An Automated Agent-Based Marketplace For Mobile Internet Access

Diplomarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Erich Bircher

2003

Leiter der Arbeit:
Prof. Dr. Torsten Braun

Forschungsgruppe Rechnernetze und Verteilte Systeme
Institut für Informatik und angewandte Mathematik

Contents

1 Introduction

1.1 Abstract

In our mobile society it is desirable to have access to the Internet and private networks from abroad. Thanks to new technologies such as Wireless LANs and mobile phones this is possible with increasing tendencies in many areas. Traditionally one would make a long-term contract with an ISP and then be able to use its and the roaming partner's access points. In the course of the delegation of tasks to computers, this negotiation and the conclusion of a contract could be done by a user supporting software, too.

Things could even be taken one step further by making these computed contracts on a short-term basis. That would mean to negotiate a new contract for every desired connection. Since the user is not bound to one ISP with one contract anymore, this would give him more flexibility and access to yet more access points. Furthermore, he could use different access technologies and he could get the most suitable connection for him (with regard to the price) and for his applications, that may demand certain network qualities (bandwidth, jitter, delay etc). Such a software could further configure the portable device of the user and establish the desired connection on time. With this and QoS supporting ISPs it would be possible to have a guarantee for certain network qualities, e.g. a guarantee for enough bandwidth to hold a video-conference on a particular time at a particular location. The idea of this diploma thesis was to realize such a software system.

The system has been realized with an agent-based marketplace. On this market, the User Agent (the software entity representing the user) can contact eligible ISP Agents (the software entity representing the ISP) and negotiate a SLA. The User Agent gets the addresses of the ISP Agents from the Marketplace Agent, which has yellow pages functions. In exchange, for payment, the User Agent gets the right to use an access point and the therefore necessary configuration data from the ISP Agent. The agents have been realized with FIPA-OS, an agent platform written in Java.

Beside a stand alone version of the software, there has been realized a demonstration version. This allowed to show not only the negotiations between agents, but also the results of them, the change over to different access points. This lab system run mainly on Linux boxes. As wireless access points 802.11b has been used, the quasi standard for W-LAN.

Furthermore, there have been made some tests on this demonstration system to look at the performance of such a system. In these tests, the delays for conversations between two agents, e.g. the negotiation of a SLA, have been measured in different scenarios. All tests yielded reasonable results, with times ranging from under one second to three seconds. Furthermore, the results of the test conclude, that the biggest fraction of the duration of a conversation is used by the platform and agent tasks handling it. This implies, that agents, that are likely to be contacted by other agents in great quantity, need to run on a powerful hardware to handle all requests in time.

1.2 Diploma Project Overview

Thanks to mobile data communication technologies like W-LAN and GPRS, traveling around and having network access is possible. A user could just conclude a contract with an ISP to use their access points. But with this the user has no guarantee for free network resources, in the worst case he could even have problems with network quality sensitive applications. It would be nice for a user to be able to reserve in advance the needed network resources. Furthermore, if his IPS does not cover the area he is in, he can not communicate. And if the user wants to connect with a new technology, he has to get a new ISP. Generally, user needs vary greatly.

A solution to these problems is being proposed in this diploma thesis. The idea is to negotiate for every desired connection a detailed short-time contract. This would allow flexibility and guarantee a defined connection quality. The negotiation of these contracts and later the establishment of the connection would be done by software supporting the user. The user just has to tell the software his connection needs at certain points (time, location, application) on his trip and the software tries to get short-term contracts with ISPs that fit the users needs. For example, a business man has to travel to London from Berne. To use his time he wants to write some emails during his train-ride to Zurich, and after checking in at the airport he wants to have a video-conference with some business partners. He just has to tell the software this information and the rest is done by it. The software checks if there are access points at the desired location and tries to negotiate an appropriate contract. For this example the software could conclude a contract with a Mobile Phone Operator for the train ride and for the video-conference at the airport a contract with an ISP offering Wireless LAN access. Since the user may be delayed or has to cancel his trip, the software has to be able to handle these cases, too.

The goal of this diploma thesis is to realize a software that negotiates contracts with ISPs and manages the connectivity of the mobile device. But what are the eligible technologies for mobile connectivity, what are the contracts (called Service Level Agreement (SLA)) between the user and an ISP about and how can problems with mobility be solved? Chapter 2 of this document tries to answer these questions. The Software has to contact the eligible IPS and negotiate with them a contract. Establishing contact, negotiating and concluding a deal is all done on a marketplace. Since the software has to implement a marketplace, Chapter 3 looks at marketplaces, what they are and what to consider to implement such. Thus if such a software is to work, entities of ISPs and marketplaces have to be part of it. One way to implement such distributed communicative systems is to use intelligent software agents, Chapter 4 introduces this technology. Related work is presented in Chapter 5. The design of the software is described in Chapter 6 while Chapter 7 is about the implementation of it. As an illustration, a demonstration has been implemented in a laboratory environment with wireless access, which Chapter 8 is about. Some performance tests of the system are described in Chapter 9 while Chapter 10 and 11 finishes this document with a look at security issues and conclusions.

2 Internet Connection Technologies for Mobile Users

2.1 Wired Connections

There are several ways to connect mobile devices to the Internet. The most important ones are listed below. Some are wireless, some need wires. The wired connections are not fully mobile, but still an option due the high availability and low price connections, and the majority of the wireless connection methods are limited to a certain area. Furthermore most work on a portable device is done on a desk and not moving around, and even the best portable devices need to recharge their batteries from time to time. So a wire more is not too big a problem. But the plugs have to be there, and since wireless devices are easier and cheaper to install (W-LAN) or already there (Mobile Phone Antennas) and more comfortable, they will prevail.

2.1.1 Phone Networks

Most buildings have not yet direct connection to the Internet, but they have a phone line. With an analog modem (connected to the phone line plug-in with a cable) there can be set up a connection to the Internet via a provider. Today probably the most used technology for people traveling around with portable devices. Due to low bandwidth (56kbit/s), this is mostly used only for emailing and some browsing. In a lot of industrialized countries there are free providers, so most of the time only local phone charges apply. A high availability, most hotel rooms have phone lines, easy to pay (e.g. payable with hotel phone bill), no security risk (except those of the Internet)and modems built in most portable computers make it still worth a consideration.

max. dist. to access-point:	5-6 m cable
bandwidth:	56kbps
mobility:	minimal, limited to cable
security:	same as Internet
connection costs:	very low
access points:	hotel rooms, private rooms, offices
distribution:	high
devices:	modem card.

2.1.2 Ethernet Plug-In

Another connection possibility with a cable is to connect the computer directly to a TCP/IP-based network. Ethernet[1] is the worldwide standard for Local Area Networks, with a bandwidth of 10Mbps or even 100Mbps (Fast Ethernet). A lot of buildings, specially from enterprises, have Ethernet-networks, most are connected to the Internet, too. Since most notebooks have an Ethernet-card, they can just be plugged with the right cable to a plug-in. But most Ethernet owner will not allow non-internal computers on their networks for security reasons. Such computers can be blocked, also each computer has to be configured correctly. So this might be just an option for special access points for non-internal computers at conference rooms, hotels, Internet cafes and universities.

An advantage of Ethernet is that the connection is very fast, limited only by traffic on other networks. Connections might be available at a low price, too.

max. dist. to access-point:	5-6 m cable
bandwidth:	10Mb or 100Mb
mobility:	minimal, limited to cable
security:	low
connection costs:	low
access points:	hotel lobbies, conference rooms Internet Cafes
distribution:	low
devices necessary:	ethernet card.

2.2 Wireless LANs

Wireless Networks offer unrestricted mobility inside a small area. Ideal for Internet access points abroad. There are 3 unlicensed bands designed for commercial use, 9MHz, 2.4GHz and 5GHz. The higher the frequency, the more bandwidth is available, but also the range declines. There are 3 main technologies today competing on the 2.4 GHz unlicensed band for wireless LANs. The problem with the 2.4 GHz band is that it is already used by a lot of devices like cordless phones, medical gear etc. But microwaves have emissions in the range of 2.4 GHz, too. There may be some problems with increased use of wireless LANs by interference between the three wireless LAN systems and the already existing devices. Some working groups are trying to solve these problems. In the future there will be even faster Wireless LAN standards, the next in 2002 with 802.11a and HyperLAN II, both operating on the 5 GHz Band with a Bandwidth up to 54 Mbps (HyperLAN II 22 Mbps). A problem for wireless LAN is, that everybody can "listen" to it inside a certain range. So the security concept of access control to buildings may be ineffective with regards to wireless services. With specials antennas the signals can be read even farther away than normal wireless devices can. Imagine a company allowing everybody to plug in their computer inside their network! So wireless LAN connections have to be strongly encrypted, not only that nobody can listen to sensitive data, but also that nobody can gain access to corporate networks.

2.2.1 802.11b

As a standard for wireless LANs 802.11b prevails. 802.11b[2] is marketed from WECA (Wireless Ethernet Compatibility Alliance) under Wi-Fi[3] (Wireless Fidelity), too. 802.11b is an extension of Ethernet to wireless communication. It is primarily used for TCP/IP, but can handle other forms of network traffic, such as AppleTalk or PC file-sharing standards, too. The 802.11b specification allows for wireless transmission of approximately 11 Mbps of data at distances up to 100 meters (or even more in the USA) over the 2.4 GHz unlicensed band. 802.11b uses DSSS (Direct Sequence Spread Spectrum), making it more vulnerable for interference than its competitors. If interference appears, it tries to change the channel. The advantage of DSSS is a higher throughput and range. Another problem for 802.11b is security. There is a built in encryption named WEP (Wireless Equivalent Privacy), but is proven to be very insecure, and in public access points it is not even used, because there has to be a known key to use WEP, and if everybody knows it, it does not make much sense anymore. Like any other wireless communication form the receiving quality is greatly influenced not only by the distance between the devices but also by the environment (buildings). If the quality sinks, there are more and more errors and therefore the bandwidth will be automatically reduced from 11 Mbps to 5.5, 2 and 1 Mbps (known as fall-back). So the full 11Mbps may only be available at distances up to 25 meters. Since it is a shared medium the more people it use the less bandwidth they get. 802.11b uses different channels, (9 or 11 channels, depending on the country), but only 3 of them do not overlap with one another, so there can be only 3 independent connections between devices with full 11Mbps in the same area. If more connections are made, the bandwidth sinks rapidly due to collisions. Each radio may act, depending on software, as a hub or for computer-to-computer transmission, but it is much more common that a WLAN installation uses one or more access points, which are dedicated standalone hardware with typically more powerful antennas. These access points often include routing, DHCP servers, NAT and other features.

max. dist. to access-point:	100m
bandwidth:	up to 11Mbps
technology:	DSSS
mobility:	maximal inside scope
security:	low
connection costs:	middle
access points:	hotels lobbys ,restaurants, airports
distribution:	low
devices :	Wireless LAN card

2.2.2 Bluetooth

Another wireless technology is Bluetooth[4]. Bluetooth tries to become a standard wireless interface between different electronic devices. It can be run on 3 different power levels. The lowest is just a replacement of a short cable, like the connection between a laptop computer and a modem, or the connection between a cellular phone and a headset (PAN, Personal Area Network). Furthermore, the strongest offers the possibility to make a wireless LAN. The main difference to 802.11b is that Bluetooth is cheaper, needs less power, might be already integrated in a lot of devices but also offers much less bandwidth and range. It uses the ISM Band at 2.4 to 2.48 GHz, using a spread spectrum, frequency hopping, full-duplex signal at up to 1600 hops/sec. The signal hops among 79 frequencies at 1MHz intervals, making it insensitive against other Bluetooth connections or other interferences. For data transmission Bluetooth supports maximal 723.2 kbps asymmetric (uplink 57.6kbps) or 433.9 kbps symmetric with a master sharing a channel with up to seven simultaneous active Slaves in a Piconet (Point to Point Connection). Several Piconet can be connected to a Scatternet . There are some built in security mechanism. Bluetooth encodes data and uses frequency hoping to prevent eavesdropping.

max. dist. to access-point:	10m
bandwidth:	up to 1Mbps
technology:	DSSS
mobility:	maximal inside scope
security:	middle
connection costs:	middle
access points:	unknown
distribution:	very low
devices necessary:	Bluetooth card

2.3 Mobile Phones

Most populated parts of the world today are covered with mobile phone networks. They can be used for limited data transmissions (up to 56kpbs), too. In the future 3G (Third Generation) networks will allow much more throughput, making it a serious competitor for mobile connection services. UMTS[5], such a 3G standard, will allow bandwidths ranging from 144kbps in rural areas to 384 kbps in suburban up to 2 Mbps in indoor/low range outdoor areas. UMTS should already be running, but high investments, both for technical equipment and for licenses for the frequencies (they have been auctioned for huge amounts), but also technical problems, no availability of portable devices and a low interest on consumer side, have postponed the start of these services for months, so they are not yet an option.

2.3.1 GSM

GSM[6] is the current second generation cellular phone standard, operating at 900MHz, 1800MHz or 1900MHz. GSM allows data transfer at 9.6Kbps or in a compressed mode 14.4kbps. With HSCSD[7] (High Speed Circuit Switched Data) up to 8 different channels can be bundled. In reality 4 are offered today, allowing 28.8 kbps synchronous download/upload or asynchronous 43.2/14.4 download/upload. If the quality of a connection is too bad, the error correction automatically changes from 14.4 kbps to 9.6 kpbs per channel. Furthermore, speech services have a higher priority, thus by high demand the amount of channels can be lowered. Another extension of the GSM network is GPRS[7] (General Packet Radio Service). GPRS is a packet-switched service, which allows a user to be connected all the time, but only the transmitted data has to be paid for. GPRS uses channel bundling, allowing maximal 8 channels for 171.2 kbps (21.4 kbps per channel). In a first phase only 50-60 kbps will be available, with an effective throughput of 30-40kbps. The GSM Networks use TDMA (Time Division Multiple Access) allowing more than one user using the same frequency. The GPRS packets will be transmitted when there are enough capacities available. On heavily used networks only a few packets can be transmitted. Here GPRS has a disadvantage over HSCSD. There was another technology planned before UMTS called EDGE. Edge uses a new modulation technology to improve the bit rate of single time slots (18kbps-48kbps). HSCSD and GPRS could be upgraded to ECSD (144 kbps) and EGPRS (384kbps). But probably EDGE will never be realized. Mobile Network Companies are likely to skip EDGE in favor of UMTS, which demands high investments.

bandwidth HSCSD:	57.5kbps
bandwidth GPRS	171.2kbps (now 40kbps)
mobility:	maximal
security:	high
connection costs:	middle
access points:	almost anywhere in populated areas
distribution:	high
devices:	Cellular Phones

2.4 Satellite Data Services

Another possibility is to use Satellites for data communication. They allow in a large area like a continent or even at every point on the earth real mobile communication, but the antenna has to be outdoor. The user can choose between a lot of different data services. They can be divided into three main categories.

2.4.1 Global Satellite Phone Networks

Satellite Phone Networks[8, 9, 10] can be used for data transmission. This can mostly be done only at a very low bandwidth. Satellite Phone Networks are truly or nearly global (some will not offer services above 70 degrees). Prices for connectivity and devices are high, but most end-devices are relatively small and lightweight.

2.4.2 Global Satellite Data Services

Some Global Satellite Service Providers offer special data services, too. Inmarsat offers a 64kbps bidirectional ISDN Service, which can be doubled by using two devices. With Inmarsat-4 a bandwidth of 400 kbps is planned, coming in 2004. Teledisc plans a satellite network for the Internet in the sky with a bandwidth of 64kbps uplink/2mbps down-link, with a focus on the African continent. Global Satellite Data Services are becoming more and more important, specially in areas with a low teledensity. Furthermore, airlines are an important target market, since some customers wish Internet access during their flights. The equipment is costly and relatively heavy and big, also services are likely to be high priced.

