internal resources and information comprehensively, and collecting and processing of information proposed higher quest to management measures.

Due to the permanent technology progress and of the new market constrains companies facing the never-ending changes, improvements and fierce race. Companies are crying for regrouping of operation flow to improve enterprise operation status and efficiency. Regrouping of operation flow is the effective path for enterprise to regain competitive power and vitality.

The implementation of Business Process Reengineering (BPR, for short) needs two foundations: modern information technology and high-caliber talents.

The need of information processing facilitates the extension of computers in enterprises and the rapid development and popularity of management information system (MIS, for short) and decision support system (DSS, for short), thus accelerating the rapid
development of computer hardware and software, computer information processing technology and message-switching technology as well as setting up the foundation for the development of e-commerce.

In the organizational management phase of globalization and knowledge economy age, the waves of informatization and globalization spread all over the world rapidly. Because of the arrival of the knowledge economy, information and knowledge become significant strategic resources. Only by organizing the global resources reasonably and winning the support of the global customers may enterprises exist and develop. To reasonably organize the global resources, apart from multinational corporations, virtual enterprises also come into being. “Virtual organizations” is coming, which makes greater demands on information acquisition, information processing, and information utility than the former periods. It is this objective need that requires enterprises to connect the former systems of MIS and DSS, speedups enormous progress of Internet with the function of information communication and releasing, and perfects the infrastructure of e-commerce.

### 1.2 Categories of e-Commerce

There are a variety of different types of e-commerce and many different ways to characterize these types. For the most part, we distinguish different types of e-commerce by the nature of the market relationship—who is selling to whom. As to enterprises and customers, e-commerce of different types and levels imply different opportunities. In terms of transaction categories, e-commerce falls into five categories:

- **business to business (B2B),**

  B2B, the mainstream in e-commerce and the principal method to improve competition ability in the competitive market, has come into existence for many years with the characteristics of carrying out commercial activities by EDI via special networks or Value-Added Networks (VAN, for short).

- **business to customers (B2C),**

  B2C takes place between business and customers, in which online sales are carried out by Internet (example: online bookstore Amazon). In recent years, the increase of number of netizens and new transaction platforms created by Internet for enterprises and customers speedups the rapid development of e-commerce. With regard to customers, it is unnecessary to set a unified standard for document transmission because only credit cards, e-money or e-wallet are involved in online sales and payment. In addition, searching and browsing functions and multimedia interface supplied by Internet facilitate consumers to look for and give an insight into products wanted. B2C has enormous potential and will be the main drive for the development of e-commerce.
- business to governments (B2G),

B2G, the business between enterprises and governments, is still in its experimental phase, focusing on administrative management, governmental invited tender, and the implementation of various economic policies etc.

- governments to governments (G2G),

(Also known as e-government). Government- to-government transactions within countries linking local governments together and also international governments, especially within the European Union, which is in the early stages of developing coordinated strategies to link up different national systems.

- customers to customers (C2C).

In this category consumers interact directly with other consumers. They exchange information such as:

- Expert knowledge where one person asks a question about anything and gets an e-mail reply from the community of other individuals, as in the case of the New York Times-affiliated abuzz.com website.
- Opinions about companies and products, for example epinions.com.

There is also an exchange of goods between people both with consumer auction sites such as e-bay and with more novel bartering sites such as swapitshop.com, where individuals swap goods with each other without the exchange of money.

### 1.2.1 B2B

Business-to-Business (B2B) e-commerce is focus on exchange of products, services or information between business entities.

Web-based B-to-B includes:

- Direct selling and support to business (as in the case of Cisco where customers can buy and also get technical support, downloads, patches online).
- E-procurement (also known as industry portals) where a purchasing agent can shop for supplies from vendors, request proposals, and, in some cases, bid to make a purchase at a desired price. For example the auto parts wholesaler (reliableautomotive.com); and the chemical B-to-B exchange (chemconnect.com).
- Information sites provide information about a particular industry for its companies and their employees. These include specialized search sites and trade and industry standards organization sites. E.g. new market - makers.com is a leading portal for B-to-B news.
Companies that are implementing B2B model get some benefits as:

- B2B e-Commerce companies hold an advantage in lowering operation cost

The online business of B2B companies covers production, supply, sales, or purchase, transfer and storage, which are so extensive that it can provide many ways to reduce the cost: First, reducing procurement cost by purchasing multitudinous commodities of single kind. Second, reducing production cost by shortening production period. Third, reducing operation cost by efficient inventory control. Fourth, reducing sales cost by global batch sale. The advantages of above ways are incomparable to e-commerce Company of other categories.

- B2B e-Commerce companies are more suitable for modern logistics management

The logistics distribution plays a prominent role in all the commercial activities of e-commerce companies. There will be no cycle of operations without logistics. Even though there is external logistics, its expense will directly influence the achievements of e-commerce companies. Compared with other e-commerce companies, the characteristics of B2B companies logistics are fewer in times and large in quantity, while logistics of other e-commerce companies are characterized as small in quantity, more in times and high in turnover speed. According to statistics, B2B e-commerce companies are more competitive than B2C e-commerce companies in reducing logistics dispatching by 20% - 60%.

- B2B e-Commerce companies are competitive in guaranteeing credit and capital security during operation course

Credit and security will ever hinder the rapid development of e-commerce whether in the past, at present or in the future, which will obsess e-commerce companies for a long term as long as the complete e-commerce activities are carried out on Internet. As for the credit issue, it is a hard nut to crack, because B2C companies have thousands of clients and it is difficult to explore the credit of clients. Additionally, the frequent less in quantity and more batches online payment makes clients more worried about the security of online payment, which will seriously affect the B2C companies to seize commercial opportunities. Contrary to B2C companies, it is easy for B2B companies to inspect and identify the credit of their trade partners, because of their operation methods, fewer in batch and larger in quantity. Moreover, payment categories of special-purpose communications link of bank networks is adopted to guarantee the safety of capital flow. These will speed-up the development of B2B e-commerce companies with less batch and client stability.

1.2.2 B2C

In B2C (Business to Customer) e-commerce, Internet is resorted by businesses or enterprises to provide customers goods and services via websites. Various types of B2C
websites spread all over Internet to supply customer a variety of goods and services, varying from flowers and books, to computers and cars etc.

From the perspective of the business relations between enterprises and customers, B2C falls into two categories: seller (enterprises)—the personal buyers, and buyer (enterprises)—the personal sellers.

Seller (enterprise)-personal buyers, is the categories in which enterprises sell goods and services to individual customers. In this e-commerce categories, the sellers first open an online store, then release the information on variety, specification, price, and capability of the goods or on variety, price and measures of service, by which the individual customers choose goods, place an order and decide to make online or off-line payment, and last deliver goods to the customers. By this kind of online shopping, customers can acquire further information on goods, shop around, and purchase goods at the lowest cost and save shopping time without going outside. This e-commerce category requires the support of high-efficient, low-cost logistics, the representative of which is the global distinguished online bookstore Amazon (http://www.amazon.com).

Enterprise personal sellers, is the category in which business purchases goods or service from individuals. This category is usually used for online job application. By these categories, enterprises release information on needed talents at first and then applicants negotiate with an enterprise online, which is very popular in this society with big talents flow for it offers a communication platform and bridges the enterprises and individuals, thus making full use of human resources.

E-commerce of B2C can be divided into tangible and intangible goods and services, of which the latter can be completed by network, while the former cannot unless the traditional methods are resorted to.

Thanks to the information transmission and processing abilities of computer network, intangible goods and services (e.g. e-information, computer software, digital audiovisual entertainment products, etc.) in general can be presented to customers directly via network. E-commerce categories of intangible goods and services are mainly online subscription categories, advertisement-supported pattern and online domination categories

1.2.3  C2C

The C2C is a schema of trade of the consumers, sharing similar characteristics with agriculture trade market or flea market. What constitutes it are vendors and purchasers, electronic trade provider suppliers, similar to space suppliers and governors agriculture trade market in the real and the flea market. In the categories of C2C, the electronic trade provider suppliers play the prominent role.

Firstly, in such wide range of the network, if without a well-known supplier trusted by both vendors and purchasers, bunching both vendor and purchaser together, it is very difficult for buyers and sellers to find each other, and also will cause loss of chances.
Secondly, the electronic trade provider suppliers also take the responsibilities of supervision and management for honesty and creditability of vendors and purchasers monitoring their transactions and minimizing the occurrences of fraud in order to protect the interests of buyers and sellers.

Thirdly, electronic trade provider suppliers can offer the technology support to vendors and purchasers, including helping sellers to create online shop, release product information, decide on pricing strategy etc., helping buyers to select products or electronic settlement etc. It is just because of the technology support that a C2C category has been accepted by a large group of users within a short period of time.

Lastly, along with the maturity of pattern C2C, the electronic trade provider supplier can still offer buyers and sellers the financial services like insurance, loans, thus better services for the both vendors and purchasers. Therefore, in the categories of C2C, the electronic trade provider tenders play a significant role, because it directly affects the precondition and foundation of this commerce category.

1.2.4 B2G, G2G and G2B

Governments, as national administrators, play a significant role in guiding, administrating and adjusting economy. The advent of e-commerce age put forward the new request to the original functions of governments. Governments should administrate e-market effectively and render better service to enterprises and the public by e-government on the one hand. Governments, as the “big clients” in economy should take the lead to adopt e-commerce and offer efficient path through electronic tender invitation for government procurement on the other hand.

Governments undertake large numbers of social, economic, cultural and service functions, and as “visible hands” in particular, they play a significant role in coordinating market economy and keeping markets from being out of order. At the age of e-commerce, the governmental supervision is definitely to change when enterprises apply e-commerce to produce and operate, banks realize the finance electronicization, and customers carry out online shopping.

B2G (business to government) imply exchange of information, services and products between business organizations and government agencies on-line.

- E-procurement services, in which businesses learn about the purchasing needs of agencies and provide services.
- A virtual workplace in which a business and a government agency could coordinate the work on a contracted project by collaborating on-line to coordinate on-line meetings, review plans and manage progress.
- Rental of on-line applications and databases designed especially for use by government agencies.

G2G (government to government) also known as e-government, transactions within countries linking local governments together and also international governments,
especially within the European Union, which is in the early stages of developing coordinated strategies to link up different national systems. In e-government, online virtual administration is built up to carry out online e-government by means of Internet, a fast, cheap and intense communication method. The establishment of e-government will boost its supervision and application functions and public service by making use of advanced electronic tools. To serve the public in a better way, government communicates with the public on line and listens to their vox populi. The polity steers the information exchange with public hearing the opinion and true feelings of the people, thus making polity serve for the community better.

G2B (government to business) involve exchange of information, services and products between government agencies and business organizations. Government sites now enable the exchange between government and business of:

- Information, guidance and advice for business on international trading, sources of funding and support, facilities.
- A database of laws, regulations and government policy for industry sectors.
- On-line application and submission of official forms (such as company and value added tax).
- On-line payment facilities.

This improves accuracy, increases speed and reduces costs, so businesses are given financial incentives to use electronic-form submission and payment facilities.

### 1.3 Supporting tools for e-Commerce

#### 1.3.1 Portals

“So-called portal websites are applications systems that lead to some comprehensive Internet information sources and provide relevant information services”, as the Wikipedia defines. At the beginning, the portal websites offered search engine and the network access service. Later, due to the increasingly competition in the market, the portal websites had no choice but to expand a variety of new services in the hope that Internet users would be attracted and maintained by the business of different kinds, which causes a wide variety of services to come up in different portal websites currently, making the network a “department store” or “network supermarket”. Recently, the portal websites mainly provide such services as news, search engine, online access, chat room, e-board, free mailbox, movies and music information, e-commerce, online community, online game, etc.

As the demand level theory proposed by Maslow defines, people have five needs: psychological needs, security needs, social needs, self-respect needs and needs of self-fulfillment, as is ordered from low to high by the intensity of needs.

Users’ needs of self-respect and self-fulfillment have not been fully exploited so far into portal world. There is neither geographically accessible (in terms of the end serving device) nor psychologically accessible (whether or not the websites are doing well has
nothing to do with users) between websites and users. The portal websites will develop in
the direction of personalized network portal from the unified network portal in the future.

One-stop unified service, all kinds of advertisement, and homogeneous news reflect the
current picture of portal websites.

By using the Web 2.0 technology, some companies has already released personal portal.
Users can subscribe to different news channels, weather forecast, finance and economic
information, etc. as long as they have a password and login in it. The combination of IM,
blog and RSS is also a wonderful creation, because possibly in the near future, netizens
will have their needs satisfied only by starting IM. It is when a real era of personal
network portals comes.

1.3.2 Client Intelligence

CRM has to rest on two solid foundations: reasonable organization structure and
reasonable information structure, after Jim Berkowitz (a well-known management
consultant). If the enterprises are using CRM with the motive of separate interest of each
department rather than the motive of adapting to philosophy, culture and strategy of
customer-centered commerce, CRM will suffer a lack of reasonable organization
foundation. This reasonable organization structure can substitute a more integrated and
shared work flow and information flow for the original collective process among
departments, thus the enterprises becomes a unified organization to effectively predict
customers’ needs, and value and simplify their work process.

On the other hand, under the competition pressure, competition for the limited clients will
turn more and more. Because the cost of obtaining a new client is far more than that of
maintaining an original client, it has been a key problem what measures the enterprises
must take to maintain the clients and improve the clients’ loyalty. The answer lies in the
record of dealing with the clients, that is, the clients database, but the enterprises do not
make effective use of it to discover valuable potential client-loyalty-related information.