2.4.3 Continental Data Services

There are companies offering satellite data services on a regional level, mostly in North America and Europe. Some of them allow bidirectional asymmetric connectivity. The equipment is also relatively big and not really usable for mobile usage, but connectivity costs are mediocre and comparable to ADSL Services.

bandwidth	9-128kbps
guaranteed bandwidth:	no, shared with others.
mobility:	medium (limited by equipment
security:	unknown
connection costs:	middle to high
access points:	almost anywhere on the globe
distribution:	high
devices:	Special equipment, satellite dish

2.5 Future Trends

Mobile Wireless LANs Internet Connectivity will be available shortly everywhere in populated areas on the world and increase strongly. While Wireless LANs will be mostly used by white collar workers, due to high bandwidth, cheap infrastructure and the concentration of these people at certain places, like restaurants, hotels, conference rooms and airports. Mobile phone based Internet access will become important for blue collar workers. They do not need too much bandwidth for data exchange at their field work, but they need it almost everywhere. Furthermore, the consumer market for mobile phone based Internet access has a huge potential. Especially with 3 generation mobile phones, but the start of these services will be delayed due to high investments and a too low interest on the consumer side.

2.6 Secure Mobile IP

With the change of Internet access points while traveling around there arise mobility problems. Every time the access point is changed from one sub-net to another, a new IP address has to be used. Most of the time this new IP address is issued from a DHCP Server. With the change of the IP address most applications get problems, because the TCP Connection hat to be rebuilt. Also problems arise if the mobile computer has a server, which will not be accessible anymore, or if the user wants to access the home network from outside, the firewall will block him. So the user has no access to his file-servers or to his e-mail from outside. If the user could access his home network, this connection should be secure, to prevent unauthorized access to data. Most mobile Internet access technology provide some security, but some of them are not reliable, also they are not providing an end-point to end-point secure connection, because the traffic on the Internet can always be sniffed.

A solution of this mobility problems is Secure Mobile IP (SecMIP) [12], known and referred to in this paper as Portable Office, too. SecMIP is a combination of MobileIP and IP Security (IPSec). The MobileIP tunnel is secured inside an IPSec tunnel. This allows the user of SecMIP a secure connection to his home network while out of office, using all of his services there and change seamlessly access points throughout his travel. Thus SecMIP is named the Portable Office, too. The Portable Office also includes improvements to Mobile IP , specially that the Foreign Agent can be located on the mobile device, allowing it to be used everywhere.

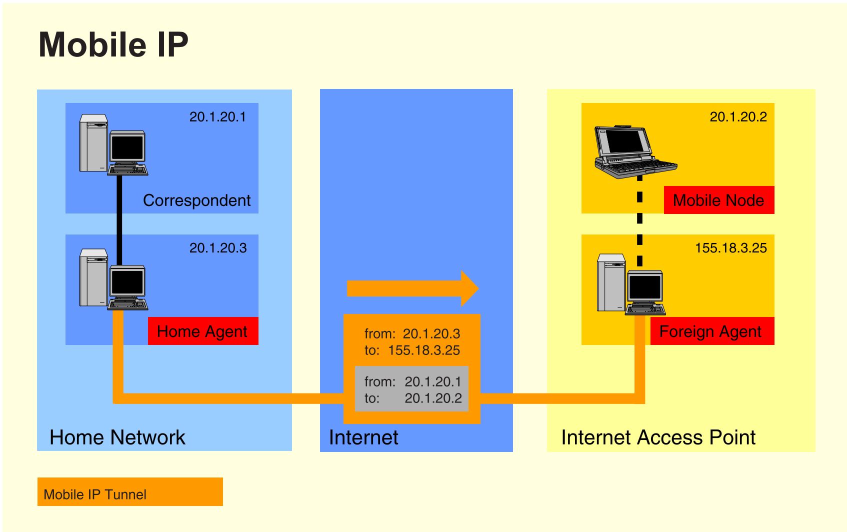

Figure 1: Mobile IP

2.6.1 Mobile IP

Mobile IP[11] is a Software that allows a portable device to keep always the same IP number (e.g. out of the address range of the home network). This is done with an IP-in-IP Tunnel between a Foreign Agent, located at the access point of the portable device and a Home Agent, located inside the home network. The Foreign Agent sends all packets from the portable device directly through the tunnel to the Home Agent, which forwards them to the home network under the static IP address of the portable device (see Figure 1). So receivers get the impression that the mobile device is always inside the home network. On the other hand all packets addressed to the static address of the portable device are picked up from the Home Agent and sent to the Foreign Agent, who then forwards them to the portable device under the originators IP address. To do this work, there has to be a Foreign Agent located at every access point the user wants to use.

The Portable Office proposes a solution which also includes encryption (see below) It includes the Foreign Agent on the mobile device. This allows to use Mobile IP from every access point. In this case the portable device has then two IP addresses, the static one that stays always the same, which the applications of it, his operating system and the correspondents in and outside of the home network can see, and the real IP Number of the Mobile Node, which changes dynamically (issued from DHCP Server's of the access points) and only the Foreign Agent and the Home Agent has to know. Mobile IP allows the user to access files and mail servers from outside of the home network and it allows a seamless change of access points.

2.6.2 IP Security

IP Security (IPSec)[12] is a set of extensions to the IP protocol family and has been deployed to implement VPNs. IPSec secures a connection on the Network layer, the layer where IP is used, thus the name IPSec. The services IPsec allows for are authentication, integrity, access control, and confidentiality. IPSec encrypts all IP Packets, optionally even the IP Headers. The encryption is based on a public/private key pair. IPSec eases building secure virtual private networks. The combination of IPSec and Mobile IP allows the generation of mobile virtual private networks.

2.7 Service Level Agreements

If a buyer buys something there is normally a written or oral contract between him and the seller. For Internet access this contract would be normally about price, per time or data volume, and bandwidth. This is common between home users and ISPs. But networks are shared, if there is more traffic, bandwidth is split up and delay and packet loss becomes bigger, too. Also overall network quality greatly varies. For home users the bottlenecks are mostly the last miles from the ISP to them, and therefore if the ISP has enough capacity the user will most of the time get a certain bandwidth with a certain quality. But there is no guarantee. Network traffic increases, and companies depend highly on networks. A lot of companies let other companies run their applications (Application Service Providers ASP) to save money, and they communicate with these applications over networks. Furthermore, some applications need a certain quality to make sense, e.g. a video conference, therefore companies demand from network providers certain network performances, so they can operate their business without problems.

These contracts are called a Service Level Agreement[14] (SLA). SLAs are not only over network services, but also used for all kinds of IT services, so enterprises, that use outsourced services, rely on them to guarantee specific levels of functionality, network bandwidth and uptime. A SLA details all the responsibilities of an IT service provider, the rights of the service provider's user, and the penalties incurred when the service provider violates any item of the SLA. Thus, an SLA identifies and defines the service offer itself, supported products, evaluation criteria and Quality of Service[15] (QoS) that customers can expect. Network SLAs cover the characteristics of the network itself, connection characteristics and network security. An important part in a network SLA is quality of service. Quality of service means delivering consistent predictable data delivery service. There are five main characteristics that qualify QoS[15]:

- Latency: The delay (measured in milliseconds) in a transmission path or in a device within a transmission path is called latency. For most applications some latency is not a big problem. Delay-sensitive applications are real time voice and video, interactive games etc.

- Jitter: When packets do not arrive at their destination in consecutive order or on a timely basis, i.e. they vary in latency, it is called jitter. Too much jitter is a problem for video or audio transmissions over a network.

- Bandwidth: Bandwidth is a measure of data transmission capacity, usually expressed in bits per second. Bandwidth indicates the theoretical maximum capacity of a connection. Video applications and large files transfers demand a high bandwidth.
- Packet Loss: The amount of packets that are lost during a transmission is called packet loss. Packet loss is measured in percentage of all data packets transmitted over a certain time, e.g. 1% over a month.
- Availability: The average availability of network services during a certain amount of time, e.g. 99.9% per month, is called availability . Another name for availability is called the uptime of a network, too.

Different applications need different QoS requirements. In the Internet with its "best effort" there is no Quality of Service guarantee. There are some technologies to provide QoS, but they demand all network layers from top-to-bottom, as well as every network element from end-to-end. Any QoS assurance are only as good as the weakest link in the "chain" between sender and receiver. There has to be some monitoring of QoS to check if the defined values of the SLA are fulfilled, too.

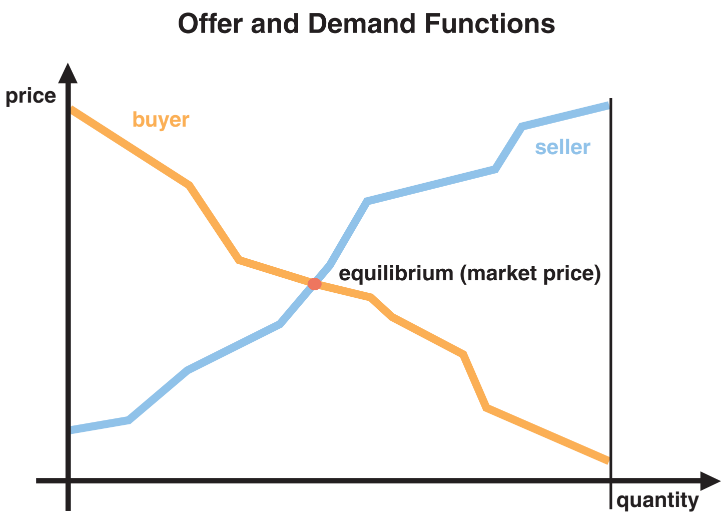

Figure 2: Offer and Demand Functions

3 Marketplaces

3.1 Definition

A marketplace[16] is commonly considered a place where a certain time offer and demand meet, the information-exchange between participants, the formation of a price, the trade to the market-price and the exclusion of sellers whose price-expectations are above and buyers whose price-expectations are below the market-price takes place.

3.2 Ideal Market

There is a model of an ideal market[16] in economics.

- homogeneity of goods: All goods in an ideal market have the same quality. To a good with 3 qualities we refer with 3 markets. In an ideal market all goods are equal, and therefore a market price for all goods in this market can be determined. In an ideal market therefore it does not matter what good to buy from which seller. An example could be the money exchange market.

- free business competition: In an ideal market everybody has access at any time, and there are no restrictions like taxes or bureaucracy. Furthermore, no agreements or mergers like cartels, cooperatives etc are allowed. There has also to be a free choice between different sellers and buyers, so no monopolies can arise and influence the price in a negative way for one side.

- market transparency: In an ideal market there is a complete information exchange between all participants, so all participants have the same knowledge about the goods, the environment, the prices, deals etc, so no participant can profit from a head start in knowledge.

- point market: The market is at one location, so the negotiation can be done on place and the exchange items versus money can be executed immediately without the need for transport.

- point of time: The ideal market is a snapshot, meaning that all deals are made immediately and the price is adjusted dynamically.

Under the assumption of 1 good = 1 quality the expectations of the participants are reduced to price and quantum of the now quasi standardized goods. Thus the offer and the demand functions can be described in a diagram (see Figure 2).

The offer function shows the quantity of an item the sellers are willing to sell at what price. If the price is high, all sellers want to sell their items, but the lower the price goes, the less sellers are willing to sell, and if the price is zero, at the latest, nobody will sell anything.

The demand function shows the interest of the buyers. If the price is low, more buyers are willing to buy an item. If the price is high, only a few buyers are willing to buy, and if the price reached a certain limit, nobody buys the item anymore.

The interception point of this two functions shows, what will be traded, that is how many items at what price. If the conditions change, also the offer and demand functions change and a new intercept point is defined. So every market tends towards an equilibrium. If the market is not ideal, i.e. one of the five points mentioned above is not fulfilled, this equilibrium can be more or less biased.

3.3 Variations from the Ideal Market

In real life there are no ideal markets, some are closer to the ideal market, like the stock market exchange, other are father away.

3.3.1 No Homogeneity of Goods

Only mass produced goods without differentiation like screws or computer chips are traded on markets fulfilling this point. A lot of agricultural goods and natural resources can be considered as homogeneous, if they meet an average quality, too. But most products, specially those in B2C (Business to Customer) markets, are differentiated and sold at different prices. These markets are split up in market segments.

3.3.2 No Free Business Competition

At some markets there is only one buyer or/and seller. This is called a monopoly. The monopolist sets the price so that he can maximize his profits. A monopoly generates normally prices above (or under) normal market prices. Some participants arrange themselves to influence the market price to their advantage.

An extreme form of this is a cartel. Most states try to prevent monopolies and cartels with laws. Although there are some accepted monopolies, mostly operated by the states themselves, like nets (public transportation, gas, water etc), generally markets with high capital invests. There are other market interventions by states, to protect consumers from high prices, to protect producers from ruin or to earn money. They are either market conformable, that means they use market mechanism and produce a new price (subventions, value added tax) or market non conformable , they influence the price directly from outside (minimum prices, maximum prices, fix prices).

3.3.3 No Transparency

A lot of times there is no free information flow. Information retention is commonly used to get better deals out of negotiations. In some auction types the offers are secret. It is an important business decision what information to restrain and what information to give away.

3.3.4 No Point Market

If a market is not limited to one location, transportation of goods and communication between participators must be taken into consideration. Examples are chains or a virtual market over the Internet.

3.3.5 No Snapshot

Most trading is done within a short time. But some business negotiation or some auctions (especially over the Internet) may last up to a week or even more.

3.4 The Modern Economy

Our economy is highly complex. There is a market for everything. Goods, services, finances, real estate, labor etc can be traded. The model of the ideal market demands homogeneity of goods, but most of the times similar goods from different offers are not exactly the same, so they have to be grouped together to create reasonable markets. What a market is is sometimes clear, like the market for certain natural resources, but mostly, specially by heavily varying goods, it is a question of definition or interpretation. Markets can be split up or they can be grouped together. In Figure 3 there is a scheme of the modern economy [17].

3.5 Administration, Business and Consumer Markets

Specially in electronic markets there is a popular grouping of markets. All possible market participants are grouped into three groups: administration (A) (government etc), business (B) and consumer (C). Each group builds with each other group and itself an own market. So there is an administration-to-administration market (A2A), an administration-to-business market (A2B) etc, 9 markets in total. The first letter tells the seller or the offer, and the second letter the buyer or the demand. So the A2B market would be e.g. the government supplies roads for companies, and they have to pay taxes for that. The two biggest markets are the B2B markets, meaning all the trade between two businesses, and the B2C market, the mass consumer markets, selling business goods and services

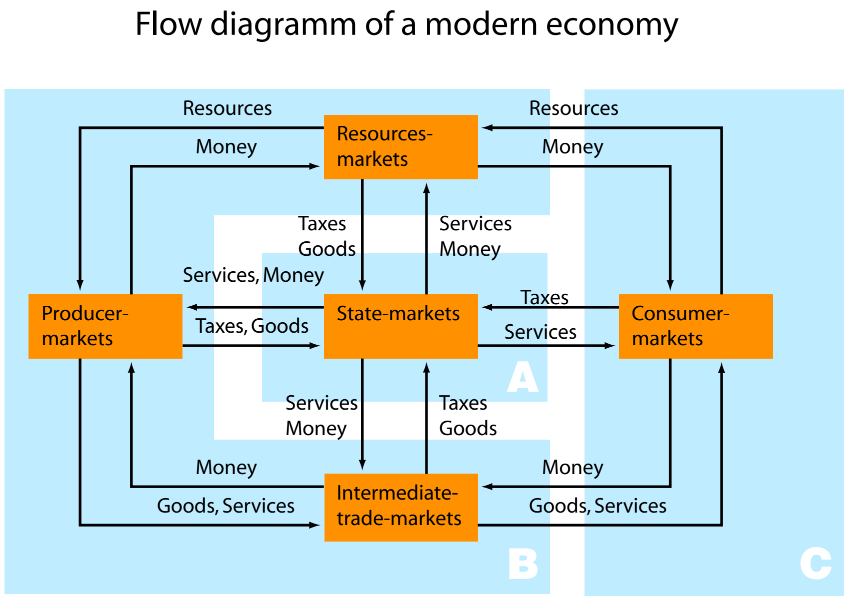

Figure 3: The Modern Economy

to consumers. These schemes can be easily put over the scheme of the modern economy above (see Figure 3).

3.6 Price Determination

To make a deal both parties have to agree on a price for a certain good. There are different methods to determine a price. Some are simple, some are complex, some are iterative, some are not, some are determined for each deal, others stay the same for all consumers. They all have in common that they try to find the equilibrium between demand and offer. The five most common ones are listed below (See Figure 4).

3.6.1 Price Negotiation

Price Negotiations are probably the oldest form to determine a price. Two parties try to find a for both sides acceptable price, mostly making offers and counter-offers. Price negotiating is widespread in some cultures (bargaining). Today price negotiations are used mainly in business to business deals. Most of the time not only the price is negotiated, but also other parts of the contract. Demands a lot of skills and knowledge from the negotiators. A good strategy and the right tactical decisions have to be made. A negotiation is made between two parties, although a party can negotiate simultaneously with different parties and choose the best deal in the end. In comparison to the other procedures very time-consuming, and therefore used only for important deals.

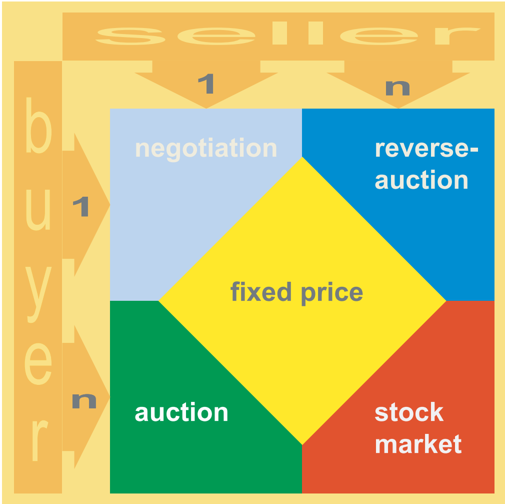

Figure 4: Price Determination

3.6.2 Auctions

A widespread form are auctions[18]. A good is sold from one seller to the highest bidder (s). In this form the bidder(s) make(s) the price. Auctions are used mostly if there is much more demand than supply. Furthermore, auctions are appropriate if the seller does not know how much his good is worth, or when the good has to be sold quickly. Auctions are used for agricultural goods, very unique goods (e.g. art) or for liquidations. Auctions go on until no bidders is willing to offer more. Some auctions have a fixed end time, e.g. auctions over the Internet or for written ones. Auctions are only about the price. The four most popular auction formats are listed below.

English Auction The English auction[18] is probably the most famous auction format, mostly the term auction is even associated with the English Auction. The English Auction is known as open-outcry or ascending-price auction, too. It is commonly used to sell art, wine and numerous other goods. In the English auction the auctioneer begins with the lowest acceptable price, or a very low price, and proceeds to solicit successively higher bids from the bidders until no one will increase the bid. The item is sold to the highest bidder. In some auctions there is a minimum price fixed for every item, such that the item is only sold if this price is reached. It is called reserve price. The reserve price can be known or unknown to the bidders.

The English auction is very simple, but it has some drawbacks, too. On the buyer side sometimes bidders pay more than the product is worth or more than they would have been willing to pay, but during the auction they got carried away and the competitive side of the auction let them continue with bidding. This is called the winners curse, the auction becomes a game. On the seller side a disadvantage is, that if there is not a high interest in an item it will be sold way beyond market value, and the seller does not receive maximum value.

Another problem to English auctions is, that they are vulnerable to rings. Rings are bidders that arrange themselves to get lower prices.

Dutch Auction Another auction format is the Dutch auction[18], getting its name from the Dutch flower markets. In a Dutch auction bidding starts at an extremely high value and is progressively lowered until a buyer accepts the price. When multiple items are auctioned, the price continues to sink until all items are gone. This form is used both on financial markets and for agricultural goods like flowers or fish. Some goods might get less in a Dutch auction than in an English auction, because the competitive side is missing. On the other hand, if a bidder is really interested in an item, he cannot afford to wait to long to enter his bid, thus he might bid close to his highest valuation.

First-Price, Sealed Bid This auction, also called a discriminatory auction[18], is a sealed auction, in contrary to the open outcry auction, thus the bids are hidden from the other bidders. In a first phase the bids are submitted, in a second period the winner is determined. If one item is auctioned, the highest bidder gets it and has to pay his bid. If multiple items are being auctioned, the highest bids win, every winning bidder has to pay exactly the amount he bid, thus it is called a discriminatory auction, too. This type is used for refinancing credits and foreign exchange. In the financial community this auction is called an English auction!

Uniform Second-Price, Sealed Bid The uniform second-price auction is also called the Vickrey Auction[18]. Like the first-price auction, the bids are sealed, and therefore unknown to the other bidders. The highest bid wins, but the bidder has only to pay an amount equal to the second-highest bid at a single item auction, or the first unsuccessful bid at a multiple item auction. Therefore at multiple item auctions, all bidders pay the same price. At a first glance, it seems that this auction format tends to make less money than at a first price auction, but some theories try to prove the contrary. The Vickrey Auction is used to refinance credit and foreign exchange, too. In the financial world it is known as a Dutch Auction!

3.6.3 Reverse-Auctions

A reverse auction is, like its name suggests, an auction, only the other way around. Some sellers offer their items, and the best offer wins the bid. A reverse-auction is most of the time not only about the price, but also other parts of the deal. Reverse auctions are commonly used on the construction and project market, e.g. which architect can realize a project. All four basic auction types can be reversed and held as a reverse auction, but only the first price sealed bid reverse auction and modifications of the English auction play a role.

3.6.4 Stock Market

This is the implementation of the ideal market. Lots of sellers and buyers determine the market price of a homogenous good at a certain place to a certain time. The price is highly variable. A single participant has no direct influence on the price, he can only accept it or not. Goods that are traded this way are like the name says stocks, but there are other homogenous goods sold this way from agricultural goods to natural resources to computer chips. The stock market exchange can also be viewed as an auction, the continuous double auction.

3.6.5 Fixed Price

Instead of negotiating a price the seller can just determine a price and the buyer can accept it or not. This is mostly used in widespread markets with lots of participants and complex products, where the price of the product is not the only decision factor. Most business to consumer markets are organized like this. The price is not really fix, because if the items does not sell or could be sold at a higher price, the price can be adapted. But the changes are slow, and the price is not determined newly for every deal, and therefore stays the same for a certain time at a certain place, for all buyers, thus providing a certain fairness. The conclusion of a deal is a lot more simpler than the four other price determination procedures, because nothing has to be negotiated, but determining the price itself in advance is a highly complex task, where a lot of factors play a role.

The price is a complex mixture of production costs (labor, real estate), costs for materials and products for producing the goods and the distribution costs. But in the end it is mostly a price generated by the marketing department, the price they believe buyers are willing to pay for a certain product. Depending on the reaction of the market, the price can be adapted afterwards. E.g. the price of a packet of coffee. First the coffee beans have to be bought at exchange markets. This costs can vary greatly. Then there are fixed cost for transporting and treating the beans. In the end the coffee may be sold under two different labels with two different prices.

3.7 Price versus other Decision Factors

In an ideal market the price is the only decision factor for buyers. But in all other markets where the goods or services are not of the same quality a lot of other factors play an important role. This may include all aspects of a good or service, like features and extras, but also quality, the look or design of a good, added features or services, guarantee regulations, availability, brand and images of that brand (see the section below, too). Sometimes the decisions are not rational at all and subconscious and psychological factors (emotions, experiences, other people's opinion, stereotypes, commercials, packaging, etc) may play an important role. An simple task like buying a toothbrush may become very complex if it is split up and analyzed. Sometimes the price does not play a role at all, or an item is only bought because the price of it is much higher than the competition, like luxury articles.

3.8 Branding

An important decision factor for buying items are brands and labels. Historically brands were established for a guarantee of quality. First only oral from person to person and on local markets, e.g. the knives of smith Ironhand are the nicest, later with the industrialization and mass distribution, the manufacturers wrote their name on the products. So if a company has established a good name, mostly for quality or superiority to other products, this helps them to sell more. Today some brands are worth more than the companies owning them, and are their main property. Even brands and labels are "rented" (licensed) to other companies, or completely sold to them. By mass distributing products, brands and products have to be known, so advertising has become important. With advertising images have been built up. Today a lot of goods are mostly sold over their image, e.g. if somebody buys a pair of Levi's jeans, they do not buy a pair of jeans but a cool image. This happens specially in fashion, but also for cigarettes, watches, cars etc. Brands and labels are mostly used for goods and services in high interest markets. Mostly on consumer markets, but they play an important role in business markets, too. But even in some low interest markets brands have become important, e.g. fruits like bananas are sold under brands, and even "brandless" products of chains became a brand itself, e.g. some supermarkets or computer-parts distributors could establish strong private brands.

3.9 Relationships

Another important tool for selling goods and services are relationships between sellers and buyers. This can be either on a personal basis from seller to buyer or on a more abstract customer to company basis. So establishing and maintaining a good relation is also an important decision factor. Relationship includes different aspects:

- Good experiences with a product or service of a company, trust.
- Impressions at the customer-front, friendly and competent sellers or employees, good customer service
- Personal relationship with a seller or employee of a company (we trust people we know and are fond of, e.g. car dealer, assurance, shop employee, sales man)
- Long term thinking (not such a good deal today, but some paybacks later) e.g. b2b negotiations
- Discount for good customers, so relationships pay off

3.10 Electronic Marketplaces

An Electronic Marketplace[19] (eMarket) describes a virtual platform on the Internet through which the parties in a market communicate, exchange ideas, realize marketing, handle transactions, manage stocks etc.

- thousands exists already
- "everything commercial" on the Internet is an eMarket...
- range from simple yellow pages to stores to auctions and reverse auctions
- 80 percent B2B!

3.11 Automated Marketplaces

An automated marketplace is a fully automatic eMarketplace. The users tell their offer or demand, the rest (including trade decision) is done automatically (except transportation of goods). There can be made a distinction between automatic marketplaces for optimization problems and for competitive markets. Marketplaces for optimization problems try to get an equilibrium in changing environments with the help of market mechanisms. That could be finding the optimal schedule for trains or an air-conditioning system in big buildings etc. Such systems are relatively simple to implement. There are a lot of such systems, mostly on a research stage, but also already fully implemented. Competitive automated markets in contrary, where different parties all try to get the maximum out of it, are much more difficult to implement and far away from usable. A exception may be the automated stock market. Here computers take over the matchmaking and calculate the stock rate, but when and what stocks to buy or sell still decide humans. Automatic competitive markets would most likely be used for goods of low interest, specially homogenous good, and in B2B markets. Most automated markets are realized with Multi Agent System (agent-based marketplaces).

3.11.1 Problems of Implementing Automated Markets

The main problems of realizing a marketplace:

- Technical:
  - Implementation (e.g. agent-based, what agent platform, software design)
  - Security aspects, authentication of participators
  - The system has to be robust and stable
  - Providing a secure service for accounting
- Design:
  - Market design (protocols, rules, defining items, participants, information exchange etc)
  - Strategies for sellers and bidders, decision functions
  - Providing fairness, consideration of fraud and foul play
  - Integration of branding and relationships
  - Mechanisms to prevent vigorous oscillation of market, like price wars etc.
- Marketing:
  - Convincing seller and buyers of advantages and security
  - Mayor market players have to participate, for the market to become interesting for buyers.

The Technical problems, which include all aspects of an implementation, are problems in the field of computer science. The marketing problems, thus, how to sell the marketplace itself to the buyers and sellers, have to be solved by business management and marketing specialists. The design of the marketplace, putting intelligence and clearly defined structures and sequences into it, is the most interesting part, but also the most challenging. This should be solved interdisciplinary by computer science, business management, mathematics and psychology specialists.

Some special problems arise with the nature of the immediate reaction of electronic systems to changes. While real markets are highly nested and complex and therefore react always slowly to changes and mostly do not react fully, automated markets react immediately and fully to changes. This can lead to unwanted effects like the immediate establishing of monopolies and freezing states or the market could get unstable and start to vigorously oscillate by competitors dropping in and out or endless cycles of price wars. Price wars happen when sellers drive each others prices down until one seller realizes that it can make more profit by raising its price again. He breaks the price war, but the other sellers are free to grab the lost consumers back by slightly undercutting the new higher price and the price war begins again. With lots of different sellers more than one price war might be running at once, and sellers may quit one just to enter another one.

4 Agents

One way to implement an automated marketplace is with agent technology[20]. But what is agent technology exactly, what is the definition of an agent? A good way to see agents is as an abstract design/model of software systems that are autonomous, communicative, intelligent and distributed.

Agents evolved out of a natural progression of trends in the IT world:

- More processing power: Computers are getting faster and more efficient, thus allowing for more complex software systems.
- Interconnection: Almost all computers are interconnected today, thus allowing distributed problem solving and information exchange.
- Communicability : Well defined communication protocols make communication between different computer systems possible.
- Delegation: More and more tasks are delegated to computers. This includes more and more white collar tasks, too.
- Intelligence: Software is getting more intelligent, thus more complex task can be delegated to computers.
- Human oriented view: More and more problems are tried to be solved with a human oriented view instead of an algorithm or an object oriented view.

4.1 Human Oriented View

Humans solve problems by interacting with one another and working towards meeting a general consensus. This idea is used to model the world with interacting software entities, which are called agents. This allows to develop complex interacting software systems in a very intuitive manner. Agents can be everywhere. For example if a business man wants to make a travel, he tells his secretary his plans, who then calls a travel agency to reserve a flight and so on. Modeling that, there would be a business man agent, a secretary agent, a travel office agent and so on. As this example shows, agents represent not only entities of individuals, they can also represent entities like organizations, governments, enterprises etc.

4.2 Definition of Agents

There exits no clear definition of agents. But most agree on some common features[20]. An agent hast to be:

- Autonomous:capable of acting independently, to control internal states, capable of action in some environment.
- Reactive: to react to changes in the environment.
- Proactive: setting and achieving goals.
- Social: ability to interact with other agents.

Agents can have other features like veracity, benevolence, rationality, learning-adaptation etc. Most of these features are aimed to make an agent more human like. In some definitions, an agent just has to be autonomous, e.g. a thermostat is an agent, too. If such an agent is also proactive, reactive and social, it is called an Intelligent Agent. The absolutely correct name for a software entity described above would be Intelligent Software Agent. In this document simply the term agent is used, meaning an Intelligent Software Agent. To make sense, agents seldomly exists alone. A system consisting of more than one agent is called a Multi Agent System (MAS), and if they are distributed, that means located on different computers, a Distributed Multi Agent System.

4.2.1 Agent Mobility

Another possible feature of an agent is mobility. Mobile Agents can change platforms and migrate to other computers with their code. There they then act autonomously (within certain boundaries). For some experts this is a much promising feature, but others question if it is really necessary to have mobile code, because agents can easily stay on their platform and just exchange information between different platforms. Agent mobility demands more security and stronger authentication. The agent world is split into two groups, one favoring Mobile Agents, others static Intelligent Software Agents. Therefore there are two main standards for agents (more about these standards and the differences between mobile agents and static agents below in the section about agent standards). Although there are some efforts to bring the two standards together.

4.3 Agents vs. Object Oriented Programming

Agent technology uses a higher level of abstraction than traditional object oriented design. [20, 21, 22] But:

- agent technology has (yet) no own programming language
- most of the time object oriented languages are used to develop agents and multi agent systems.
- Some open source agent platforms are implemented, allowing fast development of multi agent systems.

The main differences between Agent Technology and Object Oriented Languages are:

- Objects resemble objects, while agent technology is oriented on humans.
- Objects communicate via message parsing, methods are started via invoked messages. Objects encapsulate some state.
- Agents communicate with an agent communication language (ACL). They decouple communication from action and have autonomy over their behavior as well as their state. They engage in dialogs and may negotiate and coordinate the transfer of information.
- Agent have continuous active threads of control

- Agents are goal driven, reactive, have beliefs, desires, they can judge their results and change their behavior. They reason during the process of selecting a possible action
- Objects have no problem solving ability in the sense of agents.
- "Objects do it for free, agents for money"

4.4 Agent Technology vs. Expert Systems

An expert system has some knowledge about something. The main differences between agent technology are:

- agents are situated in an environment
- Agents act
- Agents interact with each other

But some real time expert systems are agents.