Based on the research literature client intelligence is classified on three categories: the
first one is the research on function, contents and essence of client intelligence. One
approach to client intelligence is the creative use of client’s information, an integration of
concepts, methodology, process and software of decision-making and overall operation
capacity to help enterprises improve client’s relationship.
At the enterprise level, the foundation of client intelligence is guidance for enterprises to make decisions about clients. It includes not only the theories and measures by which enterprises analyze and treat clients, but also client value analysis from the perspective of clients and enterprises respectively. Through analysis on consumption behaviors, satisfaction, interest rate, etc., enterprises can reach the goal of scientific decision-making and rationality. In this we have:

- Enterprise strategy level
- Client development strategies
- Clients metrics – that are computed with different BI tools and we can have, Client Satisfaction, Client Loyalty, Client Classification, etc.

As a series of computation, devices or models, it intends to obtain high-quality data or information about the theme concerned, then involve in using analytical computation, devices or models automatically or by hand to help people analyze information, reach a conclusion, form a hypothesis and test a hypothesis.

Knowledge discovery it is also a series of computation, tools or models. It intends to transform data into information, then into knowledge through discovery, or directly transform data into knowledge.

At strategic level, it intends to apply information or knowledge to such aspects as improving decision-making, operational capacity, enterprises’ construction model, etc. The strategic level is a combination of a series of concepts, methods and processes for improving the enterprises’ capacity to make decisions by use of information from many sources and application experience and hypothesis. By obtaining, managing and analyzing information, it provides all the staffs the access to knowledge so as to improve the capacity of strategic decision-making and tactic decision-making.
1.3.3 Decision Support

The economic globalization and development of information technology eliminates many barriers and the key for management is decision-making. The enterprises are facing more complex environment than ever, and it is much harder to create and maintain their competition barriers. The pressure from competition raises higher standard for the quality and speed of making decision.

Decision support system (DSS) can provide enterprises various decision-making information and solutions to many commercial problems, thus freeing managers from the burden of low-class handling and information analysis, while indulging them in the work that needs most wisdom and experience for decision-making. In this sense, the quality and efficiency in making decision are greatly improved.

Different people perceive DSS from different perspectives. In a broad sense, DSS can be regard as a terminology containing many things, in order to describe any computerized system of an organization. Based on the support system of computers, DSS is a decision-maker and administrator for handling semi-constructed problems.

An organization generally has one management information system used by senior managers, a DSS for market, finance and accounting, a MRP system for production and a few expert systems for maintenance and diagnosis.

After adopting DSS, enterprises can get such benefits as: higher quality of decision, improvement of communication, reduction in cost, improvement in perceived benefits have so much to do with the competition among enterprises, size of enterprises and friendliness of clients.

Due to these challenges faced by modern enterprise management, enterprises managers need urgently a computerized DSS. Although every enterprise differs in the real situation and demands of them, they have reasons in common for adopting computerized decision-making system.

- Speedy computation: Computers allow decision-makers to make a large amount of computation at a low cost (labor cost of high-level managers is extremely high). In many cases, timely decisions are crucial, such as for stock deal and marketing strategy etc.
- Overcoming the limits of handling and storage: Human intelligence is confined to the capability of information handling and storage. In addition, it is impossible for a person to recall accurately all the information in need.
- Perception limits: Personal ability of solving problems is limited when various knowledge and information are required. Though it is helpful to gather knowledge of different persons, it is difficult to coordinate and communicate between members of the work team. However, computer systems contribute to decrease the coordination and communication on the one hand, and facilitate people to have access to information and handle sea of information.
Cutting down expense: It cost much to gather a group of decision-makers, especially experts. Owing to the technology, the size of team and expense of travel can be decreased (team members in different places can communicate), and productivity of relevant people (e.g. analysts of financial affairs or law people) can be improved, which is necessary for decision-making. The improved productivity means the decreased cost.

Information supply: Managers can make decision by obtaining accurate, timely and latest information by means of computer technology. Audio and video data can be transferred across a long distance, whether the data are stored in different databases within the organization or even outside the organization. Necessary data can be inquired, stored and transferred rapidly and economically.

Quality supply: Alternative plans can be commented, risk analysis can rapidly be carried out, ideas of experts in different places can be gathered rapidly at a low cost and professional knowledge can directly be submitted via Internet. Moreover, decision-makers can make complex simulation, check all cases and make comments rapidly and at a low cost. All above mentioned will contribute to a better decision.

Contributing to business flow regrouping and staff assignment: It is difficult to make a decision because of the competitions not only in price, quality, time, product ordering but in services to customers. An organization must transform operation patterns, regroup business flow and structure, and authorize the staff to make innovation rapidly and frequently. Decision support system (e.g. expert system) enables the people who are lack of knowledge to work out a good idea, for which an organization can make good decision. The system can also be used in business flow regrouping, competitors’ activities analysis, products ordering and production flowing etc.

Enterprises can adopt various DSS according to their own situation. The following are the most significant applications:

- Sales support: The system offers high-level managers the support on the basis of gathered reports on the salesman, production and sales in different areas, different departments. These reports present lost business, redemptive business and new business. If necessary, extra periodical reports can be made to enable the manager to make comparison and trend analysis, which contributes to problem solving and opportunity grasping. DSS can be used to analyze and comment on products sales to figure out factors of success or failure, and to predict the potential profits and incomes on the basis of the data of the company.

- Analysis of customers and study of market: Due to the application of DSS, transaction data gathered can be analyzed by means of statistic tools. Accordingly, different consumption patterns and corresponding market strategies can be worked out so as to achieve the optimal profits. Market study consists of the following aspects: analyze growth pattern of every product by making use of prediction model so as to make an appropriate decision: to terminate or to expand a certain product; study brand and image of the enterprise so as to improve its reputation; analyze the customers’ satisfaction and study market size and potential size.
- Financial affairs analysis: Compare the expense daily, monthly, yearly, or on the basis of user-defined periods; check the trend of cash flow in the past and predict future cash requirements; make budget plans and circulation of cost of complicated projects; regroup financial affairs data of branches and form the accurate financial statements.

- Operational search and strategy plan: In terms of resources and time, work out the optimal schedule; work out everyday production plan; determine the establishment branches of large-scaled chain organizations, such as chain stores, service stations, and communication relay stations etc.; aid in working out large-sized investment plans and estimating the investment risks.

- Enterprises analysis: Critical Success Factors (CSF, for short) refer to the factors that must be taken into account to reach the goal of the organization. CSF, strategic or operational, is the focus of enterprise performance analysis, and derives from the following three factors: organizational factors, industrial factors and environmental factors. And KPI (Key Performance Index) is the metrical unit for enterprise performance analysis. The typical KPI is as the following table

<table>
<thead>
<tr>
<th>KPI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profitability</td>
<td>Profitability of every department, product and area; the comparison of departments, products and competitors</td>
</tr>
<tr>
<td>Financial affairs</td>
<td>Flow ratio, cash storage; balance sheet analysis; mercantile rate of return</td>
</tr>
<tr>
<td>Market</td>
<td>Market share, advertisement analysis, product analysis, weekly (daily) sales, consumers’ sales potential</td>
</tr>
<tr>
<td>Human resources</td>
<td>Personnel liquidity, work satisfaction</td>
</tr>
<tr>
<td>Plan</td>
<td>Sales increase/market share analysis</td>
</tr>
<tr>
<td>Economic analysis</td>
<td>Market trend, foreign trade and exchange rate, industry trend, labor cost trend</td>
</tr>
<tr>
<td>Consumers trend</td>
<td>the grade of consumers’ confidence, shopping habits</td>
</tr>
</tbody>
</table>

Table 1 Example of KPI

As described on previous document issued by this projects (Candea, 2010) a DSS comprises of the following typical components:

- Database management subsystem: subject-oriented data warehouse functioning as decision-making support, in which any data unit can be modified at any time. Data in data warehouse is generally taken from in-raw data (mainly from transaction processing system) and out-raw data (including industrial data, market survey data, census data and national economic data).

- Model management sub system: A software package including financial affairs statistics, operational search models and other quantitative models can offer people systematic analysis and appropriate software management. As for models in model database, they can be divided into strategic, political and operational ones.

- Knowledge management subsystem: Expert systems and other intelligence systems are required besides DSS to manage complex situation. Advanced DSS need to include so-called knowledge management components.
- User interface subsystem: The communication between users and DSS. In addition, intranet/Internet release should be included to realize information sharing within organizations.
- Users: Users can be regarded as one component of DSS. They mainly refer to the enterprise managers of different levels and business analysts.

### 1.4 Supporting Technologies for E-Commerce

E-commerce must be supported by corresponding technologies. General speaking, e-commerce supporting technologies can be divided into three categories:

- Information display technologies which include Web, HTML, XML and Java (or other modeling and software development languages) technologies.
- Information transmission technologies which consist of EDI, TCP/IP, WAP, WLAN and Bluetooth technologies.
- Information processing technologies which comprise some common used technology such as GPS, GIS, DSS, GDSS, IDSS.

On next chapters briefly are presented only aspects about Information processing technologies. Aspects about Information display and Information transmission are not subject of present work and are not detailed.

Relation among of the Information processing technologies - Global Positioning System, Geographical Information System, Decision Supporting System, Group Decision Support System and Intelligent Decision Supporting System - are presented in next figure:

![Figure 5 Relation amond Information processing technologies](image)

### 1.4.1 Global Positioning System (GPS)

GPS (Global Positioning System) was originally designed by the US Department of Defense, the aim of which was to pilot the aircrafts and vessels for the US forces. As the GPS satellite network was established completely in 1993, the scope of application has
been extended continuously. In 1994, the US government announced that it would 2 E-commerce Supporting Technologies provide the world with access to GPS free of charge in the next 10 years. On Feb 29 1996, the US government formally announced that GPS was opened only for military and civil applications; however, the positioning precision was reduced to restrict civil applications. On May 1st, the US president announced that was set to zero, which greatly promoted the development of GPS civil applications.

Is a popular saying in the GPS industry: “GPS is only restricted by people’s imagination”. Now the application areas of GPS include land transportation, marine transportation, civil aviation, communication, mapping, mining, agriculture, power system, medical application, scientific research and entertainment. The positioning precision has evolved from 10 meters to 1mm. GPS is a system composed of 24 satellites for global positioning and piloting, the GPS receiver at any place will receive the space signal from at least 4 satellites. Therefore, GPS has the powerful function of positioning.

The receiver interprets the positioning information from each satellite to determine the position, thus providing highly precise 3-dimensional positioning and piloting. The space part of GPS is composed of 24 satellites evenly distributed in six orbit planes: the ground supervision part, in charge of supervising the satellite and calculating satellite calendar, includes a principal controlling station, 3 upload stations and 5 monitoring stations. The client device of GPS is primarily composed of hardware and processing software. The user receives GPS satellite signal via client device, and obtains the information about location and speed and so on so as to finally implement GPS piloting and positioning.

1.4.2 Geographical Information System (GIS)

Geographical information means the relevant information to geographical distribution of the research subject, including quantity, quality, distribution characteristics, correlation of the subject and its environment. Geographical information is a kind of spatial information, and since the identification of locations is closely related to data, it is regional. Geographical information is also characterized by multi-dimensions, that is, the same location may possess an information structure covering several subjects and properties.

For instance, at a certain point, there is information concerning altitude, earth bearing strength, noise, pollution and transportation. Moreover, geographical information is evident characteristic of time sequence, namely, dynamic feature. This feature requires timely data collection and updating so that prediction of the future can be made on the basis of its temporal laws that are worked out according multi-phase data and information.

GIS is a spatial information system mainly concentrating on collecting, storing, managing, analyzing and describing information of part of or the whole surface (including the atmosphere) of the earth that is relevant with the spatial and geographical data.
GIS is an interdisciplinary subject that combines computer science and spatial data analysis. The areas it relates to include geodesy, photography, cartography, geology, remote sensing, and image analysis and so on. GIS is closely related with but differentiated from these disciplines.

### 1.4.3 Decision Supporting System (DSS)

Decision Supporting System (DSS), is an intelligent man-machine system that is based on management science, operational research, control system and behavioral science. It uses computer technology, simulation technology and information technology to solve semi-structured decision problems. DSS provides data, information and background for the decision maker to help them identify the problems and establish decision models; evaluates and selects various cases through analysis, comparison and judgment, providing necessary support for the best decision.

The concept of DSS has his inception in the 1970’s, and developed in the 1980’s because traditional MIS did not bring tremendous benefit to the enterprises while people require more advanced systems to support decision.

The concept structure of DSS contains of session system, control system, operating system, database system, model base system, and rule base system and the users. The simplest and most practical DSS logical structure (database, model base, rule base) is illustrated in Figure 6.

![Figure 6 Structures of DSS three bases](image)

The process of DSS can be described following the next steps:

- the user inputs the decision problem through the session system, which then passes the input problem to problem processing system;
- then the problem processing system begins to collect data and identify the problem according to the knowledge stored;
- if a problem occurs, the system will interact with the user via the session system until the problem is identified;
- then the system begins to search for the appropriate model to solve the problem,
- induces the feasibility of the solution,
- (And finally) renders the decision information to the user.

DSS technology includes:

1. Interfaces - of input and output and also acts as the platform of interaction.
2. Model management - the system will retrieve the existing basic models according to the problems raised by users. Therefore the model management part has to possess storage and dynamic modeling functions. Currently the implementation of model management is accomplished by model base system.
3. Knowledge management - managing the knowledge (rule and facts) of decision problem, including the retrieval, expression and management of knowledge.
4. Database - manages and stores decision-related data.
5. The system emphasizes interactive transactions. A decision needs repetitive and frequent interactions; the human factors such as personal preference, subjective judgment, ability and experience have profound influence on the decision result.

DSS is generally developed with the combination of objective-oriented method and prototype method, described as follows: first use prototype method to develop individual parts of the DSS, then assembles them to form development toolset and environment of DSS according to the general method of system generation.

The development of DSS usually aims at a specific problem, which can be divided into five phases: problem analysis, feasibility research, selection of development method, system development and decision supporting. The decision maker should be involved in the process for he is the director user and his need is the aim of the system.

The work of each phase is outlined as follows:

1. Problem analysis phase. This phase includes field investigation and analysis to identify the problem.
2. Feasibility research. Based on the analysis of the previous phase, the feasibility of the system development is studied in terms of technology, feasibility, effectiveness of solution, and economic and social benefits.
3. Determination of the development methods and strategies. This phase involves the determination of how the system development is organized and what tools, methods and approaches are employed.
4. System development phase. It involves the development of a DSS specific to the problem, including the establishment of the DSS structure, data model and evaluation standard.
5. Decision supporting phase. It means the actual operation phase after the system development is completed. It includes the result analysis, decision support and the data collection that reflects the validity of the system operation.
1.4.4 **Group Decision Supporting System (GDSS)**

Group Decision Support System (GDSS) means that multiple decision makers communicate with each other to find a feasible, conformed to and satisfying solution, but the final decision is made by a certain one, who also takes responsibility for the result.

GDSS is developed out of the DSS by increasing the number of participants and making the information source more extensive. It effectively avoids one-sidedness of decision and dogmatized behaviors.

The functions of GDSS include the following points:

1. The difference is eliminated by enforcing communication and the relation between participants is controlled and coordinated by restricting the unnecessary emotional interaction.
2. The status of participants and the justness of the conclusion are enhanced.
3. The implementation of the system is permanent or temporary.

The technical functions of GDSS include the following points:

1. Control over data exchange in the decision process.
2. Automatic selection of appropriate GDSS technology.
3. Computation and explanation of the feasible solutions.
4. If GDSS cannot reach agreement, the difference is discussed or the problem is redefined.

A typical GDSS structure is illustrated in Figure 7.

Figure 7 General structures of GDSS
GDSS is a new branch of the decision support area, an extension of DSS, which includes:

1. A Communication Base is added to facilitate the communications between the decision makers.
2. Model base is enhanced, providing voting, sorting, classified evaluation to fulfill the agreed decision.
3. The system can be self-prepared and coordinated before used, such as scheduling a meeting.
4. Necessary physical devices are extended.

The type of GDSS depends on the problem which will be decided and its environment to a certain degree, thus GDSS is generally classified into four types:

1. Decision room: it seems like a traditional meeting room, where the decision makers gather and take part in the decision making process through terminals. This process is restricted by time.
2. Local decision network: the participants of GDSS are not restricted by space. Once the LAN is equipped with public GDSS software and database is stored in the central processor, the participants can make communication between the central processor and the members or each other via the LAN.
3. Fax conference: it is intended for the group that is separated geographically but can be assembled when necessary.
4. Remote decision-making: it is intended primarily for those members who have to meet regularly but cannot meet each other physically. These dispersed decision makers keep constant communication with each other via remote decision stations.

1.4.5 **Intelligent Decision Supporting System (IDSS)**

Intelligent Decision Supporting System (IDSS) is a supporting system that combines Artificial Intelligence and DSS, and uses the technology of Expert System to enable DSS to sufficiently apply human knowledge so as to solve complex decision problems. With Expert System, the DSS can apply human knowledge more sufficiently to solve problems through logic reasoning.

The concept of IDSS was originally proposed by Bonczek in the 1980’s, the function of which was to deal with quantitative problems as well as qualitative problems. The core idea of IDSS was to combine AI and other relevant disciplines to make DSS more intelligent. In order to introduce AI to DSS, the expert system is combined with DSS, what’s more, inference machine and rule base are also introduced to DSS. In the decision process, some knowledge cannot be represented by data or model. The rule base introduced in IDSS can store the knowledge and provide important reference to the decision process.

IDSS has multiple kinds of information bases (as illustrated in Figure 8): text base (TB), database (DB), approach base (AB), model base (MB) and rule base (RB).
The text base stores numerous texts written in natural language; the database stores records of key fields; the model base stores various models that illustrate the relations of information; the rule base stores the rules. The process of information extraction from raw data to processed information is called “evolving link”.

The IDSS can be technically divided into three layers:

1. Application layer. It is directly oriented to the user of IDSS. In this layer, the decision maker can determine the state and impose restrictions on IDSS according to his need. And through the user interface he may input some information, which will be understood by DSS via information transformation. Through inference and calculation, the system will return the user the result through interface. The whole process is transparent to the user.

2. Control and coordination layer. It is oriented to the designer of IDSS, the fundamental element of which is the control and coordination modules of the databases. The system engineer establishes the connections between such modules via the standard interfaces.

3. Basic structure layer. It is oriented to programmers. The programmer realizes all bases through this layer, including the structure and communication mode of them, so as to accomplish internal management and external communication of bases.

IDSS has the following features:

1. It is based on mature technology. So it is easy to construct applicable system.
2. It sufficiently uses the information resources of all the layers.
3. It is based on rules, which enables users to use it easily.
4. It is characterized with strong modulization, which enables reuse and low cost.
5. It is flexible to combine system parts, which enables easy maintenance and powerful functions.
6. It is easily upgraded by adopting advanced supporting technology, such as AI.

The modules have to call the upper layer repetitively, which has lower efficiency than calling lower layer while IDSS is running. However, since IDSS is just used in important decisions, it is worthwhile to sacrifice the running efficiency for the efficiency of system maintenance.
Capitolul 2 Knowledge in E-Commerce

2.1 Ontologies

An ontology is a formal explicit description of concepts in a domain of discourse (classes, sometimes called concepts), properties of each concept describing various features and attributes of the concept (slots, sometimes called roles or properties), and restrictions on slots (facets, sometimes called role restrictions). Ontology together with a set of individual instances of classes constitutes a knowledge base. In reality, there is a fine line where the ontology ends and the knowledge base begins (Noy, N.F. and McGuinness, 2010).

Ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. The reasons to develop ontology maybe some of those:

- To share common understanding of the structure of information among people or software agents.
- To enable reuse of domain knowledge.
- To make domain assumptions explicit.
- To separate domain knowledge from the operational knowledge.
- To analyze domain knowledge (Noy, N.F. and McGuinness, 2010)

E-Commerce ontology is a useful knowledge tool which permits to monitor and manage the manufacturing resources more effectively, eventually figuring out the performances and the potential of a production line. Moreover, this knowledge applies in order to make decisions upon several issues related to a E-Commerce system as for example the more efficient operating of the processes or the selection of suitable resources for a specific task. Finally, a well-structured E-Commerce ontology is expected to provide information in order to make a production line working more efficiently.

Ontologies are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, and enterprise bookmarking and information architecture as a form of knowledge representation about the world or some part of it. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework. There have been created ontologies in numerous sectors such as biology, biomedical, education, electronic health, engineering design, finance, infrastructure, product and monetary.

A descriptive definition identifies the basic characteristics of ontology and provides its main requirements. Gruber’s and Borst’s ontology definitions have been merged and explained by Studer et al. (Fernández López, M., Gómez-Pérez, A, Sierra, J.P., Sierra, A.P. 1999)
Ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.

Figure 9 Class, Subclass and Attribute definition

- Concepts are taken in a broad sense. Concepts in the ontology are usually organized in taxonomies through which inheritance mechanisms can be applied.
- Relations represent a type of association between concepts of the domain. If the relation links two concepts, it is called binary relation.
- Instances are used to represent elements or individuals in ontology.
- Constants are numeric values that do not change during much time.
- Attributes describe properties of instances and of concepts. We can distinguish two types of attributes: instance and class attributes.
  - Instance attributes describe concept instances, where they take their values. These attributes are defined in a concept and inherited by its sub-concepts and instances.
  - Class attributes describe concepts and take their values in the concept where they are defined. Class attributes are neither inherited by the subclasses nor by the instances. An example is the attribute type of control of the concept company, which can be used to determine the type of control of a private company, a public company, and a shared control company.
- Formal axioms are logical expressions that are always true and are normally used to specify constraints in the ontology.
2.2 Ontologies Technologies

Based on the language structure ontology languages can be grouped into three categories, frame based, description logics based and first order logic based.

2.2.1 Frame based languages

A frame language is a meta-language. It applies the frame concept to the structuring of language properties. Frame languages are usually software languages.

Frame languages are rather focused on the recognition and description of objects and classes, and relations and interactions are considered as "secondary".

Frames are intended to help an Artificial Intelligence system recognize specific instances of patterns. Frames usually contain properties called attributes or slots. Slots may contain default values (subject to override by detecting a different value for an attribute), refer to other frames (component relationships) or contain methods for recognizing pattern instances. Frames are thus a machine-usuable formalization of concepts or schemata. In contrast, the object-oriented paradigm partitions an information domain into abstraction hierarchies (classes and subclasses) rather than partitioning into component hierarchies, and is used to implement any kind of information processing.

Like many other knowledge representation systems and languages, frames are an attempt to resemble the way human beings are storing knowledge. It seems like we are storing our knowledge in rather large chunks, and that different chunks are highly interconnected. In frame-based knowledge representations knowledge describing a particular concept is organized as a frame. The frame usually contains a name and a set of slots.

The slots describe the frame with attribute-value pairs <slotname value> or alternatively a triple containing frame name, slot name and value in some order. In many frame systems the slots are complex structures that have facets describing the properties of the
slot. The value of a slot may be a primitive such as a text string or an integer, or it may be another frame. Most systems allow multiple values for slots and some systems support procedural attachments. These attachments can be used to compute the slot value, or they can be triggers used to make consistency checking or updates of other slots. The triggers can be trigged by updates on slots.

In general Frame based languages are particularly useful when the application has the following requirements:

- Creating ontologies for domains in which the closed-world assumption is appropriate.
- The application focuses on data acquisition.
- The application domain requires constraints on slot values.
- When classes should be used as property values.

F logic and Open Knowledge Base Connectivity (OKBC) are frame based languages.

### 2.2.2 First order logic

First-order logic is a formal logical system used in mathematics, philosophy, linguistics, and computer science. It goes by many names, including: first-order predicate calculus, the lower predicate calculus, quantification theory, and predicate logic. First-order logic is distinguished from propositional logic by its use of quantifiers; each interpretation of first-order logic includes a domain of discourse over which the quantifiers range.

There are many deductive systems for first-order logic that are sound (only deriving correct results) and complete (able to derive any logically valid implication). Although the logical consequence relation is only semi decidable, much progress has been made in automated theorem proving in first-order logic. First-order logic also satisfies several metalogical theorems that make it amenable to analysis in proof theory, such as the Löwenheim–Skolem theorem and the compactness theorem.

First-order logic is of great importance to the foundations of mathematics, where it has become the standard formal logic for axiomatic systems. It has sufficient expressive power to formalize two important mathematical theories: Zermelo–Fraenkel set theory (ZF) and first-order Peano arithmetic. However, no axiom system in first order logic is strong enough too fully (categorically) describe infinite structures such as the natural numbers or the real line. Categorical axiom systems for these structures can be obtained in stronger logics such as second-order logic.

CycL and Knowledge Interchange Format (KIF) are examples of languages that support expressions in first-order logic, and, in particular, allow general predicates.
2.2.3 Description logics

Description logics (DL) are a family of formal knowledge representation languages. They are more expressive than propositional logic but have more efficient decision problems than first-order predicate logic.

DLs are used in Artificial Intelligence for formal reasoning on the concepts of an application domain (known as terminological knowledge). They are of particular importance in providing a logical formalism for Ontologies and the Semantic Web. The most notable application outside information science is in bioinformatics where DLs assist in the codification of medical knowledge.

The following requirements of applications may make a Description Logics language a preferred choice:

- Creating robust terminologies in which classes are defined.
- Need for DL reasoning to ensure logical consistency of ontologies.
- Controlled terminologies are published on the Semantic Web and accessed by other applications.
- Applications in which classification is a paradigm for reasoning.

Examples of description logics based languages include DAML+OIL, KL-ONE, RACER and OWL-DL.

2.2.4 The OWL language

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. They are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. This language derives from and supersedes DAML+OIL. It covers most of DAML+OIL features and renames most of its primitives. Like DAML+OIL, OWL is built upon Resource Description Framework (RDF) and therefore some RDF primitives are reused by OWL. OWL ontologies are written either in XML or with the triples notation for RDF. The OWL specification includes the definition of three variants of OWL, with different levels of expressiveness. These are OWL Lite, OWL DL and OWL Full.