4.5 Building Agent Systems

The main issues in building agent systems can be split up into two parts, the micro and the macro issues:

- Micro issues: These issues deal with how to build an agent. An agent can be build like a touring machine. There is an input, a control structure consisting of several layers, where the agent reasoning mechanism is located, and an output.
- Macro issues: These issues deal with how to build a society, how agents can cooperate effectively. It deals with inter-agent communication, message formats, interfaces etc. The macro issues demand standards. There are two standardization efforts for agents, FIPA and OMG MASIF.

4.6 Agent Standards

There are two main beliefs[23] for building agent systems, one for mobile agents and the other for static intelligent agents. Both directions have their own standardization efforts. The standard for mobile agents is OMG MASIF. Their idea is to enable mobile agents to migrate between agent systems of the same profile, that is language, system type, authentication type and serialization methods, via standardized CORBA IDL interfaces. Mobile agents focus on mobility of program codes together with their states among network sites. FIPA, the standardization effort for static intelligent agents, makes the agents interoperate via the standardized Agent Communication Language (ACL), the content language and the ontology which identifies the set of basic concepts used in the message content for cooperation. The following points are the main differences between the two directions[23]:

- Efficiency: In mobile agent systems the messages sent around are program code in a low level programming language, while the intelligent agent systems send around their messages in an agent communication language and a predicate logic based content language. Intelligent agent messages usually take less time and transport capacity to migrate between source and destination sites. However, it is generally more efficient to execute a mobile agent due to its lower level implementation.

- Adaptability: With a high number of operations encapsulated and on-demand migrations, the mobile agent approach can help to dynamically adapt interfaces and services of remote systems, reduce dependency on the constant availability of underlying network connectivity, achieve dynamic load balance and enable dynamic distribution of functions. Mobile agents can sometimes be used to modify or replace remote applications or their components (autonomous software downloading and configuration) if the mobile agents and the remote applications are implemented in the same language. Intelligent agents usually do not directly support this kind of modification or replacement. But they can use the knowledge contained in the messages easily to integrate it into their own knowledge, making intelligent agent technology more appropriate for adapting intelligent agent interfaces and functionality.

- Syntactical Interoperability: Mobile agents require homogeneous platforms for interoperability, while the intelligent agent paradigm supports the interoperability among heterogeneous environments.

- Richness of Interaction Protocols: ACL can provide a richer set of semantically standardized interactions between static software systems than the mobile agent paradigm, where move and receive agents is the operation being standardized.

- Semantic Interoperability: The intelligent agent approach supports not only syntax-based interoperability but also interoperability based on semantics. This feature will be very useful for complex and dynamic cooperation problems.

- Binding AI alike Technologies: Agent communication, with its strong association with Artificial Intelligence (AI), can easily support the bindings of AI-like technologies into the individual static agents. This feature can further increase the flexibility, tolerance, robustness of the cooperation and negotiation among agents.

- Security: It is easier to analyze the behavior of an intelligent agent message. Therefore, the receiving intelligent agent can check the messages for subtle security and contract violations. Intelligent agents are therefore safer than mobile agents.

- Reliability: Agent communication paradigm and its languages can be more easily associated with a formal theory for agent interactions. This theory enables the formal analysis and verification of the global distributed systems and can further increase the reliability of agent-based applications.

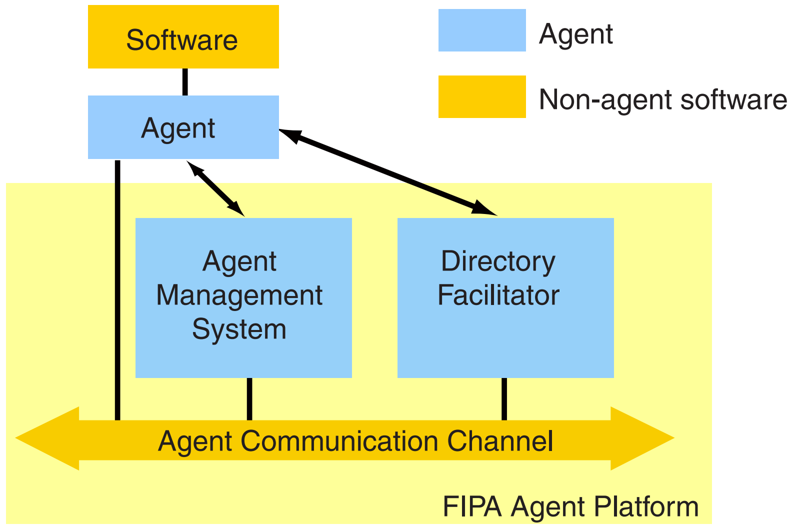

Figure 5: The FIPA Agent Platform

There are efforts to bring the two standards together e.g. the agent platform Grasshopper has implemented both of them, but mostly the integration and integration strategies for the two standards are still theoretical.

4.7 FIPA

The Foundation of Intelligent Physical Agents (FIPA)[24] is an organization for defining standards for Multi Agent Systems. Their initial vision was physical agents (robots), that is where the name comes from, but today they just deal with software agents. The key focus of FIPA is:

- Specifying communication and inter-operability between agents
- Specifying external behavior as opposed to internal behavior. That means it does not specify how agent process and reason about information they receive.
- Use in heterogeneous environments

In FIPA every agent is located on a platform. Platforms are then linked together. A platform consists of three main parts. The Agent Management System (AMS), whose main focus is the agent lifecycle management (management of platform, starting and deleting of agents, access control etc.), the Directory Facilitator (DF), which provides yellow pages services, and an agent communication channel (ACC), which enables agents to communicate with each other (see Figure 5).

The AMS and the DF are actually agents itself, too. Each new agent has to register at the AMS and the DF. It can get information about other agents from the DF and then contact them over the agent communication channel. These other agents can stay on the same platform or on another one, as long as

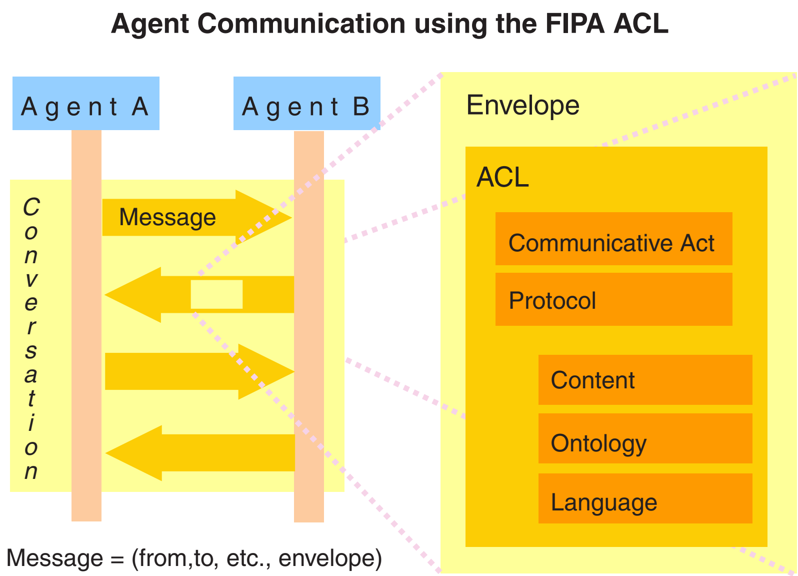

Figure 6: FIPA Agent Communication

the other platform is indirectly or directly linked together with this platform.

4.8 FIPA Agent Communication

The main part of FIPA is the definition of the communication between agents. So it should be possible that agents can communicate with each other, even if they are implemented in different environments. Two agents are communicating with each other in a conversation (see Figure 6). Each time a message is sent to the corresponding partner. A message consists of the addresses of the sender and the receiver and an envelope. This envelope is written in the FIPA Agent Communication Language and consists of a Communicative Act, a Protocol and the Content of the message.

The Communicative Act describes the basic intention of a agent message. FIPA Communicative Acts are basic types. They can be assertive like inform, refuse, failure etc, directive like request, query etc, commissive (promises) like agree etc or expressive (wishes) like subscribe etc. An agent can also ask for something and the recipient can agree or refuse.

The Communicative Acts of messages occur in patterns called conversation or dialog, and are defined in protocols (see Figure 7). There are 13 protocols defined now in FIPA, eg. FIPA-request, FIPA-cfp (call for proposal), FIPA-auction-English etc. In the FIPA Request protocol e.g. the initiator makes first a request. The participant can then answer with a not-understood, a refuse or an agree. In the case he agrees the participant has to send a failure or an inform, after that the conversation ends.

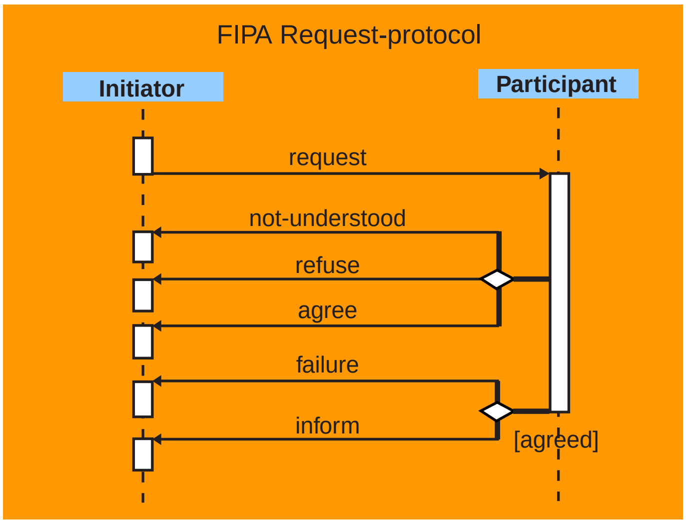

Figure 7: FIPA Request-Protocol

The message content consists of the content itself, an ontology and the content language. The content can be anything, text, an object etc. The ontology defines terms, the relationship between terms and operations for the content of a particular domain. The content language says in what language the content is written, e.g. sl0 (an agent language), English etc.

4.9 FIPA Platforms

Public available FIPA platforms:

- Agent Development Kit[25](Tryllian BV): The ADK is a mobile component-based development platform that allows to build reliable and scalable industrial strength applications. It uses a reliable and lightweight runtime environment based on Java.
- April Agent Platform[26](Jonathan Dale and Johnny Knottenbelt): The April Agent Platform (AAP) is a FIPA-compliant agent platform that is designed to be a lightweight and powerful solution for developing agent-based systems. It is written using the April programming language and the InterAgent Communication System (IMC).
- Comtec Agent Platform[27](Information-Technology Promotion Agency, Japan and Communication Technologies): Comtec Agent Platform is an open-source, Java based, free implementation of FIPA agent communication, agent management, agent message transport and some of the applications. Unique to the Comtec Platform is the implementation of FIPA Ontology Service and Agent/Software Integration, which require SL2 as the content language. Comtec Agent Platform is based on FIPA 97 and 98 specifications, which are now obsolete.

- FIPA-OS[28](Emporphia): FIPA-OS was the first Open Source implementation of the FIPA standard and has already recorded thousands of downloads. Dedicated developers from around the world have contributed to numerous bug fixes and upgrades, leading to over 10 formal new releases. FIPA-OS now supports most of the FIPA experimental specifications currently under development. With the new in-depth developer's guides, it is an ideal starting point for any agent developer wishing to benefit from FIPA technology. FIPA-OS is a component-based toolkit implemented in 100% pure Java. One of the most significant contributions recently is a small-footprint version of FIPA-OS (FIPA-OS), aimed at PDAs and smart mobile phones, which has been developed by the University of Helsinki as part of the IST project Crumpet.

- Grasshopper[29](Germany): Grasshopper is an open 100% Java-based mobile intelligent agent platform, which is compliant to both available international agent standards, namely the OMG MASIF and FIPA specifications. Grasshopper includes two optional open source extensions providing the OMG MASIF and FIPA standard interfaces for agent/platform interoperability.

- Jack[30](The Agent Oriented Software Group): JACK Intelligent Agents, is an environment for building, running and integrating commercial-grade multi-agent systems using a component-based approach. JACK is based upon the company's Research and Development work on software agent technologies. The JACK Agent Language is a programming language that extends Java with agent-oriented concepts.

- JADE[31](TILAB, formerly CSELT): JADE simplifies the development of multi-agent applications, which comply with the latest FIPA 2000 specifications. While appearing as a single entity to the outside world, a JADE agent platform can be distributed over several hosts. Agents can also migrate or clone themselves to other hosts of the platform, regardless of the OS. The life cycle of agents can be remotely controlled via a GUI, which also allows debugging tools to be started. The communication architecture tries to offer (agent transparent) flexible and efficient messaging by choosing, on an as-needed-basis, the best of the FIPA-compliant Message Transport Protocols (MTP) that are activated at platform run time. JADE is implemented in version 1.2 of JAVA and has no further dependency on third-party software.

- Java Agent Services[32](Fujitsu, Sun, IBM, HP, Spawar, InterX, Institute of Human and Machine Cognition, Comtec, Verizon) The Java Agent Services (JAS) project defines an industry standard specification and API for the deployment of agent platform-service infrastructures. It is an implementation of the FIPA Abstract Architecture within the Java Community Process [www.jcp.org] initiative and is intended to form the basis for creating commercial grade applications based on FIPA specifications. Specifically, the project consists of a Java API (in the javax.agent namespace) for deploying open platform architectures that support the plug-in of third-party platform service technology. The API provides interfaces for message creation, message encoding, message transport, directory and naming. This design is intended to ensure that a JAS based system deployment remains transparent to shifts in the underlying technology without causing interruption to service delivery and therefore the business process.

- LEAP[33](Fr): LEAP (Lightweight Extensible Agent Platform) is a development and run-time environment for Intelligent Agents, is the precursor of the second generation of FIPA compliant platforms. It represents a major technical challenge - it aims to become the first integrated agent development environment capable of generating agent applications in the ZEUS environment and executing them on run-time environments derived from JADE, implemented over a large family of devices (computers, PDA and mobile phones) and communication mechanisms (TCP/IP, WAP). In this way LEAP benefits from the advanced design-time features of Zeus and the lightweight and extensible properties of JADE.

- ZEUS[34](BT Labs): ZEUS is an Open Source agent system entirely implemented in Java, developed by BT Labs and can be considered a toolkit for constructing collaborative multi-agent applications. Zeus provides support for generic agent functionality and has sophisticated support for the planning and scheduling of an agent's actions. Moreover, Zeus provides facilities for supporting agent communication using FIPA ACL as the message transport and TCP/IP sockets as the delivery mechanism. Zeus provides facilities for building agents in a visual environment and support for redirecting agent behavior, too. The Zeus approach to planning and scheduling involves representing goals and actions using descriptions that include the resources they require and the pre-conditions they need to be met in order to function. This allows goals to be represented using a chain of actions that have to be fulfilled before the goal can be met. This action chain is built up using a process of backwards chaining.

4.10 OMG MASIF

The Object Management Group(OMG)[35] was founded in 1989 by 11 companies including 3Com, HP, Canon, Sun, Unisys and American Airlines. Today it includes more than 800 members. It is a not-for-profit corporation formed to produce and maintain specifications for interoperable enterprize applications. Conformance to these specifications will make it possible to develop a heterogeneous computing environment across all major hardware platforms and operating systems. Its most famous standard is CORBA. In 1995 the OMG started working on a standard called Mobile Agent Facility (MAF), in order to promote interoperability among agent platforms. Later the standards name was changed to to Mobile Agent System Interoperability Facility[36](MASIF).

The MASIF identifies a Distributed Agent Environment (DAE) and a Distributed Processing Environment (DPE). In a DAE, there are the following elements:

- Place: A place is a context in which an agent can execute, so a place is an execution environment.

- Agency: An agency is an agent system. An agency can have several places.

An agent system represents a platform that can create, interpret, execute, and transfer agents.

- Region: A region is a group of agencies that belong to a single authority.

Two interfaces represent the core of the MASIF standard. The MAFAgentSystem is associated with every agency and provides operations for the management and transfer of agents. The MAFFinder: It is associated with a region. It supports localization of agents, agencies, and places in the scope of a region.

The following agent functionalities are covered by MASIF:

- Agent management: This comprises the creation, termination, suspension, and resumption of agents. The MAFAgentSystem provides several methods for this purpose.
- Agent tracking: Agencies, places and agents are registered in a region registration component via MAFFinder.
- Agent transport: MAFAgentSystem offers two methods to support agent migration.
- Agent and agency naming: Standardized syntax and semantics of agent and agency names enable agents and agencies to identify each other and allow clients to identify agents and agencies.
- Agent type and location syntax: Agency types provide information about important aspects of specific agencies, such as the used implementation language. The location is standardized in order to enable to locate each other.

MASIF relies on CORBA to handle agent security. MASIF does not address the agent communication aspect.

4.11 MASIF Platforms

- AGLET[37]: IBM's mobile agent platform is implemented in Java. An Aglets is actually a Java object that can move from one host on the Internet to another. That is, an aglet that executes on one host can suddenly halt execution, dispatch itself to a remote host, and resume execution there. When the aglet moves, it takes along its program code as well as its data. The hosts need a ATP server to receive aglets.
- Concordia[38]: This is Mitsubishi Electric's Mobile Agent Environment. Concordia is a full-featured framework for the development and management of network-efficient mobile agent applications which extend to any device supporting Java. Concordia is written in Java and is portable to any platform running Java.
- Grasshopper[29] (Germany): Grasshopper is an open 100% Java-based mobile intelligent agent platform, which is compliant to both available international agent standards, namely the OMG MASIF and FIPA specifications. Grasshopper includes two optional open source extensions providing the OMG MASIF and FIPA standad interfaces for agent/platform interoperability.

- Kafka[39]: Kafka is an agent library designed for constructing multi-agent-based distributed applications from Fujitsu. Kafka is a flexible, extendable, and easy-to-use Java class library for programmers who are familiar with distributed programming. It is based on Java's RMI. Kafka is now integrated together with Pathwalker, a process-oriented programming library.

5 Related Work on Agent-Based Marketplaces and W-LAN Support Software

5.1 Agent-Based Marketplaces

Work related to this diploma thesis in the field of agents can be found on the one hand at agent-based solutions to network management and provider selection like the Shuffle project or on the other hand at agent-based marketplaces.

5.1.1 Shuffle

Shuffle[40] is an agent-based approach to controlling resources in UMTS networks. It is a project in the IST Program. Participants are European telecommunication companies like Portugal Telecom Inovao S.A. and Swisscom AG, universities like National Technical University of Athens and Queen Mary University of London and enterprises like Martel, Nortel and Emorphia. The project aims to create a novel architecture for efficient, scaleable and robust real time control of third generation mobile systems in the context of realistic business models of network providers, service providers and customers and the relationships between these actors. This goal is planned to be reached with intelligent software agents complying to the FIPA standard. A demonstrator was realized with FIPA-OS. This project together with the Portable Office Project, were the two starting points for this diploma thesis.

5.1.2 Sardine

Sardine[41] is a simulated airline ticket system based on an auction driven marketplace. It emphases on dynamic seller strategies in an auction driven marketplace and on their vision for interfaces that facilitate negotiation between buyers and sellers. The system supports multiple attributes and weights on the bidder side to help the bidder find their favored flights.

5.1.3 Other Agent-Based Marketplaces

There are a lot of papers about agent-based marketplaces. Most of these marketplaces are used to solve optimization problems, that is to find an equilibrium with the help of a market. Examples would be calculating a train schedule or to optimizing the air-conditioning in a large building. There are also works about agent-based marketplaces in a competitive environment, that is where every agent looks out just for himself and may therefore take actions against other agents, but these papers are mostly limited to a special case and are not suited for a more realistic case. A good information source for the current work in the field of agent-based marketplaces and intelligent agent systems in general is the annual book to the AAMAS conference[42] (Autonomous Agents and Multi-Agent Systems), formerly AGENTS - International Conference on Autonomous Agents, organized by ACM (Association for Computing Machinery).

5.1.4 Industry

It is difficult to say what the industry is doing right now with agents and marketplaces. Their implementations are naturally barred. Otherwise they try to sell products and claim therefore that they are made with "agents", which sounds great and advanced, but if such concepts are really behind it is often questionable. It can be said, that specially in the B2B markets, automated marketplaces are built, which may contain agent-like concepts. There is research in the field of agents and their use in agent-based marketplaces, too. This is mostly about simple marketplaces used for matchmaking and problem solving. A special interest exists in agent deployment into portable devices like cell phones. But generally agent technology is still in an emerging phase, therefore it will take some time to see agent-based market and agents in generally applied at large.

5.2 WLAN Access Support Software

Most WLAN access providers offer software to their customers that helps them in connecting to access points. There is a trend for user supporting, task delegated and intelligent software in this field. These approaches are not as explorative as the one of this project, but they show possible applications for agent technology in the future. Vicarious for this kind of software, the following two are presented.

5.2.1 The Portable Office

The Portable Office[13], as used in this projects demonstration, allows to change seamlessly access points and a secure connection to the home network. This is done with the combination of Mobile IP and IPSec. Furthermore, the developers are working on more intelligent features that lets the user configure the software so that for example it connects always over the cheapest technology or over the hot spot with the best connectivity.

5.2.2 BOINGO

In the same direction aims the free software from the world wide WLAN service provider Boingo[43]. This software includes a Wi-Fi sniffer that "sniffs" the airwaves for available commercial, private, and free wireless networks, an one-click connection to WLAN access points, a location directory to easily locate access points, the possibility of managing security keys and network settings for private wireless networks and a VPN, too.

5.3 Economics

Surprisingly, related work could not be found in the field of economics. Economic studies only engage in what is, but not what could be, maybe because of its empiric nature. E-commerce theory only looks at classical engagements in the field of electric markets. But there is a need for theories in the field of marketplaces that could be used to build one. Macro economics studies markets as a whole, which is not usable in realizing a single one, and fields in micro economics like marketing studies how markets can be grouped, analyzed and influenced, but not how they work and can be implemented. Therefore, the theory about markets in this document had to be derived by ourselves.

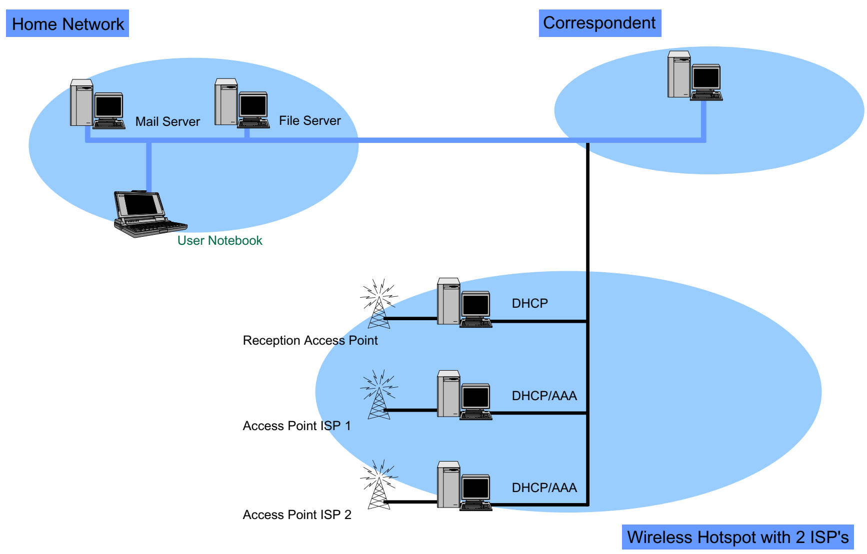

Figure 8: Layer 1: The Network

6 System Design

6.1 Overall Design

The goal of this diploma thesis is to realize a marketplace for temporally limited Internet access for mobile devices. To realize a market it needs a buyer, a seller and a marketplace entity (see next section about the software design). This entities run on different computers and are connected over the Internet. The seller entity, representing the user, should support the user as much as possible and manage the network connectivity of the portable device of the user on which it runs, too. This means it should change automatically over to the access point where it has bought access time from the seller (representing an ISP) at the desired times. The design should be adaptable to different connection technologies and more complex and dynamic situations. This has to be considered specially for the software design. For an overall design let us have a look at a really simple scenario. A user goes to a location where two ISPs are offering service at. There the user needs a connection to his home network. This connection should be secure and allow him to change access points without disruption. The parts of this design can be grouped into three layers.

6.1.1 Layer 1: The Network

The first layer is the network layer. It consists of the physical networks and computers and complies to the physical layer in computer network theory. The two main parts of this layer are the home network of the user with his computers, mail-server, file-server etc, and the hot spot location, where the user may go to and connect his portable device to the Internet, e.g. a Wireless-LAN hot spot.

At such an example hot spot there are 2 ISP offering access. Both ISPs have to

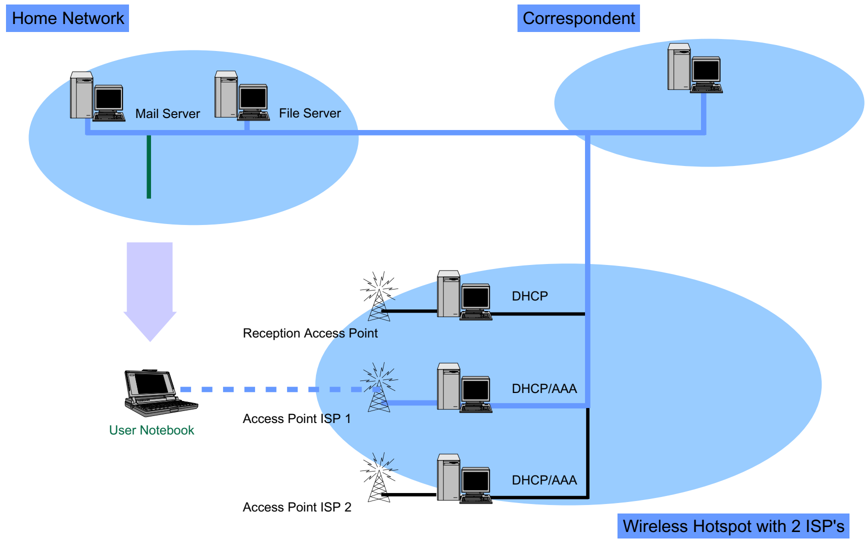

Figure 9: Layer 1: The Network, User Out-Of-Office

have an antenna and some kind of admission control and IP-address management. How the ISP solves this is not so important, as long as it is connected somehow to the other layer with the market on it. Thus, the entity representing the ISP can give the user entity the right configuration data for connecting after concluding a deal. IP-address management is most likely done with a DHCP server, but the ISP could also give an IP address directly to the user, and use this IP for admission control (e.g. this IP could only be used during a certain time). Other ways for admission control for wireless LAN could be to use a private ESSID and the use of a WEP-key, like realized in this project's demonstration version. In a more professional environment, the ISP would likely use some kind of a AAA Server (authentication, admission, accounting), maybe with a user name and a password or some code that would allow the user to use this access point for a certain period of time.

Another important part of the hot spot is the reception area. This is a public access point where users without a contract can connect to and conclude a contract with one of the ISPs to use their access points. This public access point has to has some kind of traffic limitations to prevent abuse. This could be achieved with a firewall or some other traffic control mechanism. It has to be emphasized that this reception point allows the use of the second-layer parts (Portable Office), but also works without, in both cases without neglecting possible abuses. Figure 8 shows the parts of the network, with the user located at his office and where he can use the Internet and his local services normally. Figure 9 shows the user out of office. He gets connectivity to the Internet and to his home network over an access point of an ISP at an hot spot location.

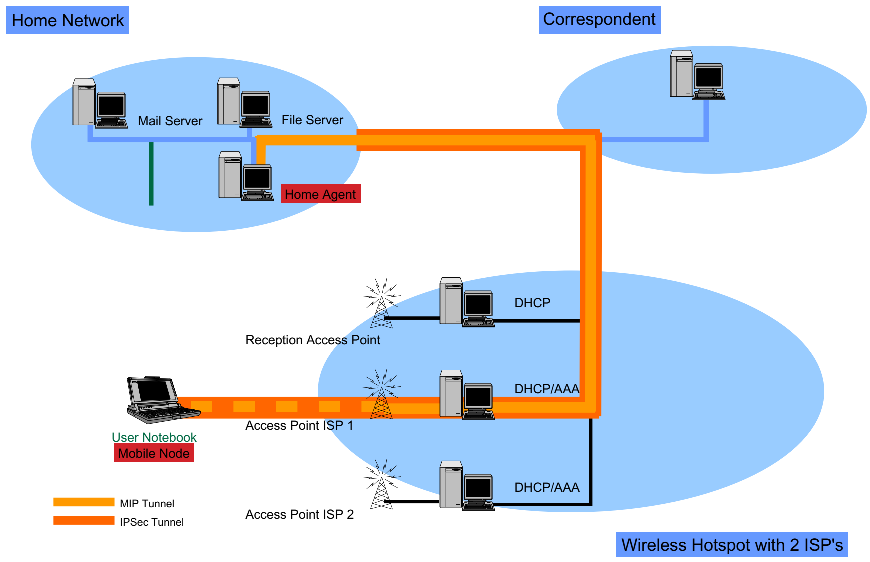

Figure 10: Layer 2: The Portable Office

6.1.2 Layer 2: The Portable Office

The second layer consists of software that supports mobility and network connectivity. It takes part mostly in the network layer (of computer network theory). Some of these parts are implemented into the operating system of the computer. But today's operating systems do not support mobility sufficiently yet. To allow a seamless handover between different access points (thus a change of IP address) and to be able to connect securely to the home network and the services the Portable Office is used (for more information see chapter about SecMIP).

The Portable Office is an ideal complement for supporting the user, and some parts of the third layer may depend on it to work properly, too. Third layer components do not support mobility, they bind themselves on a IP address, thus they can not communicate anymore after a change of the IP address. If the user does not need it, he may connect to correspondents without using the Portable Office.

The Portable Office was chosen because it was developed at the University of Berne together with Swisscom, too. This allowed access to the newest versions and support. As an alternative, Mobile IP could be used (which is also implemented into the Portable Office) but the newest version of the Portable Office (only on Computers with Windows 2000 Operating System) does not need a Foreign Agent at each access point, which is a big step forward (the functionality of the Foreign Agent is built into the Mobile Node, actually). Traditional Mobile IP provides no encryption either.

Thus, for the Portable Office, only the Home Agent (server software) on a computer inside the home network and the Mobile Node (client software) on the portable device have to be installed. This makes things much easier. Now every hot spot can be used, because there is no need anymore for any software installation at the hot spot location. Figure 10 shows the Portable Office layer. The user is in a out-of-office situation, but he has access to the home network over the Mobile IP tunnel. This tunnel is established between the Mobile Node and the Home Agent. This connection is secured through another tunnel, the IP Sec tunnel. The IP Sec Tunnel is build between the Mobile Node and the firewall of the home network.