2.2.4.1 OWL Lite

OWL Lite was originally intended to support those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing quick migration path for systems utilizing thesauri and other taxonomies. In practice, however, most of the expressiveness constraints placed on OWL Lite amount to little more than syntactic inconveniences: most of the constructs available in OWL DL can be built using complex combinations of OWL Lite features. Development of OWL Lite
tools has thus proven almost as difficult as development of tools for OWL DL, and OWL Lite is not widely used.

### 2.2.4.2 OWL DL

OWL DL was designed to provide the maximum expressiveness possible while retaining computational completeness (either φ or ¬φ belong), decidability (there is an effective procedure to determine whether φ is derivable or not), and the availability of practical reasoning algorithms. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, number restrictions may not be placed upon properties which are declared to be transitive). OWL DL is so named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL.

### 2.2.4.3 OWL Full

OWL Full is based on a different semantics from OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL DL. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for OWL Full.

### 2.2.4.4 Choosing the Sub-Language to use

Although many factors come into deciding the appropriate sub-language to use, there are some simple rules.

- The choice between OWL-Lite and OWL-DL may be based upon whether the simple constructs of OWL-Lite are sufficient or not.
- The choice between OWL-DL and OWL-Full may be based upon whether it is important to be able to carry out automated reasoning on the ontology or whether it is important to be able to use highly expressive and powerful modeling facilities such as meta-classes (classes of classes).

The Protégé-OWL plug-in does not make the distinction between editing OWL-Lite and OWL-DL ontologies. It does however offer the option to constrain the ontology being edited to OWL-DL, or allow the expressiveness of OWL-Full. During the development of the VFF ontology the ontology language was constrained to OWL-DL to take advantage of the automated reasoning capabilities.

### 2.2.5 Knowledge representation

An ontology in OWL starts with the declaration of the RDF root node. In this node we must include the namespaces for the RDF, RDFS and OWL KR ontologies. If XML Schema data types are used, it may be helpful to include a namespace for XML Schema,
which is usually prefixed as xsd (and which points to the newest URL of XML Schema, as in RDF(S)).

```xml
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#">
    As in other languages, the order in which definitions appear in OWL ontologies is not relevant since it is based in RDF(S). OWL ontologies usually define the header of the ontology first and then the ontology terms. The header of an OWL ontology may include: the ontology documentation (rdfs:comment); the ontology version (owl:versionInfo); and the imported ontologies (owl:imports). The ontology header may also include information about version control with the primitives owl:backwardCompatibleWith, owl:incompatibleWith, and owl:priorVersion. Inside the ontology we can also find definitions of deprecated classes and properties with the owl:DeprecatedClass and owl:DeprecatedProperty primitives.

    <owl:Ontology rdf:about="">
        <owl:versionInfo>1.0</owl:versionInfo>
        <rdfs:comment>Sample ontology for travel agencies</rdfs:comment>
        <owl:imports rdf:resource="http://delicias.dia.fi.upm.es/owl/units"/>
        <owl:imports rdf:resource="http://www.w3.org/2001/XMLSchema"/>
    </owl:Ontology>

    Concepts are known as classes in OWL and are created with the primitive owl:Class. Besides its name, a class may also contain its documentation (with rdfs:comment) and any number of expressions with the following list of primitives:

    - rdfs:subClassOf contains class expressions. It allows defining the superclass of the class.
    - owl:disjointWith asserts that the class cannot share instances with the class expression in this primitive.
    - owl:equivalentClass also contains class expressions. This primitive defines necessary and sufficient conditions for the class (i.e., it is used for redefining concepts already defined).
    - owl:oneOf defines a class by enumerating exhaustively all its instances. This is a way to define a class extensionally.
```
- owl:intersectionOf, owl:unionOf and owl:complementOf define a class expression as a conjunction, a disjunction, or a negation of other class expressions respectively.

The first two primitives (rdfs:subClassOf and owl:disjointWith) define necessary conditions for the class (they can be used in the definition of primitive concepts), while the rest of primitives define necessary and sufficient conditions for the class (that is, they are used to define defined concepts). In OWL Lite the only primitives that can be used are rdfs:subClassOf, owl:equivalentClass, and owl:intersectionOf. In all the cases they can only be used with class identifiers and property restrictions. According to the DL terminology, OWL is a SHOIN language. This means that class expressions can be built with the following constructors:

- Conjunction (owl:intersectionOf), disjunction (owl:unionOf), and negation (owl:complementOf) of class expressions.
- Collections of individuals (owl:oneOf).
- Property restrictions (owl:Restriction). They contain a reference to the property to which the restriction is applied with the primitive owl:onProperty and an element for expressing the restriction. The following restrictions can be applied to properties: value restriction (owl:allValuesFrom), existential restriction (owl:someValuesFrom), role fillers (owl:hasValue), and number restriction (owl:cardinality, owl:maxCardinality, owl:minCardinality).
- Besides, role expressions can express inverse roles (owl:inverseOf) and role hierarchies (rdfs:subPropertyOf).

In OWL Lite class expressions can only contain class names and property restrictions. The primitive owl:hasValue cannot be used in property restrictions. Besides, the primitives owl:allValuesFrom and owl:someValuesFrom only contain class identifiers or named datatypes, and cardinality restrictions only take the values 0 or 1. OWL DL does not impose any of these restrictions.

Disjoint knowledge can be expressed with the owl:disjointWith primitive. One individual can not belong to two disjoint classes.

Concept attributes must be defined as properties in the ontology. There are two types of properties: owl:ObjectProperty, whose range is a class, and owl:DatatypeProperty, whose range is a datatype. To define a property we may explicit its domain and range with the primitives rdfs:domain and rdfs:range. The primitive rdfs:range can refer to a class expression in OWL DL and only to a class identifier or to a named datatype in OWL Lite. We can state that a property is a subproperty of other properties with the primitive rdfs:subPropertyOf. Finally, we can express equivalence between properties with owl:equivalentProperty, and inverse properties with owl:inverseOf.

Binary relations are defined with the primitive owl:ObjectProperty. The global domain and range of a relation can be explicitly defined with the rdfs:domain and rdfs:range
primitives, which can contain any class expression (except for OWL Lite, in which they can only contain class identifiers, or named datatypes for rdfs:range). However it is not compulsory to define explicitly the global domain and range of a property. Instead, these restrictions can be defined locally inside class definitions with property restrictions, as presented in some of the concept definitions above.

In OWL we can define property hierarchies (with the rdfs:subPropertyOf primitive), we can state equivalences between properties (with the owl:equivalentProperty primitive), and we can assert the inverse of a property (with the owl:inverseOf primitive). In addition to using owl:ObjectProperty to define binary relations, we can provide additional logical information about it with the following primitives:

- owl:TransitiveProperty. This primitive states that the relation is transitive. For example if the property P relates individual a to individual b, and also individual b to individual c, then we can infer that individual a is related to individual c via property P.
- owl:SymmetricProperty. This primitive states that the relation is symmetric. For example if the property relates individual a to individual b then individual b is also related to individual a via property P.

We can also state global cardinality restrictions on all kinds of properties (either object properties or data type properties):

- owl:FunctionalProperty. This primitive states that the relation or attribute has only one value when applied to an instance of a concept of this domain. This primitive can be used for object properties (binary relations) and for datatype properties (concept attributes).
- owl:InverseFunctionalProperty. This primitive declares that the property is unambiguous, that is, for each instance of its range there is, at most, one instance of its domain that can take it. Consequently, this primitive can be used only with object properties. However, in OWL Full it can also be used with datatype properties.

Concerning higher parity relations, they must be defined as concepts in OWL.

Functions are not components of the OWL knowledge model, though binary functions can be represented with the owl:FunctionalProperty primitive previously discussed. Higher parity functions cannot be represented in this language.

Formal axioms are also not components of the OWL knowledge model.

Finally, instances are defined using only RDF vocabulary. In OWL, we must use the attribute rdf:datatype to express the value type of data type properties.
2.2.6 Reasoning mechanisms

The model-theoretic semantics of OWL is described by Patel-Schneider and colleagues (2003). This semantics is described in two different ways: as an extension of the RDF(S) model theory and as a direct model-theoretic semantics of OWL. Both of them have the same semantic consequences on OWL ontologies, and they are based on the DAML+OIL model-theoretic semantics, taking into account the differences between both languages. Like DAML+OIL, OWL allows including any additional statements (RDF triples) in its ontologies apart from those explicitly defined in the language. However, OWL is silent on the semantic consequences (or lack thereof) of such additional triples. A set of test cases has been defined by Carroll and De Roo (2003) including entailment tests, non-entailment tests, consistency tests, inconsistency tests, etc. They illustrate the correct usage of the OWL and the formal meaning of its constructs. Due to its similarities with OIL and DAML+OIL, inference engines used for these languages (FaCT, RACER, TRIPLE, etc.) can be easily adapted for reasoning with it. A reasoning engine already available is Euler38. As with other languages, these inference engines will permit performing automatic classifications of OWL ontology concepts, and detecting inconsistencies in OWL concept taxonomies. Furthermore, we can say that multiple inheritance is allowed in OWL ontologies (as we have discussed when describing how to create class expressions). In the semantics of OWL, however, there is no explanation on how conflicts in multiple inheritances can be solved. Constraint checking can be performed on the values of properties and their cardinalities. OWL assumes monotonic reasoning, even if class definitions or property definitions are split up in different Web resources. This means that facts and entailments declared explicitly or obtained with inference engines can only be added, never deleted, and that new information cannot negate previous information. As with DAML+OIL, many tools will be available for authoring OWL ontologies; tools capable of editing RDF(S) ontologies can also be used for developing OWL ontologies provided that the ontology developer uses the OWL KR primitives. In addition, RDF(S) query engines, storage systems, and parsers can be employed to manage OWL ontologies since they can be serialized in RDF(S).

2.3 Ontologies Use Cases

Three main categories of ontology applications can be identify: 1) neutral authoring, 2) common access to information, and 3) indexing for search

There is a common thread that binds these different communities: the need to overcome barriers created by disparate vocabularies, approaches, representations, and tools in their respective contexts. There is a need to share meaning of terms in a given domain. Achieving a shared understanding is accomplished by agreeing on an appropriate way to conceptualize the domain, and then to make it explicit in some language. Ontology as
result can be applied in a wide variety of contexts for various purposes. The following is adapted from Uschold, 1998.

- Communication between people. Here, an unambiguous but informal ontology may be sufficient.

- Inter-Operability among computer systems achieved by translating between different modeling methods, paradigms, languages and software tools; here, the ontology is used as an interchange format.

- Systems Engineering Benefits:
  - Re-Usability: the ontology is the basis for a formal encoding of the important entities, attributes, processes and their inter-relationships in the domain of interest. This formal representation may be (or become so by automatic translation) a re-usable and/or shared component in a software system.
  - Search: Ontology may be used as meta-data serving as an index into a repository of information.
  - Reliability: A formal representation also makes possible the automation of consistency checking resulting in more reliable software.
  - Specification: the ontology can assist the process of identifying requirements and defining a specification for an IT system (knowledge based or otherwise).
  - Maintenance: use of ontologies in system development, or as part of an end application, can render maintenance easier in a number of ways. Systems which are built using explicit ontologies serve to improve documentation of the software which reduces maintenance costs. Maintenance is also an important benefit if ontology is used as a neutral authoring language with multiple target languages - it only has to be maintained in one place.
  - Knowledge Acquisition: speed and reliability may be increased by using an existing ontology as the starting point and basis for guiding knowledge acquisition when building knowledge-based systems

On each exploitation scenario actors are involved and each actor represents a role that a person or application may play. A person or application may play more than one role in a scenario. Actors may play either a primary or a secondary role in a scenario.
The follow list describes the actors:

- Ontology Author: (OA) the role of the author of ontology. This role is usually played by a person.

- (Operational) Data Author: (DA) the role of the author of operational data in the language which uses and/or is defined in terms of the vocabulary of the ontology.

- Application Developer: (AD) the role of the developer of the Application.

- Application User: (AU) the role of the user of the Application.

- Knowledge Worker: (KW) the role of the person who makes use of the knowledge.

In next sub-chapters different use cases are described; a detailed description can be found on, Mike Uschold, Robert Jasper (1999)

2.3.1 Use Case – Neutral Authoring

An information artifact is authored in a single language, and is converted into a different form for use in multiple target systems. Benefits of this approach include knowledge reuse, improved maintainability and long term knowledge retention.

Use cases on neutral authoring are covering different aspects.

- the authored artifact may be an ontology, or operational data.

- the process of converting the artifact to a different form varies. It may be direct language to language translation, manual or automated, in which case the translation process may exploit both the syntax and/or semantics of represented concepts. Alternatively, the conversion may best be viewed as design, whereby the ontology is used by the developer as a requirements specification for the target applications. This does not result in a different explicit representation of the ontology, but rather the ontology is implicit in the application. In this case, there is no direct language to language translation. (R. Waldinger 1996 and K. Williamson 1997)
Benefit of authoring neutral ontologies is decreased cost of reuse and increased maintainability of knowledge. To accomplish this, the actors develop an ontology, which they can convert into multiple operational target systems. Supporting technologies include unidirectional ontology translators and software development processes (e.g., KADS). The principle actors are the ontology author and application user. In this scenario, the ontology author creates an ontology, which they convert into an operational target system (e.g., a knowledge base). Application users then interact with an operational system to perform their desired tasks.

Below are some practical examples of neutral authoring use case:

- An ontology author creates an ontology (e.g., for titanium alloys) in an ontology authoring language (e.g., Ontolingua). An application developer translates the ontology into Loom syntax (possibly assisted by automatic translation tools). An application developer directly imports this translated ontology into Loom and it becomes part of the end application, which may contain additional information in its knowledge base. An application user interacts with the final system to answer questions about titanium alloys. This ontology can be reused if another application developer wishes to make use of it in another language, e.g., Prolog. In that case, they will have to translate the ontology into Prolog and proceed as per the Loom example. Note that in this authoring scenario, only one-way translation is required. This contrasts with the case described in the next section, where two way translations is required when ontology is used as an interchange format. This example illustrates how to achieve knowledge reuse by virtue of the fact that ontology authored in a single language can be used in multiple applications.

- An ontology author creates an ontology using the conceptual modeling language (CML) from the KADS methodology. The application developer uses this ontology as part of the requirements specification when developing the target KBS (e.g., for diagnosing faults). An application user then interacts with the KBS to solve their tasks. In this example, maintainability is improved because there is
an explicit representation of the ontology that the software is based on. Reuse is achieved if the ontology is used for different applications in the same domain.

- Automated software synthesis into multiple target languages (e.g., using R. Waldinger 1996 and K. Williamson 1997) is a generalization of the neutral authoring language scenario. Application developers play a key role in both development of the ontology and problem statement specification. Typically, the developers semi-automatically refine the specification and ontology into an operational target application.

![Figure 12: Authoring Operational Data](image)

**2.3.2 Use Case – Common Access to Information**

The basic idea of this approach is to use ontologies to enable multiple target applications (or humans) to have access to heterogeneous sources of information which is otherwise unintelligible. Benefits of this approach include interoperability, and knowledge reuse. The scenarios in this category differ in a number of ways. First, the direct consumers of the information may be humans or computer applications. Second, the information artifact may play the role of an ontology, or operational data; non-computational (e.g., product data) or computational (e.g., services). Another important distinction is whether the target applications agree on the same shared ontology or whether each has its own local ontology. In the former case, the information is made intelligible via translators, and in the latter case, via ontology mapping rules. Finally, access to the information may be via sharing or exchange.

![Figure 13: Human Communication](image)
A major benefit of ontology development is to promote common understanding among knowledge workers. To accomplish this, the authors develop a common shared ontology, which other knowledge workers reference in their work. Non-computational skills such as library classification are valuable in building such ontologies, which commonly take the form of glossaries. Supporting technologies include ontology editors and browsers. The principle actors are the ontology authors and knowledge workers. The information being shared is ontology. In this scenario, the ontology authors create an ontology which knowledge workers reference in their work.

Within this use can we can find some interesting application such:

2.3.3 Data Access via Shared Ontology

The motivation behind data access via a shared ontology is reducing the cost of multiple applications having common access to data. This may in turn, facilitate interoperability. This is accomplished through developers agreeing on a shared ontology, which defines a common language for exchanging or sharing operational data. Supporting technologies include translators, parser generators and printers. The principle actors are ontology authors and application developers. In this scenario, an ontology author creates an ontology, which different application developers agree to use. This defines an interchange format for which two-way translation is required between it and the application formats. Each pair of translators, for a given application, in effect, defines an application interface that can be used to read and write data. Often the translators are manually created, in other cases, explicit readers and writers can be automatically created using parser generators and printers (see variation below). Inter-operation between the multiple applications is made possible by allowing them to access the same information.

Figure 14 Data Access via Shared Ontology
2.3.4 Data Access via Mapped Ontologies

There is no explicit shared ontology; instead, mapping rules are used to define what a term in one ontology means in ontology. A mediator uses these rules at runtime so that applications can access each other’s data. This approach has the advantage of not requiring the application developers to explicitly agree on a shared ontology. Supporting technologies include parser generators, printers, and mediators. The principle actors are ontology authors and application developers. In this scenario, each application wishing to exchange data has its own local ontology. Application developers cooperate to create a shared mapping that relates terms in different ontologies. This mapping is used to generate a mediator, which maps operational data expressed in the terminology of ontology into operational data expressed the other ontology.

![Diagram of Data Access via Mapped Ontology]

Figure 15 Data Access via Mapped Ontology

2.3.5 Shared Services

Developers achieve this by agreeing on a shared ontology, which defines interfaces in multiple target languages. This is very similar to data access via shared ontologies, except for the focus of what is being shared. Supporting technologies include interface generators and marshaling routines. The principle actors are ontology authors and application developers.

In this scenario, an ontology author creates an ontology, which different application developers agree to use. Parser generators and printers are used to generate application interface definitions that each application uses to read and write data.

For example: an ontology author uses a language such as IDL or UML to create ontology for objects in a domain of discourse (e.g., product data management). The ontology is used to generate interface code for the client and server (e.g., using CORBA). Client applications can then interface with services on the server regardless of location, operating system, or location.
2.4 Ontologies on E-Commerce

In last years the complexity of product description in the Semantic Web have been investigated as research but one proposal - GoodRelations ontology - that covers the representational needs of typical e-commerce scenarios in the commodity segment. This ontology has been developed as part of the myOntology and ebSemantics initiatives, in which the infrastructure for Semantic Web-based e-commerce for Austria is being developed.

On next sections the GoodRelation ontology is shortly presented as described by author, Martin Hepp.

2.4.1 Scenarios and Questions that GoodRelations ontology is addressing

Scenario 1: A Web resource represents an entity that, in general, offers items of a particular kind of good for sale, either to wholesalers or to end users, or both; they might offer concrete, identifiable instances or it may be that it is only said that such instances exist.

Scenario 2: A Web resource describes the make and model of a commodity and its specifications. Such Web resources are usually within the domain name space of the respective manufacturer. There may exist actual instances of this make and model, which by default inherit their properties from the specifications of the make and model, but also have additional properties of their own (e.g. the date of production, the serial number).

Scenario 3: A Web resource represents (1) the description of a particular range of products, determined either by product classes or makes and models, and property ranges, and (2) a concrete offer to rent out unidentified instances thereof. Such Web resources are usually within the domain name space of rental agencies or Web pages of local dealers.
Scenario 4: A particular Web resource represents (1) the description of a particular range of products, determined either by product classes, makes and models, or property ranges, and (2) a concrete offer to provide a certain type of service for this range of products (e.g. maintenance, repair, or disposal). Such Web resources are usually within the domain name space of local dealers.

To specify the scope and purpose of the GoodRelations ontology the competency questions are used, that is a standard technique in ontology engineering methodologies (Uschold, Grüninger, 1996). All this questions where part of interviews with developers of recommender systems, operators of Web shops, and other domain experts.

CQ1: Which retrievable Web Resources describe an offer?

- {to sell | to provide the service of | to repair | to maintain | to lease out | to dispose}
- {a concrete individual | some unknown individuals} of
- a {given good | given service | spare part for a given good | consumables and supplies for a given good} described by a {type of good | specific make and model}
- that meet certain requirements on {properties | intervals for properties}
- for which the offering party accepts a given method of payment and
  - provides a certain method of delivery
  - to {consumers | retailers}
  - in a given {country | region}?

CQ2: For which time frame is the offer valid?

CQ3: Which types of customers are eligible?

CQ4: Which are the eligible customer regions?

CQ5: Which shipping / delivery methods are available?

CQ6: Which methods of payment are accepted?
CQ7: For any such offer, what is the price and currency for a given quantity, delivery region, and type of customer, per unit of measurement? Does the price include VAT and sales taxes?

CQ8: What is the shipping charge and currency for a given delivery method to a given region? Does it include VAT?

CQ9: What is the payment charge and currency for a given payment method? Does it include VAT?

CQ10: What is the mail address and which are the contact details of the offering business entity?

CQ11: Which are the locations from which the product or service is being provided, what are the contact details and opening hours of each location?

CQ12: What is the scope and duration of the warranty promise or warranty promises included in the offer?

CQ13: Which offerings on the Web refer to {spare parts | consumables or supplies} for a given {type of good | make and model}

2.4.2 Domain concepts

The most important conceptual elements of the domain are described in this section. Bold characters are used when introducing and defining a conceptual element, and underlining to refer to the particular definition of a word or group of words defined elsewhere.

A visualization of GoodRelations domain capture in the form of an entity relationship diagram is shown in next figure.

Web Resource: A retrievable Web resource that contains information related to a business entity, an offering, a product model, or similar. A Web Resource combines information about multiple conceptual entities, e.g. about the offering party and the types of products. In the GoodRelations ontology, is assumed that Web Resources to be instances of rdfs:Resource and use the rdfs:seeAlso property for linking the actual conceptual entities (business entities, offerings, product instances, etc.) to Web Resources that contain a human-readable description. This is why there is no class “Web Resource” in the GoodRelations ontology.
Figure 17 GoodRelations Domain Capture
**Business Entity**: A legal agent making a particular offering. This can be a legal body or a person. A Business Entity has at least a primary mailing address and contact details. For this, typical address standards (vCard) and location data can be attached. (Example: Siemens Austria AG, Volkswagen Ltd., Peter Miller's Cell phone Shop).

**Offering**: The public, not necessarily binding, not necessarily exclusive, announcement by a Business Entity to provide a certain Business Function for a certain Product or Service Instance to a particular target audience. An Offering is specified by the type of product or service or bundle it refers to, what Business Function is being offered (sales, rental,…), and a set of commercial properties. It can either refer to a clearly specified instance (Product or Service Instance) or to a set of anonymous instances of a given type (existentially quantified Product or Service Instances, see below). An offering may be constrained in terms of the eligible type of business partner, countries, quantities, and other commercial properties. The definition of the commercial properties, the type of product offered, and the business function are explained in the following sections in more detail. (Example: Peter Miller offers to repair TV sets made by Siemens, Volkswagen Innsbruck sells a particular instance of a Volkswagen Golf at $10,000.).

**Business Function**: The type of activity or access offered by the Business Entity on the Product or Services though the Offering. The idea of standardizing business functions was first put to practice by the UNSPSC Business Functions Identifiers [9]. The most important functions are “sell”, “lease out”, “maintain”, “repair”, “provide service”, “dispose”, and “buy”. (Example: A particular offering made by Miller Rentals Ltd. says that they (1) sell Volkswagen Golf convertibles, (2) lease out a particular Ford pick-up truck, and (3) dispose car wrecks of any make and model.)

**Product or Service: Instance, Model, and Class**: In the products and services domain, we find multiple types of conceptual entities when it comes to describing what is being offered: First, actual products, like for example my cell phone or a concrete TV set. Second, certain product makes and models, e.g. the cell phone make and model Sony 1234 or the car model Ford T. There usually exist actual products that are of the respective make and model, but they all have an identity of their own. In particular, they differ in several properties. Third, classes of actual products that are similar in function or nature, like for example the class “cell phone” which subsumes all actual cell phones.

**Property of a Product or Service**: Products usually have certain characteristic features, often physical or chemical attributes or such related to the context of usage. A Product or Service Property is the type of a characteristic feature of an actual product or service instance. The value may be a quantitative or a qualitative value. In the former case, a Product or Service Property is a ternary relation between a Product or Services Instance, a Unit of Measurement, and this value. In the latter case, it is a
binary relation between a Product or Services Instance and this value. A quantitative value is a numerical interval that represents the range of a certain quantitative feature in terms of the lower and upper bounds for one particular. It is to be interpreted in combination with the respective Unit Of Measurement. Most quantitative values are intervals even if they are in practice often treated as a single point. A qualitative value is an entity that represents the state of a certain qualitative feature and can be either a literal value or an enumerative value. Products or Services Classes may be defined by constraints on Product or Services Properties. Product or Services Models may provide default values for Product or Services Properties.

**Unit of Measurement**: The point of reference and the scale for quantitative values. Those can refer to the technical/physical (e.g. weight) or commercial perspectives (e.g. sales units). Work on standardizing Units of Measurement has been ongoing in various standardization bodies, but there is still no single, widely agreed standard in place.

**Commercial Property of the Offering**: An offering is usually more specific than just stating the general availability of a given type of product or service. In many cases, it is desirable to specify additional commercial aspects, first of all the price.

**Relationships between Multiple Product Models or Product Instances**: In the commodities sector, it is often valuable to express several relationships between Product Instances and/or Product Models. GoodRelations defines “isConsumableFor”, “isSimilarTo”, and “isAccessoryOrSparePartFor” as the most common ones.

Specification of additional commercial aspects is needed than just the fact that a certain good is being offered. For example, list prices, quantities, eligible regions, and eligible types of business partners as relevant details. In the following, relevant elements that are dealing with these details in GoodRelations ontology are described.

**Price Specification**: The price asked for a given Offering by the respective Business Entity. An Offering may be linked to multiple Price Specifications that specify alternative prices for non-overlapping sets of conditions (e.g. quantities or sales regions). A Price Specification is characterized by (1) the lower and upper limits and the Unit of Measurement of the eligible quantity, (2) by a monetary amount per unit of the Product or Service Instance in the given Unit of Measurement specified as a literal value of type float in combination with a currency, and (3) whether this prices includes local sales taxes, namely VAT. (Example: The price, including VAT, for 1 kg of a given material is 5 Euros per kg for 0 – 5 kg and 4 Euros for quantities above 5 kg). GoodRelations ontology not model the full detail of local taxes.
Variants of Price Specification are Delivery Charge Specification (a conceptual entity that specifies the additional costs asked for delivery of a given Offering using a particular Delivery Method by the respective Business Entity) and Payment Charge Specification (a conceptual entity that specifies the additional costs asked for settling the payment after accepting a given Offering using a particular Payment Method).