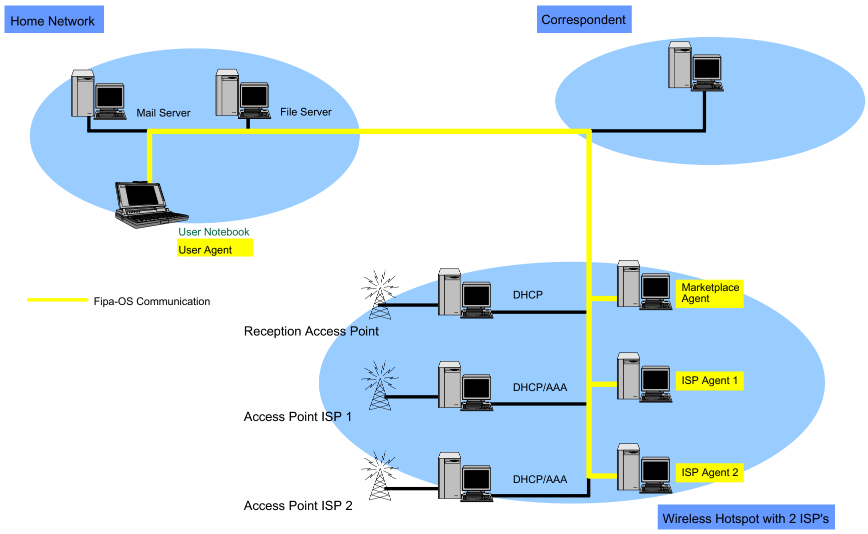

Figure 11: Layer 3: The Agent Layer

6.1.3 Layer 3: The Agent Layer

The third layer is the main layer of this project. Here is the software that supports the user in getting connectivity if he is out-of-office. It contacts ISPs, makes contracts between the users and the ISPs and establishes connections. This layer is the control layer, supplying the ”intelligence”. It takes part in the application layer of network theory.

One approach to design and implement this software is with the help of agent technology (see also introductory chapter about it). Instead of making everything up from scratch, agent technologies helps designing the entities and the communication between them in an intuitive and human oriented way and allows the use of already implemented core parts (the platforms).

There are two main standardizing bodies for agents, the FIPA and the OMG MASIF. OMG MASIF is for mobile agents, while the FIPA specifications agent are for intelligent agents. FIPA has been chosen because intelligent static agents better fitted the needs of this project than mobile agents, a variety of implemented platforms, already some experiences with FIPA-based platforms and the questioning of the need for agent mobility in general.

The platform that has been chosen is FIPA-OS, because of experiences with it, its closeness to the FIPA specifications and the platform also had to be fully open-source, which FIPA-OS is. Alternatives to FIPA-OS could be Jade or Grasshopper. On the third layer, every participator has to install the FIPA-OS platform on a computer and their respective agents on it. The platforms and agents need a Java virtual machine to run.

The main entities of the third layer are the User Agent, located on the portable device of the user, the Marketplace Agent and the ISP Agents, both likely installed at the hot spot location. Figure 11 shows an overview of the agent layer. It shows the standard situation where the user is in his home environment. He can check emails, use a file-server and communicate with a correspondent over the Internet. Now, if the user plans to go to place out-of-office wanting to connect to his home network, he thus, may tell his User Agent this.

This User Agent is installed together with a Fipa-OS platform on the users portable device. The User Agent contacts the corresponding Marketplace Agent, which returns a list of ISP Agents. The User Agent then negotiates a contract with one of these ISP Agents. It gets the configuration data from the awarded ISP after the conclusion of the contract. With this configuration data, it can later establish at the desired location a connection with the help of the OS of the portable device.

If the users comes to an access point where he has not a contract with the ISP, his User-Agent can contact the ISPs of this access point over the reception access point and so has the possibility to get a contract and the necessary configuration data.

6.1.4 Use Cases

The requirements for the whole system are condensed into three use cases.

Use Case 1 User X travels from A to B via C. He needs one or more Internet connections which may demand different QoS, e.g. the user may to want to have a video-conference or just check his emails. He plans his travel in advance and knows his demands. He tells his personal agent the necessary information, that is location, times and usage of the desired connection . The agent will then make the necessary negotiations out of the home network with the eligible ISPs

Use Case 2 This is the same case as case 1, but this time the user X will be delayed or has to cancel his whole trip. He informs his personal agent about the changes. The agent will now try to inform the ISP about the delay and get some extra time and/or a refund.

Use Case 3 X leaves his home office and goes to C, where he decides short-term that he wants to use the Internet. He informs his personal agent about this. The agent will make the negotiations on place on arrival with the local ISP.

The system should be able to make negotiation in advance, at the hot spot location and be able to handle changes in schedules and revocations.

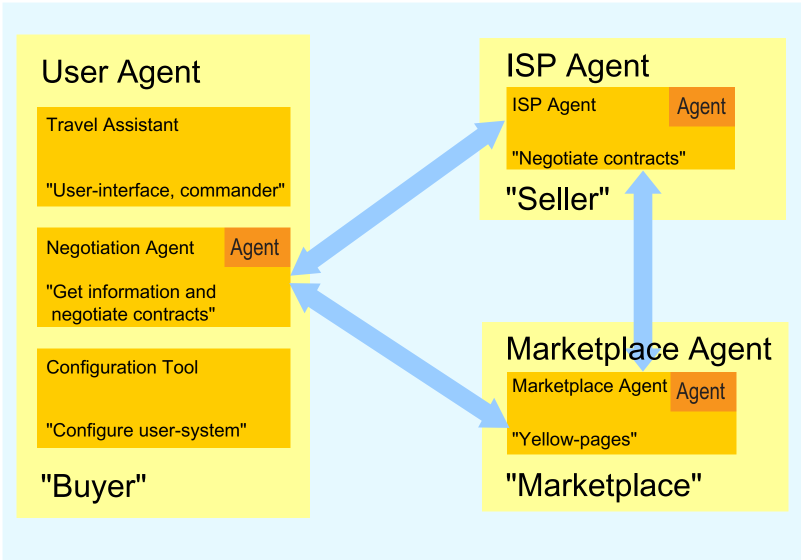

Figure 12: The Main Entities Of The Software

6.2 Software Design

The main part of this project is to develop an agent-based marketplace. Since this software realisation is agent-based, the design is quite simple. It can later be implemented in an object oriented programming language, in this case Java, with the help of the FIPA-OS agent platform and their classes. For a simple market only a seller and a buyer are needed, like in early times of human culture, where as soon as people meet, trading has been done. But as the goods got more diverse and the trading got national and international, the problem of finding two interested parties arose, the matchmaking. Hence, humans organized places called marketplaces, where sellers and buyers can meet.

Today, almost all trading is done within marketplaces. A marketplace can be very abstract, it ranges from physical locations like places and buildings to virtual sites like yellow-pages or e-marketplaces, some forms are only temporal like trade shows etc while others are nonstop accessible. An interesting example today would be the Internet auctioneer eBay. There, a person has the chance to sell e.g. an item, that most people would not be interested in and he normally would have to throw away. The buyer on the far-side might have been looking for exactly that item for a long time. To find two interested parties for a deal is a big (organizational) problem for most companies, even if their products are very good. Marketing deals with these problems among other things.

If a reasonable market is designed with agents, it needs thus at least a seller, a buyer and a marketplace entity. Figure 12 shows an overview of the main entities.

6.2.1 The Buyer Entity

The buyer entity represents the user in this case. It is called therefore the User Agent. Its main task is to go and buy the desired services, but has a lot of others tasks to do, too. Some main tasks are communication with the user, negotiation with the ISPs and the configuration of the portable device of the user.

The User Agent was therefore split up into three entities: The Travel Assistant, the Negotiation Agent and the Configuration Tool. The idea behind this is, that the FIPA-OS Agent, in this case it is the Negotiation Agent, needs a FIPA-OS platform running under it, which in many times is not necessary for other tasks the User Agent has to do, therefore the User Agent may start the FIPA-OS platform and the Negotiation Agent only when it is needed.

Furthermore, the system should be extensible with other agents, which may provide other services for traveling, and the Negotiation Agent is just one of them with a clear defined assignment, and therefore an entity for itself. The following paragraphs describe the main entities building the user agent.

The Travel Assistant - TA This is the interface to the user and the commanding entity of the User Agent. It should be extensible with other user supporting tasks and agents for traveling, therefore it is called Travel Assistant. In this implementation it only looks for connectivity for out of office situations as described in this and the previous chapters. The Travel Assistant communicates with the user over a Graphical User Interface (GUI) and delegates tasks to the Negotiation Agent and the Configuration Tool. It starts and ends the FIPA-OS Platform and the Negotiation Agent.

The Negotiation Agent - NA The Negotiation Agent is an intelligent software agent. It contacts the Marketplace Agents to get information about available locations and ISP offering service there and negotiates with eligible ISP Agents a short-time SLA on demand.

The Configuration Tool - CT From the Travel Assistant the configuration Tool gets all the negotiated contracts and the appertaining information like start and end times, technology to be used and configuration data like IP addresses, ESSIDs, WEP-keys or user names and passwords. It tries to establish the desired connections in time.

6.2.2 The Seller Entity

The ISP Agent - ISPA The ISP Agent represents an ISP at a certain hot spot location. It promotes his presence by subscribing to the according Marketplace Agent. If someone needs Internet access time at its hot spot, it negotiates with the interested agent a contract. To do this, it has to has information about the ISP, the access point, like configuration data and information about available resources, and information about pricing and offering, too. The main task of the ISP Agent is to sell services for the ISP.

6.2.3 The Marketplace Entity

The Role of the Marketplace Agent There are some different approaches how to design the marketplace entity (and also a combination between them). There are two opinions about the role of the marketplace-agent. One is that the marketplace agent helps buyers (or maybe also sellers) to find an eligible seller, and that then the seller and buyer negotiate the deal by themselves. This approach is reality-oriented, where marketplaces (like the name suggests) are locations, places, fairs etc. or on a broader level virtual in the form of yellow pages (a directory, printed or electronically). In this approach the seller and buyer agents are intelligent and the marketplace agent has only an intermediary function.

The other approach is that the marketplace not only brings the buyer and seller together, but also computes the negotiation and the contract between them. This approach would have dumb sellers and bidders that tell the marketplace only their offers and demands and an intelligent marketplace, that acts as a black-box. This approach could be interesting for companies and governments wanting to control a market and take money out of it. Another argument of this centralized approach could be that customers would trust such a system more than the individual negotiation between an unknown seller agent and their buyer agent. Some argue, that a fair and intelligent negotiation could be made easier this way, too. Interesting is that even some agent specialists favour approach. This can be viewed as a contradiction, because this ends up in the old discussion of centralized vs. distributed computing.

Agent technology is human and real life oriented and distributed problem solving is a main part of it. The advantage of decentralization is more flexibility and lightweight components. If a seller wants to change his strategy it is far easier if he can just change his agent, instead of trying to tell his new strategy wishes to a marketplace agent. Thus, it was chosen that for this project the marketplace should have only an intermediary function.

The Trading Area of the Marketplace Another design question is the scope which a sole marketplace represents. A marketplace could be a huge directory that buyers can query, e.g. a marketplace that can give the addresses of all car dealers in a city or the same marketplace could give the addresses of all assurance companies of a region. On the farside, the markets can be organized into smaller units where there is a marketplace for cars and another marketplace for assurances etc.

For this project the second way was chosen. There is a marketplace for every access hot spot. The idea behind this is that this way the marketplace agent could be located physically at the reception area, and thus, users without contracts would only have to contact this local marketplace-agent and the local ISP agents. This way, all participants could be located in one location. This decision is mainly a design decision, again. But it has to be considered, that the more different entities the marketplace represents, the more queries are likely to occur and therefore the system must be able to handle.

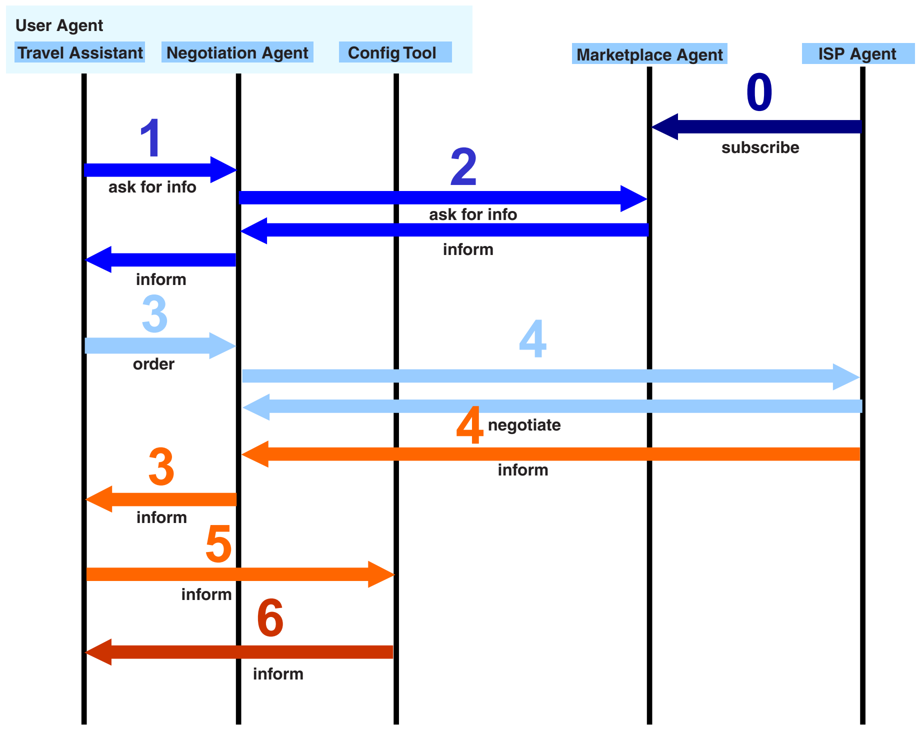

Figure 13: Processes: Inform And Negotiate

Information Exchange of the Marketplace More important is the information exchange between the buyer-agent and the marketplace agent (who should have some information about the sellers). The best way is a flexible marketplace agent who is able to handle different queries. Some buyer agents may only need the addresses of buyer agents who sell a certain product, others may want more information and filter out eligible sellers by themselves. This approach gives the buyer agent the possibility of giving the user some hints or recommendations, because he has some knowledge about sellers who are not eligible (at the moment at least).

In this design the marketplace agents gives the interested buyer-agent, e.g. some information about the registered ISPs, that is the technology they offer and accepted payment methods, but that could be some other usable information, too. By this way the buyer agent can decide for himself the eligible ISPs, but also inform the user that e.g. if he would have a certain credit card or if he would have 802.11b he could chose from much more ISPs.

The Marketplace Agent - MA The Marketplace Agent represents a geographical hot spot location for Internet access. Its main task is to inform interested parties about all the ISP offering service at its location. To have this information all ISPs offering service there register with it. The marketplace gives then the information out to interested Negotiation Agents upon request.

6.2.4 Processes

Now, all the main entities are defined. The second part of the software design is about the processes between these entities, how they communicate and in what timely order they do it, thus defining a running market.

The ISPA Registers with the MA (Figure 13, Step 0) Every ISPA that wants to sell services at a marketplace registers itself with the desired MA. The conversation is between agents in ACL with the FIPA-Subscribe-Protocol. The content from the ISPA to the MA is information about the ISP, that is the name of the ISP, the technology of the access point and accepted payment methods. If a IPS does not want to sell services anymore, the ISPA can deregister with the MA.

The TA Demands Information about the Marketplaces (Figure 13, Step 1) If the TA needs information about the marketplaces (locations etc) it orders them from the NA. This conversation is internal between objects over method invocations. After the NA collected the desired information, it passes them on to the TA. The information content is a list of all marketplaces with their location and information about the ISPs registered there.

The NA Asks all MA for Information (Figure 13, Step 2) After the TA demanded information from the NA, the NA contacts all MAs it can find and demands information about the ISPAs that are registered there. The conversation between the NA and a MA is between agents in ACL with the FIPA-Request-Protocol. The Information content from the MA to the NA is a list with information about the ISPAs registered at this MA.

The TA Informs the NA to Start a Negotiation (Figure 13, Step 3) After the user has finished the interaction with the TA and made a request, the TA gives the NA a list with all the desired connections and information about them like QoS prospects, times, locations etc. The conversation is between objects with method invocations. The information content from TA to the NA is a SLA with target values. If the NA could conclude a contract with an ISP, it returns this SLA and the configuration data received from the ISPA. The SLA consists of the names of participants, location, times, price, repayment regulations and QoS requirements.

Negotiation between NA and ISPA (Figure 13, Step 4) For every desired SLA the NA contacts all reasonable ISPAs to negotiate and conclude a contract with one of them. The conversation is between agents in ACL with a negotiation protocol. This protocol depends on the type of negotiation, e.g. FIPA-ContractNet-Protocol or a FIPA-EnglishAuction-Protocol. The Information exchanged between the NA and the ISPA are proposed SLAs. If the NA accepts a proposal it sends the ISPA accounting information and receives the necessary configuration information.