Validity: The time interval that specifies the period during which the offer is valid.

Currency: The Unit of Measurement for monetary values. For currencies, we suggest using the well-established ISO 4217 standard, which is the recommended encoding for currencies in international payment transactions.

Warranty Promise and Warranty Scope: The duration and scope of services that will be provided to a customer free of charge in case of a defect or malfunction.

Payment Method: A standardized procedure for transferring the monetary amount for a purchase.

Delivery Method: A standardized procedure for transferring the Product or Service Instance to the destination of fulfillment chosen by the customer.

Business Entity Type: The legal form, the size, the main line of business, the position in the value chain, or any combination thereof, of a Business Entity. From the ontological point of view, Business Entity Types are mostly roles that a Business Entity has in the market. Business Entity Types are important for specifying eligible customers, since Offerings are often meant only for Business Entities of a certain size, legal structure, or role in the value chain.

Country or Region: Countries or Regions are geographical or geopolitical areas. In GoodRelations, they are used for specifying the areas for which the Offering is valid.

Location of Sales or Service Provisioning: A location from which the specified Business Function on the particular Product or Service Instance is being offered by the Business Entity.

2.4.3 URIs for E-Commerce on the Semantic Web

When using GoodRelations, unique identifiers are needed not only for the current Web resources, like corporate Web pages or pages in an e-shopping system, but for all conceptual elements as described in the domain capture. For example, are needed unique identifiers for (1) business entities, (2) makes and models, (3) offerings, and (maybe) (4) available product instances. It is tempting to assume that can simply take the URIs of existing Web resources for those purposes, but this is not advisable since
the available Web resources often tangle multiple conceptual entities – the Web page
describing a certain offer has one single URI, but it may contain the description of a
product make and model, an offering, a price specification, a warranty promise, all at
the same time – and, in the case of a small shop, even the business entity itself. When
move on to the Semantic Web, unique identifiers are needed for every single entity of
interest. In a nutshell, can be used the URIs of existing Web resources only as the
identifiers for the Web resources themselves. For all other conceptual entities, have
been introduced new identifiers, unless such have already been defined for the
Semantic Web elsewhere. For that, there exist three approaches.

**Using blank RDF nodes**: however, for all entities that may at any later point be
linked to other entities this is not recommended, since it is then not possible to refer
externally to such a node, which means that data from multiple sources cannot be
merged easily.

**Creating completely new URIs for all significant entities**: when doing so, one must
have authority to define the meaning of the respective URI. For URIs in the HTTP
scheme, this requires that one is the owner of the respective domain name space.

**Creating hash URIs derived from the URI of the original Web resource**: In RDF,
adding fragment identifiers to a URI creates a new identifier. It must be noted that
also in this case, one must have authority to introduce new URIs in the respective
domain space.

GoodRelations is agnostic of which approach is taken and makes no assumption on
who defines the URI for a respective conceptual element; it is just necessary that
distinct identifiers for distinct entities exist. Blank nodes are acceptable for entities
that will never be linked to outside data (e.g. workarounds for n-ary relations). The
URI of the current, retrievable Web resource should become the URI of the
conceptual entity Web Resource, and other conceptual entities, namely Offering,
should point to this via rdfs:seeAlso.

### 2.5 Architectures for Integrating CBR-Systems with eCommerce

Today, a company’s product information is already stored in conventional databases.
One way to make this information accessible to an Internet user is to prompt him to
fill in a search form via a web-browser. The form entries are translated into a query
which is interpreted by the database system. The results are then shown in the
browser. Problems with this approach arise when either no or too many products are
returned. In the first case, the user does not know how to change the query in order to
find a similar product; in the second case, no support is given to refine the query,
reducing the number of results to the relevant products. The reason for these problems
is the exact matching performed by conventional database systems for given queries. To overcome this limitation, similarity based search techniques known from case-based reasoning (CBR) approaches have been proposed for e-commerce applications (Vollrath, Wilke, Bergmann 1999). The basic idea of Case-Based Reasoning is to solve new problems by comparing them to similar old problems that already have been solved in the past. When a new problem has to be solved, the CBR-system (knowledge system) searches for the most similar old problem. So, on one hand we have existing e-commerce systems, on the other hand we have CBR-functionality to overcome the shortcomings mentioned above.

In next figure the main steps of electronic sales process are described and the capabilities of existing intelligent CBR sales support applications are colored red. The future potentials of CBR sales supporting systems are colored blue.

![Figure 18 The electronic sales process](image)

What happened when a customer are looking for a product? Only a customer that knows the product part number is assured of finding the desired information. If the customer is not looking for a specific part but for a product with particular qualities, the success of the search is much less certain. Unless very familiar with the contents of the database, the customer will get either no answer to the query or an excess of answers that might not be sorted in any usable way. The situation is even more frustrating if the customer does not know exactly what qualities to look for.

The basic idea of CBR is to solve new problems by comparing them to problems already solved (Lenz 1998). The key assumption is that if two problems are similar, then their solutions are probably also similar. CBR systems are based on some measure of this similarity. Old problems and their solutions are stored in a database of
cases — the case base. Often the cases are stored as collections of attribute-value pairs but for complex tasks it is useful to explicitly represent the hierarchical structure of the cases by describing them as structured objects, using inheritance, object decomposition, and possibly other relations between the object parts.

When a new problem is presented, a CBR system searches the case base for the most similar old problem and, if necessary, adapts the old solution to meet the new requirements.

When a new problem has to be solved, the CBR system searches for the most similar old problem. The solution to this old problem can be adapted to more precisely meet the requirements of the new problem. Figure 19 illustrates the steps taken in a case-based reasoning system. One of the strengths of CBR is its proximity to the human way of problem solving (“Yes, this reminds me of something . . .”, “I have seen something similar before . . .”). Human beings typically think in terms of similarities and preferences. These concepts often can be directly mapped to similarity measures as they are used in CBR. For configurable products such as workstations, automobiles, complex machines, and so on, the solution is not only the part number, but possibly the entire configuration.

### 2.5.1 Similarity function

CBR systems have more specific domain knowledge built into them than do ordinary database systems. The main part of this additional knowledge is implemented in a similarity measure, a function that assesses the similarity of a given query to the cases in the case base. The similarity values are ordinal values that are often normalized to the interval [0, 1]. A value of 0 means “does not satisfy the query at all” and a value of 1 says “that’s exactly what you asked for.”

For a better explanation how similarity measure is used to find the best solutions for a given problem, consider the cases being represented as a fixed length vector of $n$ attributes. These attributes can have numerical values or their values can be arranged to reflect some kind of order. For example, part of the description of an electronic device might be:
CBR systems use similarity measures to find the stored cases that are closest to the new cases. The cases are distributed over an n-dimensional problem space. The circle around the query Q indicates a similarity threshold for the cases to be retrieved.

The problem space can now be seen as an n-dimensional space where similar problems are placed close together (see Figure 20). The term “close” is defined by the similarity measure. When a new problem is presented to the system, the similarity to all the cases in the case base is calculated (the number of visited cases can be optimized by suitable indexing techniques, as described in the following section).

The cases within a similarity range of a certain threshold (or sometimes a fixed number of similar cases) are then presented to the user. This retrieval approach has a major advantage over retrieval techniques that use hard selection criteria: near misses are avoided. If a customer asks for features that cannot be fully satisfied by products from the database, the system still offers something close to the request. The solutions are ranked by their similarity to the query. This is essential when the system finds dozens or hundreds of solutions. Unlike rankings in other retrieval systems (such as some Internet search engines), this ranking is based not only on some statistical frequencies but on knowledge that has been acquired from real experts. To define such a similarity measure we start with individual (in CBR terminology, local) similarity measures for all the attributes. For numerical attributes a natural approach is to calculate the difference between the query value and the case’s value and normalize the result to the interval [0, 1], for example by applying the formula $1 - \left(\frac{d}{d + 1}\right)$ to the difference $d$. Returning to our description of an electronic device, it is often the case that the tolerances for the supply voltage of a device are much tighter than the user’s tolerances for the device’s maximum power consumption. The similarity function must reflect this domain-dependent knowledge. In real-world
applications not all attribute values are numerical values. A case description very often contains Boolean or symbolic attributes.

In such cases, the local similarity measure can be described by a table that defines the similarities for all possible pairs of attribute values. Even more complex attribute types such as taxonomy types or complex objects are sometimes required.

Once the local similarity functions are defined, the global similarity of two cases must be derived from the local similarities. The usual way to do this is to apply a weighted sum to all the local similarities. Consider a query q is described by the attributes q1, . . . , qn and a case c is described by c1, . . . , cn, where attribute values with corresponding indices belong to the same attribute. The similarity s between q and c can be calculated from the local similarities s(i) as follows:

\[ S(q, c) = \sum_{i=1}^{n} w_i S_i(q_i, c_i) \]

where

\[ \sum_{i=1}^{n} w_i = 1 \text{ and } w_i \geq 0 \text{ for all } i \]

Although the weights \( w_i \) have to be assessed by a domain expert, they can be manipulated by the user to express individual preferences. Sometimes they can also be automatically assessed by inductive learning algorithms. This general method of defining a similarity measure as a weighted sum can be modified and refined in many ways. In general the computation of the similarity measure depends on the data model and the type of application. A simple weighted sum is not always adequate. For example, nonlinear dependencies between the weights can cause the relevance of a certain feature to depend on the values of other features. CBR offers a number of possible solutions to account for such nonlinear similarity measures.

It is also possible to use more general similarity measures than weighted sums. Although complex measures make it hard, if not impossible, to use specialized indexing techniques to improve retrieval times, the retrieval results can be very accurate.

Finding a suitable similarity measure is often the most critical part of the design and implementation of a CBR system. Once a good similarity measure has been found and implemented, maintaining a CBR system is rather easy. Typically, both the similarity measure and the domain model can remain unchanged longer than can the knowledge bases of other types of expert systems. Often the only part that undergoes significant changes over time is the case base.
2.5.2 Indexing

To find the cases that best satisfy a query, the similarities of the query to all cases in the case base must be calculated. But it is not always necessary to do a linear search. A number of indexing techniques have been developed to reduce the retrieval times of CBR systems. Generally it is difficult to adapt standard database-indexing algorithms to similarity-based systems because database approaches support only hard selection criteria and do not take soft similarity values into account.

One example of efficient indexing structures is a k-d tree (Wess, Althoff, Derwand, 1994). This modified version of a decision tree can significantly reduce the number of visited cases while still accounting for soft similarity values. Another approach to fast case retrieval uses case retrieval nets, which represent the cases and their attributes as interconnected information entities. After the query’s information entities are activated, a spreading activation algorithm is used to retrieve the best matching cases. They are also highly scalable. Both of these indexing approaches significantly speed up the retrieval times by compiling the similarity measures into indexing structures. They become inefficient if the case base must be updated frequently—that is, if the number of updates is in the order of the number of read accesses.

2.5.3 Adaption

CBR also differs from standard database approaches in its ability not only to retrieve existing cases but also to adapt case solutions to the new problem and thus create new solutions. Thus CBR offers support for interactive or automatic product configuration. Of course, this requires the solution (here, the product) to be configurable. A simple example is a CBR-based catalog of used cars. A customer enters a number of preferences into the retrieval system. One preference may be that the car has a sliding roof. The dealer’s stock may not contain a car with a sliding roof, but a CBR system with adaptation capabilities will know that a sliding roof can be built into a car later. Thus the system will automatically adapt the descriptions of the cars that best match the other user preferences to contain an entry saying that there is a sliding roof and will automatically add the price for the sliding roof and its installation to the price attribute. This process can be totally transparent to the user. The knowledge needed to perform this kind of adaptation must be represented in some suitable form during the development of the CBR system. One possible representation is a set of rules that perform certain actions given that the required preconditions for the actions are valid.
A rule for the sliding roof example could be written as follows:

```plaintext
if query.slidingroof == true and
    case.slidingroof == false and
    case.cabriolet == false
do: case.slidingroof := true;
case.price := case.price + $200
```

Depending on the application domain, the adaptation process can be more or less complicated.

While many products cannot be adapted because they have no modifiable structure, adaptation is indispensable if the system must perform some kind of configuration task, like interactive configuration of personal computers.

### 2.5.4 Architectures for eCommerce using CBR

Next figure shows an architecture overview. The system consists of three main components, forming three-tier architecture:

1. The Product Database stores the product information.
2. An Application Server contains all components for e-commerce and the Internet connection. Additionally, the application server contains a Database Connectivity which communicates with the database.
3. WWW-Browser as the client communicates with the application server over the Internet. It has two components: the Sales Assistant to advise the customer and **Presentation** to show the query results.

The application server contains the usual components for e-commerce like modules for distribution, accounting, etc.

![Figure 21 General Architecture](image-url)
Placing the CBR technology in the client could be accomplished via browser plug-ins or Java-applets. In this case just the presentation-task of non-trivial domains leads to a fat client. This is of no further concern as long as the context of utilization is an intranet, with its usually good network performance.

When about e-commerce the Internet bandwidth is too small for a fat client. A fat client would have to travel over the network for each customer, increasing network traffic considerably. Then a thin (ner) client must be designed that contains what is needed for the representation and the sales assistant (or loads additional components during a sales session). Consequently, the application-server or the database should do the main work.

A main scenario can be described when a possible architecture is designed:

1. A customer surfs in the Internet and reaches the company’s homepage.

2. First gets some introductory HTML pages, which are quickly downloaded (because of their small size). There is an option to get intelligent customer sales assistant. Choosing this option will start downloading a Java applet to the clients machine or activate a special Plug-in.

3. The HTTP-Server is responsible for the client’s requests for HTML pages. After the download, the HTTP-Server delegates the control of the client’s interaction to the knowledge server. The Knowledge Server is a service which includes the domain knowledge; in particular, it manages the similarity calculation. After the delegation, the HTTP-Server is able again to answer page requests from new customers.

4. The sales assistant – a Java applet or a special Plug-in – now directly interacts with the knowledge server.

The knowledge server controls the dialog with the user. It accepts queries from sales assistant component and performs the similarity calculation between the query and the products stored in the database. Furthermore, the application server has to perform the mapping between the object model (used for case representation) and the data stored in the relational database. In addition, it controls the dialog with the user via the sales assistant to refine a given query and restarts the retrieval if necessary.

In architecture design presented in next picture, the communication traffic between application server and database could be very high, because of similarity calculations, in the worst case all data travels from the database to the application-server; this could happen again for each client’s request. This disadvantage could be avoided by bulk loading all cases into the application server’s memory when starting the system the first time. However, this wastes a lot of memory and might lead to consistency problems between memory and the database (if the data fits into memory at all).
If the e-commerce server and the database are kept separate on different hosts then other approach must be followed. Imagine a broker offering last minute travels to customers over the Internet. This broker is not an autonomous travel agency, but rather a mediator between travel agencies and customers who are searching for information to plan their holidays. As a consequence, the databases of the travel agencies are not in the same location as the broker’s application server. The disadvantages of previous architecture are: all cases travel over the network from the database of the travel agencies to the broker’s application server. The consistency problem of bulk loading or an index is hard to manage.

On the next figure is designed an architecture where the net traffic will be largely reduced: only the query and the appropriate results – the most similar products – have to be sent over the network.
One solution for the problem mentioned above is to put a big part of the retrieval engine close to the database. On the same host with database we can have a retrieval engine that communicate (based on fast protocols i.e. RMI for Java) with the main retrieval engine to exchange objects – query and results – between the knowledge server (Retrieval Engine A) and the DB-Retrieval module (Retrieval Engine B). Even if the knowledge server gets the customer’s query, it just delegates the query to the second part of the Retrieval Engine (B), where the similarity calculation will be done. Another component inside the database maps between objects and rows of the relational table. Between Retrieval Engine B and the database is suppose that most of the communications traffic will take place, but it will be limited inside one host. After calculating the most similar cases, only these cases have to be presented to the Retrieval Engine A, thus network traffic will be reduced to the query and the few most similar cases. Furthermore the work could be distributed between the Retrieval Engines (RE) A/B, the Mapping Module and the database. While RE-A is serving customer’s requests, might do some of the Sales Assistant’s work, RE-B is calculating the similarities of the results coming from the Mapping Module. The database is processing the given SQL-queries from the mapping tool.

Most of the important database vendors are providing some possibilities to run inside of the database some customized application a new option can be for delegating some of the necessary work into the database. In this variant, the retrieval engine and the object-relational-mapper are integrated inside the database (for example - in a complete Java-virtual machine). A higher performance is expected, because the DB-vendor may use inter process-communication or shared memory concepts between application and the database.

### 2.6 Knowledge Repository Search Software

#### 2.6.1 Introduction

Case-based reasoning (CBR) is the process of solving new problems based on the solutions of similar past problems.

Case-based reasoning has been formalized for purposes of computer reasoning as a four-step process (Aamodt, Agnar, Plaza - 1994):

1. **Retrieve**: Given a target problem, retrieve cases from memory, that are relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.
2. **Reuse**: Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation.

3. **Revise**: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.

4. **Retain**: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

iKRS is a software platform developed within Ropardo SRL that allows users to search information in a database by multiple fields and retrieve, a set of results filtered by a specified field relevance. For example you have Field A and Field B, ideally you want a result that has value X for Field A and value Y for Field B but, the result would still be relevant if Field A has the value X but Field B has a different value so, when performing the search we specify that the Field A is a 100% relevant and Field B is less relevant (say 50% relevant). This technique greatly reduces the time needed for finding specific information and is superior to SQL instructions for complex cases.

iKRS works with the following notions:

- **Case** – each problem encountered in the field is encoded as a case. A problem has a definition and a solution. The definition part is used when performing searches. The solution part is shown to the user and describes how the problem was solved. To use an object oriented language a case (problem) is an instance of the case (problem) type. Whenever we speak of cases we speak of concrete instances (with real values for the fields) from the case type.

- **Case type** – all the problems having same definition are assigned to the same case type. The search is performed over all the problems sharing same type.

- **Similarity** – it is used to compute the distance (how close) between the needed information and the problem description. It has real values from 0 to 1. 0 means there is no match between the search and the problem. 1 means a perfect match; all the search terms are present in the problem definition.

- **Weight** – the search is performed field by field; for some situations some fields are more important (relevant) in the search than others. The user can specify this by setting a weight to a specific field in the search. The weight values are from 0 to 100. A 0 weight means the field is not used during the search. A 100 means the field is fully used in the similarity computation.
The iKRS platform can manage tens of case (problem) types but all operations are done in the context of the selected case (problem) type. For the current implementation the relevant type is Best Practice.

2.6.2 How to Use iKnowledge Repository Search (iKRS)

2.6.2.1 Connecting to iKRS

To access iKRS you only need to type the following URL into your browser's address bar: http://ikrs.host4u.ro/iKRS (temporary address).

![iKRS: case type chooser dialog](image)

When the page loads you will be prompted to choose one of the available case types before continuing to the application.

The name of the selected case type will be displayed at all times and the application is now ready for use. At this point the user can choose to search for a case, switch the case type or add a new case to the database, the rest of the operations from the menu (opening, saving, printing and deleting will be available only after a case has been retrieved and selected). All of these actions will be explained in the following sections.
2.6.2.2 Basic Operations

Searching a case

Searching for a case is straightforward and simple: you just need to click the “Search” menu item from the menu bar and the search dialog will appear. A search is done using the attributes defined in the case description component; all these fields will appear in the search dialog.

The dialog itself is composed from two main areas:

- the search fields;
- the configuration options.

The search fields area contains all the fields from the case description and the configuration options area contains two options for configuring the search (Automatic and Custom), the default being Automatic Configuration.

With the default configuration selected, all you have to do is fill in the fields you want to search by and press the “Search” button, the dialog will close and the results will appear on the main page of iKRS. This configuration option reads all the non-blank fields from the “search field’s area” assigning them a weight of 1 and sets the maximum number of cases retrieved to 5.
If you want a more specific, complex search, you can use the “Custom Configuration” option. This option will allow you to set different weights for each non-blank field from the “search fields” area, also giving you the possibility to specify the maximum number of cases retrieved. To use this option you have to select the “Custom configuration” option and click the “Search” button, instead of the dialog closing and showing the results, this time a new dialog box will appear. The new dialog box will contain all the fields that you entered a value for in the previous dialog (Search Dialog) and a slider bar and a spinner control used to set the field's weight for each field. You can use either one of these controls (slider bar or numeric spinner) to set the weight of the field.

Computing a similarity result means computing a weighted average on the fields equality, so setting a weight from this dialog sets the field's weight when calculating the weighted average, this way a field can affect the result of the similarity in a more direct or indirect way. Using weighted average and allowing users to set each field's length contributes to defining complex and goal oriented queries.

You can also set the maximum number of cases to retrieve from the bottom area (above the button panel). After all these settings have been made, you can click the “Search” button which will close the dialog and show the results on the application's main page.
As stated before, the main page of the iKRS application will display the results of the search, and also the parameters used in the search.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Name</th>
<th>Case Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Auto Maintenance</td>
<td>0.67</td>
</tr>
<tr>
<td>36</td>
<td>Auto Maintenance</td>
<td>0.67</td>
</tr>
<tr>
<td>9</td>
<td>APOP (Advance Product Quality Planning)</td>
<td>0.33</td>
</tr>
<tr>
<td>5</td>
<td>Critical Chain</td>
<td>0.33</td>
</tr>
<tr>
<td>7</td>
<td>Call Design</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Figure 27 iKRS: the "Weights" dialog

Figure 28 iKRS: the search parameters area

Figure 29 iKRS: the results of the search displayed in a table
2.6.2.3 CRUD Operations

Creating a case

To add a new case you have to select the “Case Management / Add Case” menu item, this will add a new tab on the main page and focus it.

![Figure 30 iKRS: the "Add Case" tab](image)

At the top of the tab there is a button panel which contains the “Save” and the “Close” button, clicking on any of these buttons will close the tab but the “Save” button will save a new case constructed with the values of the fields displayed in the tab before closing the tab.

The tab contains the fields for the case description component (on the left) and the fields for the solution component (on the right). These areas are resizable and scrollable.

After clicking the “Save” button, a message box will appear notifying the user if the case has been saved successfully or not along with the new case ID (if save succeeded).

![Figure 31 iKRS: the save message](image)
Viewing a case

A case can be opened for viewing in three ways:

- from the Case Management menu
- by double-clicking a row from the result table
- by clicking the “Details” button present on each row from the result table

In order for opening a case from the menu bar, a case from the results table needs to be selected (otherwise the menu item will be disabled).

![Case Management Menu](image)

Figure 32 iKRS: opening a case from the menu bar

![Case Table](image)

Figure 33 iKRS: Opening a case by double clicking a row or by pressing the Details button

After using one of these methods to open a case, a new tab will be created, and focused, that will display the case and the related actions (which are also available from the “Case Management” menu).
Figure 34 iKRS: An opened case

Updating a case

A case can be updated only from the “Case view tab” (Error! Reference source not found.). You can modify both the case description component and the case solution component and click the “Update Case” button located at the top of the “Case view tab”.

Figure 35 iKRS: the "Update Case" button

After you click the update button you will get a message confirming that the case has been updated successfully or a message displaying the error.

Figure 36 iKRS: the update confirmation message
Saving as a new case

This option allows you to save the displayed case as a new case in the data base (this implies generating new id's for case description and the case solution components).

You would generally use this feature when you want to save a case that is similar with a case retrieved as a search result. Using this option is a time saver because you don't have to write all the data from the ground up, you just modify and adapt the existing one.

NOTE: You have to make the desired changes before clicking “Save as New Case”

This action is available both from the “Case Management” menu and from the “Case view tab”. If the action is executed from the menu, the new case will be identical with the case selected from the result table (except for the id's) and no editing window will appear.

![Figure 37 iKRS: Saving as new case options](image)

If the action is successful, a message similar to the one shown when adding a new case will be displayed, confirming that the case has been saved and showing the new case's id.

Deleting a case

You can delete a case by using the “Delete Case” menu item from the “Case Management” menu, or by clicking the “Delete” button from the “Case view tab”.

To avoid accidental deletion of a case, a user confirmation is required, afterwards the case is deleted from the database and a message is shown, informing the user if the delete was successful.
2.6.2.4 Additional Operations

Printing a case

The print feature is available, as you've already guessed by now, both from the “Case Management” menu and the “Case view tab” (through the “Print” button). This feature will export a custom report (built with Jasper Reports) specific for each case type, to a .pdf format.
After the print action was performed, your browser will ask you to save a .pdf file to your computer, or download it automatically (depending on your browser/settings). The file name is composed from the case type, the name of the case and the id of the case.

Switching the case type

Switching the case type allows users to work with a different case type than the one selected when the page first loaded.

This option is available only from the menu of the iKRS application.

After clicking on the menu item, a user confirmation dialog will appear to avoid data loss by accidentally switching case types (when switching the case type all unsaved data will be lost).
If the user confirms the change of the case type, all temporary data will be deleted and the user is prompted to choose a case type from the “Case type chooser” dialogue (Error! Reference source not found.).

2.6.3 iKnowledge Repository Search (iKRS) Public API

Third party application can use the CBR system through two simple web services:
iKRS WebService – provides create, retrieve and update operations for CBR cases
iKRS Admin WebService – provides methods for cleaning the CBR and retrieve raw information (all the case types and cases available in the database)

2.6.3.1 Data Structures

The communication through the web services is done using a small number of data structures. These are simple POJO objects with a few metadata fields and XML serialization for the case definition and solution. The case solution can have a file attachment in which case it is sent using MTOM extension. The structures are kept simple and extensible.

The actual case definition and solution are sent as XML content over the web services. The web service client will need to de-serialize them in proper structures using case-based knowledge.

- **CbrCaseType**

  The CBR system can hold many types of cases. The CbrCaseType is used to distinguish between different types of cases used during the operations. The fields in the structure are:

  - String caseTypeId – it is a unique identification field, each case type needs one. It is provided and registered by the knowledge engineer during the implementation of a new case type.