The TA Informs the CT about SLA (Figure 13, Step 5) After the TA received from the NA the contracted SLA and the configuration data, it passes

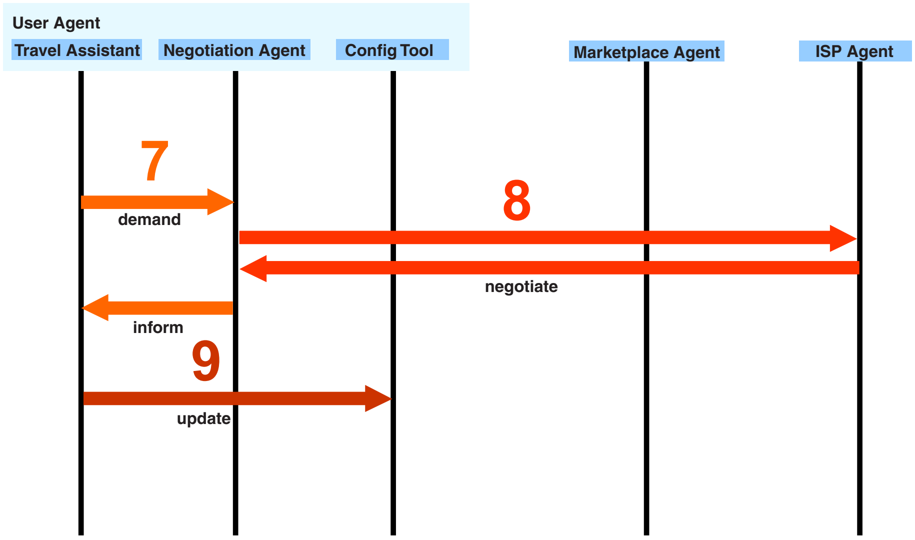

Figure 14: Processes: Handling Delays And Revocations

the relevant information from the SLA and the configuration data on to the CT. This is done through a method call.

The CT Informs the TA about Connects (Figure 13, Step 6) The CT tries to establish with the configuration data on the in the SLA described times a connection. It informs the TA if it succeeds or fails with a method invocation between objects. The TA can then inform the user.

The TA Informs the NA about Delays and Revocations of the SLA (Figure 14, Step 7) If the user wants to change a SLA or revoke it, he tells it to the TA, that then informs the NA. After the NA has contact the ISPA, the TA gets a result from the NA. This may be a new SLA and eventually new configuration data, an accept or refusal from the ISP. The conversation is between objects.

The NA Contacts ISPA about Delays and Revocations of the SLA(Figure 14, Step 8) The NA contacts the ISPA to demand a new SLA or to revoke it. The FIPA-Request-Protocol could be used for this inter-agent conversation. The content is a SLA and maybe, if necessary, new configuration data.

The TA Updates the CT (Figure 14, Step 9) In case the NA could get a new SLA, the TA informs the CT about it. Inter object communication with optional content (SLA andor configuration data).

6.3 Market Design

The market design deals with question like the rules of a market and the intelligence of the participants in it. In the software design, the entities, their role and the communication between then are described. Now it has to be specified what price negotiation protocol to use, the information exchange and what the agents have do with this information.

6.3.1 Price Determination

The negotiation protocol is decided by the choice how to determine the price and to how to negotiate. From the five possibilities of price determination (see chapter about markets), reverse-auctions, negotiations and fixed-price could be used.

Auctions do not make sense, because a user wants a contract immediately. But auctions have certain end-times, so that enough interested buyers can participate. In this project there would have to be fixed auction end-times and a fix amount of auctions per day, e.g. four auctions a day. Auctions do not make sense therfore. Furthermore, every user wants a customized contract, but at an auction this can not be considered. The same is with a stock-market, the product is not enough homogeneous to be traded that way.

A good way could be reverse auction, the user tells his perceptions and the sellers could try to outbid each other. Negotiations could also be used, but have to be implemented cleverly to make sense. Both reverse-auction and negotiations may end up with the minimum price the ISP is willing to accept. In reverse auction because every ISP wants to win and so they bid each other down till their minimum price is reached. In negotiations buyers agents could try to negotiate so many times repeatedly until they have found out the minimum price the seller agent is to offer.

So a fixed price may be the best solution, because the two other protocols may end up at the same price, unless they are with some very smart algorithms implemented (further research in this area would have to prove the advantages of certain protocols for automated markets). The fixed price is easy to implement, fast because just a call for proposal is used, fair and can be as dynamic as the other two protocols by adapting to changes in demand etc. A fixed price means the offer is not negotiable, but every fixed offer can be generated dynamically, thus the market and other environments can be taken into consideration. In this project a fixed price negotiation was chosen, because of the afore mentioned reasons.

6.3.2 Items

The right to use a network connected to the Internet over an access point with defined Qos, temporally, is the item of trading between the seller and the buyer.

6.3.3 Participators

All customers with the necessary software can participate. The responsibility to make a contract with a customer lies on the side of the ISP, that is the ISP has to check if the account information of the user is correct and trustworthy. On the seller side, all ISP that offer access at a location or hot spot where a marketplace is associated with can participate. In a real case there would to be some procedure to ensure that only trusted ISPAs could participate, to prevent user being cheated on, but in a first version, there is no admission control for customers, ISPs or marketplaces foreseen.

6.3.4 Information Exchange

In negotiations, where the buyer starts the negotiation and demands an offer, the offers goes only to him, thus making it a sealed offer. If it is an unique deal, then it can make sense and be implemented as a sealed negotiation, that is the information about the offer is hidden to the others. But when the offers occur in a great number like in this system, it is difficult to keep all the offers secret, because every competitor can easily use a dummy buyer-agent to get to know the others offers. Of course with every ISP having a different strategy and dynamic pricing, this information shows just tendencies, but everybody will know how the other's offers will look like, if they want to. So it is assumed that in this system there is a free information exchange and the offers are public, although the information flow itself is organized like in a system without free information exchange and the offers are sealed.

6.3.5 Rules

The rules are simple, as soon as a NA accepts a proposed SLA, it has to give accounting information to the ISPA, the ISPA in exchange has to give back usable configuration data. The buyer can establish with this data, if present at the hot spot location, a connection with the in the SLA defined QoS at the described times. If the ISP fails to fulfill the SLA, he has to compensate it accordingly to the values agreed on in the SLA. There is no right for compensation in case of delays or non using of the service for the customer, except it is stated in the SLA.

6.3.6 Strategies for Bidders

With a fixed price protocol, the strategy for bidders is really simple. The bidder agent just demands all eligible offers and accepts the most valuable one. The offers are valued by a decision function. This is sufficient for most agents that only bid on behalf of a user and are not ordered to deal with items and make money.

Decision Function There are different approaches for decision functions. The key is, such a function should best represent the user and his preferences. One way would could be to do this with Bayesian nets, fuzzy logic and other approaches in the field of artificial intelligence. Some of these approaches learn from the user, and could therefore not be properly used immediately. Most of these functions are really complex and demand a lot of knowledge to implement, too. This functions are itself a huge field of research.

Another way is, to calculate an area or a distance between ideal values and offered values. It was chosen such a simple and easy to implement weighted distance function, which seemed to fit this implementation. For more information about the implemented decision function see the according section in the chapter about the implementation.

6.3.7 Strategies for Sellers

The strategy and intelligence behind the seller could be one of the most interesting parts of a marketplace, specially in a competitive one. The sellers and thus their strategies compete against each other. There has to be put a great effort into designing the strategies, so that such a system really works and makes sense. The different strategies should be developed by different people, also there have to be reasonable tests in great numbers to compare and analyze the results. But it is has to be understood, that it is not simple different strategies competing against each other, as some papers propose, but that behind these strategies and agents are companies with different premises. A big company with lots of access points has other opportunities than a smaller one. To generate a strategy, that may work in real life, is a complex task. Many aspects have to be considered. Furthermore, the causing factors behind the main strategy are more likely to be marketing factors rather than mathematical.

For the reasons mentioned above and the emphasis on other areas, this part was not designed and implemented. Instead, the offers are planned to be created by a user and communicated to the seller agent over a GUI (see chapter about implementation). The design for creating offers would likely include a database of the ISP with available bandwidth for the different access points and times. Out of the received data the ISPA would generate with the help of some strategic rules a dynamic offer for the prospective customer.

6.3.8 Others

Providing fairness, consideration of fraud and foul play, integration of branding and relationships are not considered in this design.

7 Implementation

For the implementation of the marketplace and the agents FIPA-OS has been chosen. Since FIPA-OS is completely written in Java, Java Version 1.3 was used as the programming language. FIPA-OS was chosen because it is open source, some experiences with it, it is written in Java, allowing it to run on different computer systems like Linux, Microsoft Windows etc and its closeness to the FIPA standards.

The first version of this implementation was a software demonstration version only to run on one or more computers, while in a second version the code was extended to realize a demonstration in a laboratory environment with WLAN access points realizing a hot spot with two ISPs and a reception access point. The main classes, the three agents, the NA, MA and ISPA, could be directly implemented from the design. For each agent a basic FIPA-OS agent class had to be extended. Beside the main classes a lot of help-classes were implemented for the conversations, the Graphical User Interfaces, for internal data storage and some classes for the more complex NA, which are described in an own section following.

7.0.9 Changes to the Original Design

There have been two changes to the design. In the design the Configuration Tool was designated as a task running in the background all the time, trying to establish a connection. This being done with the configuration data over the in the SLA described technology at the agreed times. For a demonstration this does not make much sense, because then the user would always have to wait until the start times would be reached. Instead, it seemed much more practical if the user could establish the entitled connections with a click on the button in the user GUI. Therefore the Configuration Tool and the corresponding calls have not been implemented, instead these tasks were implemented into the TA and could be started via the TA GUI.

The second change to the design is about the ability of the software to deal with delay and the cancel of a SLA. These parts were not implemented because of time reasons and because they are not really so interesting. But they would be very important in such a system in real-life , therefore they have been considered during the design. They could be easily implemented, too. There would have to be another conversation implemented between the NA and the ISPA, and some changes in the user GUI and the corresponding method calls.

7.0.10 Integration of the FIPA-OS Platform

An important point in the design is the splitting up of the user agent into the NA and the TA. The NA can be kept light and with a clear defined appointments. It has to contact all available marketplaces and get information from them and it has to be able to negotiate a SLA with an eligible ISP out of specifications he got. To run the the NA a FIPA-OS platform has to be also running, because on start-up every agent has to register itself with the DF and AMS of a FIPA-OS platform, also later to communicate the agent needs the services of the platform. Therefore the idea would be that the TA starts the platform and after that the NA, if it needs information or wants to have a contract negotiated. But a FIPA-OS platform can not just be started that easily, it has to be cross-registered with other platforms, if it runs in a multi platform environment. Because of consistency reasons, a platform builds up a database with all other agents and platforms in it, and if after a crash or shutdown something has been changed, all the databases have to be deleted to let the system start up properly and without faults.

Therefore the start-up code of the FIPA-OS platforms has not been included into the program code, all FIPA-OS Platforms have to be started from a user, and then the particular agents can be started separately. In case of the user agent, the TA runs first only, and does not need a FIPA-OS platform running. If the user needs a new SLA he informs the TA, who then changes the GUI and starts a NA, at this point a FIPA-OS platform has to run. When the user has finished, the NA is shutdown and the GUI is changed again, and only the TA is running in the back-ground.

If the FIPA-OS platform will become more user-friendly (specially with an easier start-up), the start-up of the platform could be easily integrated into the software directly. It could also be done in this version, with changes in the platform source code, but out of the afore mentioned reasons the platform and the agents are separated in this implementation.

7.1 Message Contents between Agents

For agents it is important that they understand each other. First they have both to have the protocol of the conversation implemented. The conversations are handled in the section below. Then they have to know what to do with an inquiry over a certain protocol. In a multi-agent environment with lots of different tasks the ontology tells the agents about what this conversation is. Since this system is closed and has only one purpose, this has not to be considered. An ISPA knows when a NA starts a conversation with the FIPA-ContractNet Protocol, that this agents wants to negotiate a SLA. Third the agents have to both understand the content they send each other. Therefore the content has to be defined clearly. In a more complex system the language tags and the ontology would help an agent to know what content to expect. The content itself can be any Java object, it just has to be able to be serialized although (that is the Serializable Interface of Java has to be implemented). All participants in a conversation have both to know the Java content objects that are included into the messages. In this implementation there are three objects that are sent between the agents. Their meaning was already described in the design chapter. The definition of the fields, their type and a short description is described in the following tabulations (Table 1).

7.2 Conversations between Agents

Each conversation (communication between agents are called conversations) must have a protocol, preferable a FIPA protocol, but in FIPA-OS it is also possible to implement an own protocol. The existing FIPA protocols cover a

Field	Type	Description
Agent ID	AgentID	The Agent ID of the ISP
ISP name	String	The name of the ISP
Technology	String	Constant specifying the access point technology
Payment	HashSet	Constants specifying the payments

ISP Info

Field	Type	Description
User-id	String	Customer name
User-agent	AgentID	Customer Agent-ID
ISP-ID	String	ISP name;
ISP-Agent	AgentID	ISP Agent-UD
SLA-ID	String	Unique SLA-ID-number
Start	Date	Start date of SLA (incl times)
End	Date	End Date of SLA (incl times)
Location	String	Location of Hot spot
Technology	String	Constant of access technology
Payment-method	String	Constant of payment method
Average-velocity	Integer	Average velocity of user, information field
Latency	Integer	Maximal latency in ms, QoS Parameter
Jitter	Integer	Maximal jitter in ms, QoS Parameter
Bandwidth	Integer	Minimal bandwidth in kbps, QoS Parameter
Availability	Double	Minimal availability in %, QoS Parameter
Packet-loss	Double	Maximal packet-loss in %, QoS Parameter
Price	double	Price per minute
Credits	Integer	credits for not fulfilling QoS
Payback 100	Integer	Payback of costs in minutes before start
Payback 50	Integer	Payback of 50% of costs in minutes before start
Usage	String	Usage of connection, information field
Portable Office	Boolean	Indicates if SecMIP is used, information field

SLA

Field	Type	Description
Access Point	String	ESSID or other ID for access point
IP	String	If access point has no DHCP Server
Key	String	WEP-key or other key for access point
User name	String	User Name for access control
Password	String	Password for access control

Configuration Data

Table 1: Message Contents

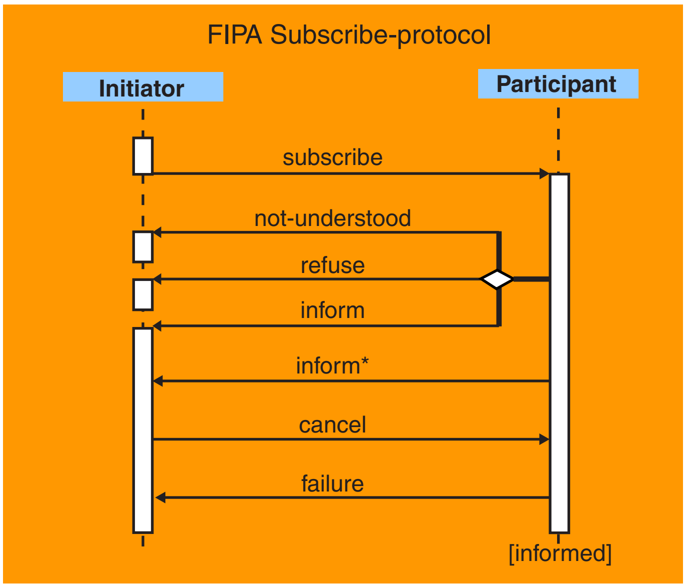

Figure 15: FIPA Subscribe-protocol

wide range, so for most cases an existing one can be used. For every conversation there is a Initiator, who starts the conversation, and there is a the Participator.