- **CbrDefinition**

  The case definition is stored in a CbrDefinition data structure. The available fields are:

  - String xml – the case definition is serialized as an XML for transfer over the web services

- **CbrSolution**
The case solution is transfer using a CbrSolution data structure. The available fields are:

- String xml – the case solution serialized as XML
- File attachment – an optional file attachment for the case solution; if it is available it is sent using MTOM (and requires a MTOM compatible client)

• CbrCase

The entire case (definition and solution) is sent as a CbrCase structure. The available fields include some meta-information related to the case and the case definition and solution:

- long caseId – it is the internal id, unique for each case of a given type; it has to be provided in some WS operations. The same case id may be found for cases having different case types (it depends on specific CBR implementation of each case type).
- CbrCaseType caseType – store the type for the case
- Date createdOn – the creation date for the case (in the CBR database)
- Date lastUpdated – the date of the last update performed on the case
- String remarks – it is a free field which can be used to store various remarks or information related to the case
- double similarity – when the case is retrieved during a search operation this field stores the similarity (score) for the search. It can have values from 0 to 1. A case with similarity 0 won’t be retrieved during the search.
- CbrDefinition definition – the actual case definition
- CbrSolution solution – the actual case solution

• CbrSearchField

Each search in the CBR needs a list of fields to be searched for. Each field is provided in a CbrSearchField structure:

- String fieldName – specifies which field has to include in the similarity computation
- String fieldValue – specifies what value has to be used during similarity computation
- **int weight** – specifies how important (how much weight) the field is in the similarity computation result. It can have values from 0 to 100. If the weight is 0 then the field won’t be used for similarity computation, even if fieldValue is provided.

- **int rank** – specifies an ordering for the field evaluation (the fields are sorted ascendant using the rank). For the same rank the field evaluation order can be seen as random (it’s not truly random but no guarantee for a consistent specific processing order are given).

- **CbrSearchPattern**

The CbrSearchPattern data structure is a simple set of CbrSearchField structures:

- **Set fields** – list of fields to be used during search

### 2.6.3.2 iKRS WebService

The basic iKRS web service allows for search, retrieve and update operations on the cases from CBR.

- **retrieveResults**

The main method that will be used during regular CBR sessions is retrieveResults:

```java
CbrCase[] retrieveResults(CbrCaseType caseType, CbrSearchPattern pattern, int caseNo);
```

The method will search for the cases of a given type using the search fields in the pattern. The first caseNo results are return to the user, sorted descendent by similarity value. If the caseNo is 0 then all the cases which have a similarity strictly greater than 0 will be returned. The sorting for the cases that have the same similarity value it is not guaranteed to remain the same between several searches (even with identical parameters).

- **getCaseSolution**

getCaseSolution may be used to retrieve the solution to a case:

```java
CbrSolution getCaseSolution(long caseId, CbrCaseType caseType);
```

The method will retrieve the solution for the case identified by the caseId and having the requested caseType.

- **updateCaseSolution**
If the user needs to update a solution it can use the updateCaseSolution method (the case definition cannot be updated, if there are some changes then a new case should be added to the CBR).

```java
long updateCaseSolution(long caseId, CbrCaseType caseType, CbrSolution solution);
```

The new solution will replace the existing one. To safely update the solution the operation sequence would be:

- getCaseSolution()
- change the solution
- updateCaseSolution()

The method returns the case id if it is successful (the same id as the one from parameters) or a negative value if there was an error.

- addCase

The addCase method is used to create new cases in the database. The case can be new or it can be a slight modification of an existing ones (i.e. if the definition changes).

```java
long addCase(CbrCase case);
```

The method takes the case definition and solution and persist them. From the meta-information fields only caseType is required. The remarks field can have a comment or remark from the user or it can be empty. The other meta-information fields are not used (as they are filled by the server) and can have any value.

The returned value can be a positive value (and represents the new case id) or a negative one if there is an error.

### 2.6.3.3 iKRS Admin WebService

The admin web service methods can be used to manage the CBR database and case types. If additional operations are required they can be added but most of the management operations are done by the knowledge engineer during the setup phase.

- zapDatabase

The zapDatabase method will clear all the stored cases for a given case type.

```java
void zapDatabase(CbrCaseType caseType);
```
It can be used to reinitialize the database for the caseType. No checks regarding dependencies between cases are made and all data is deleted (including the attachments to case solutions).

- getAvailableCaseTypes

Method getAvailableCaseTypes will return a list with the case types register in the CBR. The user can select any case type and start working on it.

CbrCaseType[] getAvailableCaseType();

This method should be call before starting working on CBR to allow the user to select a specific case type.

- getAllCaseIds

There may be situation in which the user needs to know how many cases are in the database. The getAllCaseIds will return all the ids for the cases having a given type:

long[] getAllCaseIds(CbrCaseType caseType);

This has limited usage, for statistic or maintenance purposes (the user can retrieve only case solution given a caseId, not the case definition). The normal method to retrieve cases is to call retrieveResults.
Capitolul 3  **GDSS software framework**

iDecisionSupport is a collaborative environment designed as a framework for decisional support tools developed within Ropardo S.R.L.. This means that different (third party) decision support tools can be plugged in the system and work together to support decisional meetings where users can reach a consensus or make a decision regarding different issues. In order to achieve this functionality a system is built on a distributed architecture as depicted in Figure 45.

![Figure 45. iDecisionSupport general architecture](image)

iDecisionSupport (iDS) is made up of the following components:

- the iDS Server which is the central component that handles the decisional meetings, register tools, provide access rights for users and many other aspects that will be described later in the document

- the iDS Web Client which provides web access/interface to the whole system

- the decision support tools which implement various parts from decision processes; they can be used to model complex decisional workflows

- collaborative platform, used for email and calendar features (the current implementation uses Domino Server)

Each of these components (including each individual tool) can reside anywhere on the internet. The integration is done through web services exposed by the iDS server and the APIs exposed by the collaboration platform.
### 3.1.1 iDS Server

The iDS Server are the central component which is responsible for the following:

- managing the project based concept and the tree structure of decisional meetings – the meetings are grouped in plans, each plan can contain meetings or other sub plans and plans are grouped in projects. This way the decisional activities within a company can be structured around projects real projects that take place in that company.

- managing the meetings – each meeting is divided into several phases (initial, commitment, work, finished). The server is responsible for managing the process of stepping from one phase to another and assures that the meeting data is consistent through the flow. After a meeting is over the server is responsible for storing the meeting results and providing it as input for other meetings, if needed.

- access rights – the meetings, plans and projects have a list of members along with their roles (rights) and the server is responsible for enforcing those rights. There are 3 main roles in the system (Facilitator, Active and Observer) that can be set for the above entities (Project, Plan and Meeting) and another role for the system administrator which allows the user to enter the server administration area.

- xslt transformation – it is used to provide optional data transformation between tools.

- managing tools – each decision support tool must be registered in the server so it can be used for decisional meetings. The server is responsible for registering them and relating each meeting to its selected tool and for providing the tool with the necessary data to support the meeting. This is done through the tools web service (see Figure 46).

- generating meeting reports – the server is responsible for generating reports for finished meetings using the meetings results.

- shared plans templates - the server supports templates for shared plans chains; after each meeting the facilitator can choose the next meeting from the available ones in the template.

- tasks – the server provides a task manager which can schedule various tasks to run and perform maintenance activities; each tool can register its own tasks to run in the server context.
- calendar – the server manage its own internal calendar, generated from the meetings and schedules in the system; it is possible to synchronize it with user’s calendars on various external platforms, if the user allows it

- notifications – handle various types of notifications for the users (emails, calendar)

- administration – user / roles/ groups / agents / workflows / report templates administration

![Diagram of iDS Server Components](image)

- Figure 46. iDS Server Components

### 3.1.2 iDS Web Client

The iDS Web Client is a web application (portlet) that provides access to the system through a web browser. It is responsible for:

- displaying a dashboard that shows the projects and the meetings for the user

- showing a detailed view of the project tree structure which offers CRUD operations for projects, plans and sessions

- controlling the meeting phases – provides controls for going from a meetings phase to another

- providing administration actions

- providing placeholders for meeting tools

The last feature is the place where the user actually interacts with the tools. The web client provides placeholders (implemented as iframes, currently) inside the meetings
views. The decision support tools will render themselves in these placeholders using the context of the current meeting and user.

3.1.3 iDS support tool

A decision support tool (see Figure 48 for a general view) is a piece of software that is used by the iDS framework for performing decisional meetings. There can be voting tools for meetings where participants need to reach a consensus over some issues. Or it could be a brainstorming tool for a meeting where the participants must spawn ideas for different issues. Tools may be aware of one another and are able to exchange results through the iDS Framework. Tools can be web based or desktop applications and (in the web application case) must be able to run in parallel in the context of different participants, from separate decisional meetings that can be hosted by more than one iDS server. A web-based tool will run inside an iframe provided by the iDS Framework. Each meeting (in the iDS Web Client) contains placeholders (iframes) where the tool can be accessed. The tool must render the meeting’s context related information. There are two types of tools: for commitment and for meeting.

Commitment tools are used to support the commitment phase of a meeting (if it has one). The commitment phase is used when the invited participants are required to confirm their participation or motivate if they are not available. This phase can be used to discuss and define several other parameters for the meeting (like the issues to be decided or the duration).

Meeting tools are the ones that are used in the actual meeting phase and must be able to work in three different modes: config, work, and report.
- config mode is used when the meeting is not started yet. It allows for the users to configure the meeting (add issues, set anonymity, select tool parameters like voting type, etc)

- work mode is used after the meeting is started and allows the users to do the actual work (vote, brainstorm etc)

- reports mode is used after the meeting is finished and presents read-only (report) view of what happened in the meeting

![IDS Tool general view](image)

### 3.1.4 IDS Web Client - Architecture Design

The main web client’s functions are to facilitate the access to the server using a graphical interface (web-based). The user has to:

- login in the system
- manage the structures (projects, plans, meetings)
- attend meetings
- generate or view reports

The web client is implemented as a JSR-286 portlet\(^1\) and runs inside iPortal (a Liferay\(^2\) portal customization).

---

3.1.5 Login

Login in the system is done through iPortal portal using the standard login page. It uses the SSO provided by the iSecurity through iPortal. The users from the server database are synchronized with the iSecurity to have the same information in both places. The system can be configured to work with its own set of users, standalone, but this makes the system less user-friendly while run inside a portal (the portlet can be accessed only by already authenticated users).

3.1.6 Manage projects

The main view open in client is the dashboard view (Figure 49). It presents a list of projects the current user is member of (and has access to) and the list of open meetings in these projects (decision map).

In the list of projects the user can:

- create a new project (if enough rights are granted)
- search for a particular project using either the simple or the advance search fields
- view the project by clicking on its name
- close the project (if enough rights are granted) using the project’s Actions button
- re-open the project (if it is close and enough rights are granted) using the Open link (visible only when the project is closed)
- change the project (if enough rights are granted) using the project’s Actions button

- Figure 49 Web Client – Dashboard View
### 3.1.6.1 New project

When the user needs to create a new project the Project Information view is used (Figure 50). The fields have default value. The possible actions are:

- **Save** – saves the information in database and keeps the view open for further changes
- **Save & Close** – saves the information in database and close the view returning to the dashboard
- **Close** – cancel the changes and returns to the dashboard

The user has to provide:

- project name
- author (by default is the current user)
- a short description
- the timeframe the project is valid (from – to)
- list of members and their possible roles in the project

![Figure 50 Web Client - New Project](image)

### 3.1.6.2 Project information

When the user clicks the project name then the project information view (Figure 51) is displayed. The view is similar with the new project but the information is read-only, no changes are allowed.
In the upper bar there are other 3 views available:

- decision map – lists the open meetings in the current project
- decision view – lists the plans and the meetings in the current project – it is used to structure the project
- discussion list – allows the users to discuss various topics related to the project
3.1.7 **Manage plans**

The plans management is done in the project’s decision view.

The user can create and delete plans and meetings. The plans tree structure can have any number of levels. The meetings are leafs and cannot have any descendants. Meetings can form chains by taking information generated in previous ones and can offer information from the current node to successors.

Creation of new plans is similar with the creation of a new project.
Meetings are created from the decision view.

Each meeting has:

- subject
- timeframe (date and time)
- session type (which iDS tool to be used)
- topics (list of free text items which will be provided to the session tool)
- participants
- description

Figure 55 Web Client - New Plan
Meetings can be chained such that output from previous ones can be used as input into the current one (Figure 57).

Finally, a meeting can require confirmation in which case notification emails are sent to the participants. They are expected to follow the links and confirm their commitment to the meeting.

After a meeting is defined it presents 3 sections:

- **general** – corresponds with configuration and commitment phase, changes can be done to the meeting
- work – is active when the main meeting is taking place
- report – becomes active after meeting has concluded and contains the reports generated during the meeting

Meeting can be moved into the next stage manually using a Next button.

Figure 58 Web Client - Meeting in configuration

For a meeting in progress the meeting (session) tool is display in the iframe placeholder. In Figure 59 the tools in Action Plan (developed in GWT3).

Figure 59 Web Client - Meeting in progress

3 http://code.google.com/webtoolkit/gettingstarted.html
When the meeting ends (either by reaching the allowed time or by using the Finish action) the reports are generated (if available) and the users can see the results in the Reports view.

**3.1.9 Manage reports**

The iDS system generates reports using JasperReports library, as PDF files. Each tools can have a number of report templates defined which can be selected for reporting. The PDF files can be downloaded for later usage or can be sent by email (as notification at meeting’s end); the user will follow the link and will land in the meeting’s report view.

![Figure 60 Web Client - Meeting reports](image)
Capitolul 5 References


C. Candea, Referat Doctorat: Sisteme bazate pe agenti inteligenti pentru asistarea deciziei, Mai 2010