With the help of the FIPA-OS TaskGenerator, the basic classes for the conversation can be generated. The TaskGenerator needs to know what protocol to use, the ontology and the language. It generates 5 classes. For the Initiator it creates the InitiatorAbility class. This is a Java interface that has to be implemented later. In the implementing class the reaction to the answers of the Participant has to be defined. The second generated class is the InitiatorHandlerTask. In this class the conversation is started and controlled, it is the conversation task, in most cases it can just be used and does not has to be adapted, cause the intelligence is in the implementation of the InitiatorAbility. For the Participant also an interface is generated, that has to be implemented, the ParticipantAbility, and the corresponding task, the ParticipantHandlerTask.

Additionally, the TaskGenerator generates another Task, the DaemonTask. Every agent that acts as a Participant has to have a DaemonTask. This task is set as a listener-task. As soon as another agent starts a conversation, this task gets the first contact, starts a HandlerTask and gives over to it the conversation. Thus, the number of simultaneous conversation is limited only through computing power. This may be the case in a conversation that lasts for a long time, a subscription for example. If an agent, like the MA, acts as a participant in more than one conversation, the functionality of the generated DaemonTasks of all participating conversations have to be combined into a new DaemonTask, that can start the according tasks to handle the incoming conversations.

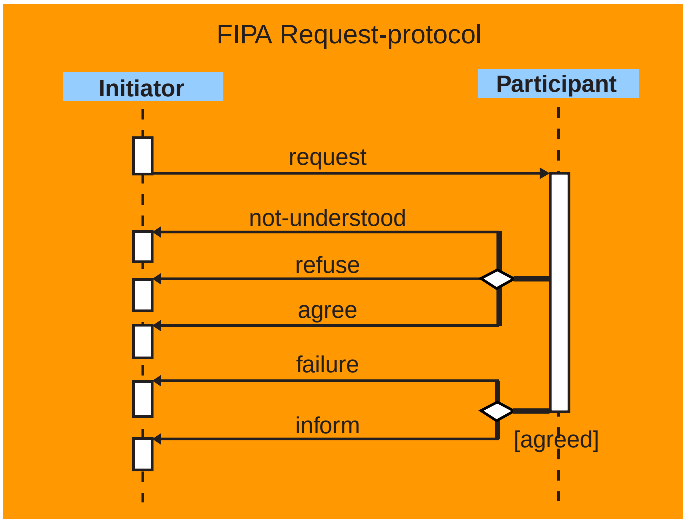

Figure 16: FIPA Request-protocol

7.2.1 The ISPA Registers with the MA

Every ISPA that wants to sell services at a certain location has to register itself at the according marketplace. The ISPA is the Initiator and the MA is the Participant. The Conversation is a FIPA-Subscribe-Protocol (see Figure 15). The content from the ISPA to the MA is an ISP-Info object (see Table 1). The special thing about this conversation is, it lasts until the ISPA cancels its subscription, that is, it will consist most of the running time of the system.

7.2.2 The NA Demands Information from the MAs

The NA asks all the MAs it can find for pieces of information about the ISPAs offering service there. With this information the NA can later decide which ISPAs it has to contact for a certain hot spot location the user wants to get access at. Furthermore, it can filter out ISPAs who sell access with technologies or payment methods that the user can not provide. It uses this information to inform the user about it via the TA, too. The NA is the Initiator and the MA is the Participant. The conversation is made with the FIPA-Request-Protocol (see Figure 16). The content from the MA to the NA is a List object with a collection of ISP-Info objects in it. Every-time, after the NA has been started from the TA, it first makes a DF-Search for all MAs, and then contacts all of the found MAs and demands from them information with this conversation. It passes then the gathered information on to the TA.

7.2.3 The Negotiation between the NA and the ISPA

If the user ordered a new contract from the TA, the TA passes this command on to the NA. The NA then contacts all eligible ISPAs and tries to negotiate with one of them a SLA. The process to find an agreement is that every ISPA is asked for an offer. Then the NA decides on the basis of a value it calculates

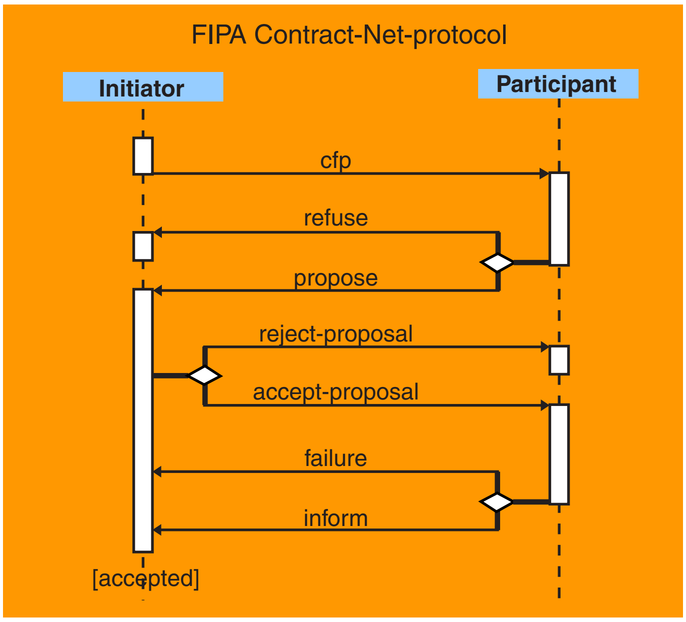

Figure 17: FIPA Contract-Net-protocol

with the help of the user's preferences. The NA is the Initiator and the ISPA is the Participant. The conversation is with the FIPA-Contract-Net-Protocol (see Figure 17). The NA sends the ISPA as content a SLA object (see Table 1), with certain target values, e.g. start and end times, needed QoS etc. Other fields of the SLA remain empty, like price etc. The ISPA then sends back a proposal SLA object. The NA can then accept, if at least one offer meets the user's expectations. The NA could send the ISPA now accounting information like credit card numbers etc, but this is not implemented. The awarded ISPA then has to accept to finish the conversation by returning an Configuration-Data object (see Table 1). This data and the negotiated SLA is then forwarded from the NA to the TA, which will display the information in his GUI.

7.3 ISP Agent

Like in the design chapter described, the ISPA is not very intelligent, and therefore held simple. It is controlled by the user over a GUI and has two main functionalities.

First, it looks up all MAs and let us a user register it at one (or more) Marketplaces. For this it has an internal list with all MAs. In this list it is also stated to which MA the ISPA is registered to. A listener lets the GUI always display the proper information.

The second functionality is that a user can enter a proposal over the GUI. This proposal is then sent to a NA demanding an offer. If the NA accepted the offer, the ISPA informs about it, too.

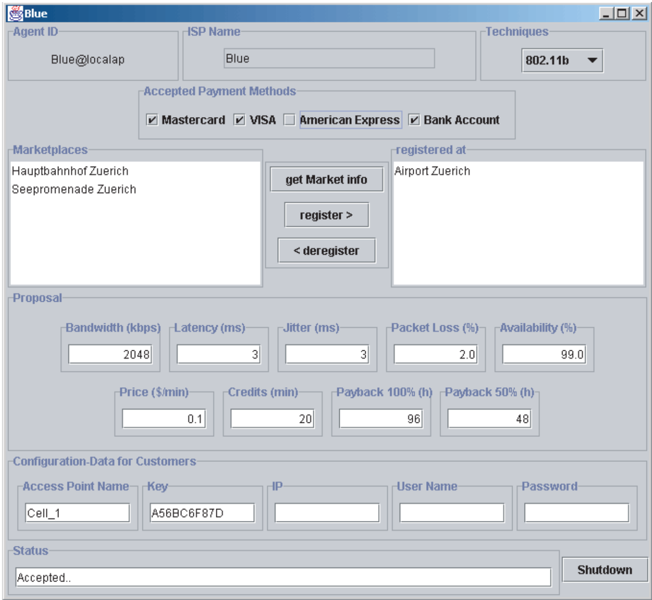

Figure 18: The GUI Of The ISP Agent

The user can also enter configuration data over the GUI, which is sent to the customer in case of a contract. In case the customer wants to connect to a real access point, this configuration data has to work.

Figure 18 shows the ISPA-GUI. On top of it is all information about the ISP like his name, the agent id, the technology of his access point and the accepted payment methods. This information is sent to the MA upon registration. The second panel from the top shows all available marketplaces the ISPA could register at in a list on the left, and on the right a list with all marketplaces the ISPA is registered at. With a click on the buttons in the middle the user can search for markets, register or deregister one from the corresponding lists. In the next panel the user can enter the relevant data for an offer, that is QoS Parameters, price, credits for not fulfilling the SLA and payback information in case of delays or revocations. In the next panel the user can enter the configuration data like access point name (ESSID), key (e.g. WEP-key), IP address (if empty this means a DHCP server is used), a user name and a password. The last panel consists of an information field, there, the user is informed when an offer is accepted etc and a button to shut down the agent properly.

7.4 Marketplace Agent

The MA is quite simple as well. It lets ISPAs register itself at it, together with some information about the ISP, and gives this information to a NA on demand. The MA has a simple GUI, where it displays the information it has about the ISPAs registered there (Figure 19). The MA has an intern data object with all

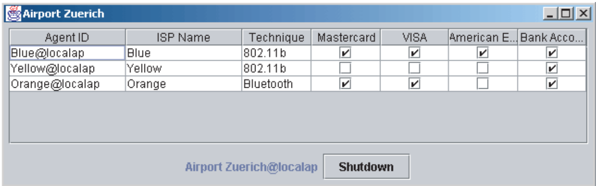

Figure 19: The GUI Of The Marketplace Agent

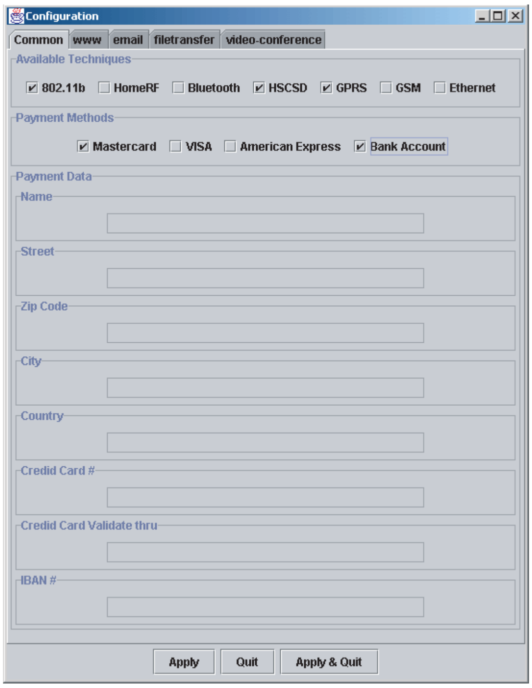

Figure 20: The Configuration GUI - Common Tab

the information about the ISPAs in it. The GUI has a listener on this data, to always display the effective data.

7.5 User Agent

The User Agent has been implemented in two main classes as described in the design. One of these two classes is the TA with a GUI and the other is the NA that makes the negotiations and other conversations with the other agents. The User Agent can be in two states.

In the sleeping-state, only the TA runs with a simple GUI. In this GUI the user can shut down the whole system, or if he needs connectivity or wants to look at or edit his existing contracts, he can start up the main GUI. When the TA changes his GUI it also starts the NA and orders him to get all marketplaces

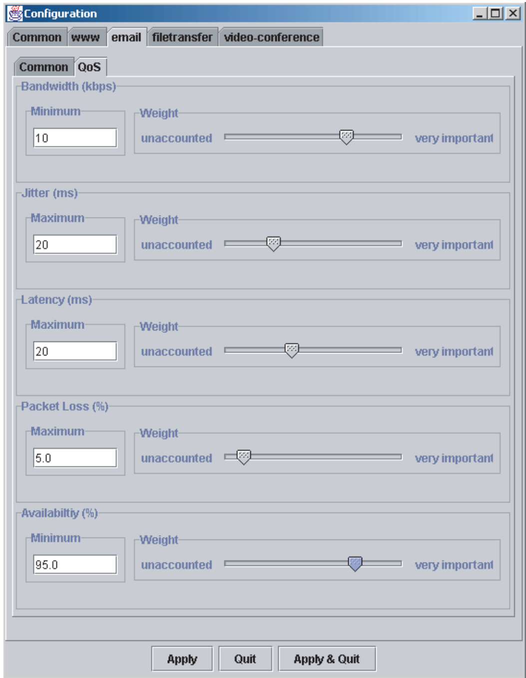

Figure 21: The Configuration GUI - Profiles Tab

with a DF-Search, then, it orders him to get all information about the ISPA registered there. This way, the TA can fill its list with the available locations correctly at its start-up. If the user is finished he can close the main GUI, and the system goes back into the sleeping-state, that is, the NA is shut down and the GUI is changed.

This splitting up into these two states does not make much sense just for a demonstration version, but it was included due to the design. The idea is to have the User Agent running all the time, so it can establish the desired connectivity with the help of the now not implemented CT anytime. But to do this, it does not need the NA and the FIPA-OS platform running all the time. In the main state the TA takes orders from the user over the GUI and passes them on to the NA.

7.5.1 Preferences Profiles for Different Applications

There have to be certain goal values for each desired SLA, weights and acceptable limits from the user, so the NA can make the right decisions and get an acceptable contract. To make life easier, four predefined profiles for four popular applications have been included. These four applications are web-browsing, file-download, email and video-conference. The user can edit and customize these profiles. This is done in the preferences GUI.

In the first tab, the common tab, the user has to enlist all his access tech-

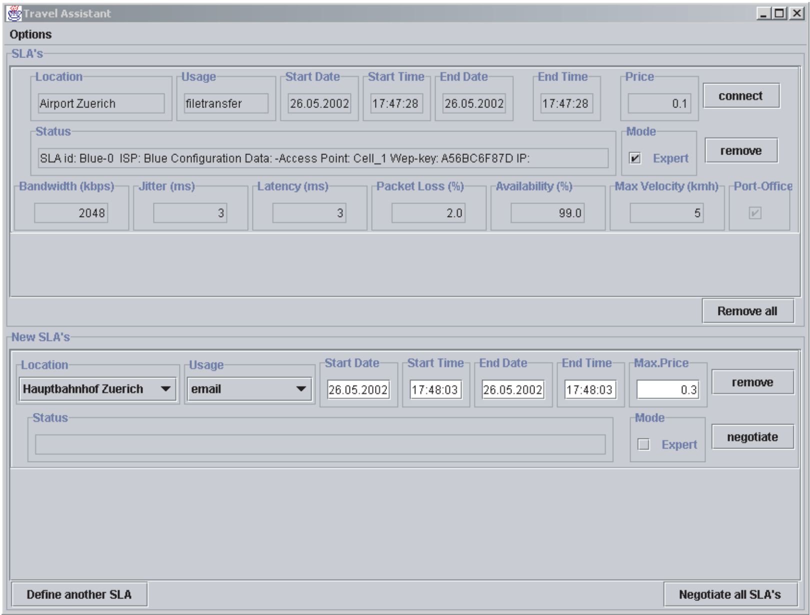

Figure 22: The Main GUI Of The User Agent

nologies his portable device supports and the payment methods he can pay with. There are some not implemented fields where accounting information to pay the ISP could be entered (Figure 20), too.

In the next four tabs the predefined profiles for surfing the web, e-mail, file download and video-conference can be viewed and edited. Each of these tabs consists again of two sub-tabs, one for the preferences in the QoS parameters and the other for the preferences for the price, credits by not fulfilling the SLA, payback in case of a revocation and for deviations for the start and end times (see Figure 21).

The user can enter for each of these parameters boundary values, that is a minimum or maximum value, that the SLA has to have for that parameter. The user can enter a valuation, how important this parameter is to him, too. The scale ranges from zero to ten. Zero means that this parameter does not count at all (the boundary values of this parameter are still considered) towards the valuation for each proposal. Ten means that this value is weighted greatly toward the decision.

How these values are used to determine a satisfying contract for the user is described below in the sub-chapter about the decision function of the NA. These preferences are stored in a configuration object of the TA, this way they could also easily be saved into a file.

7.5.2 The Main GUI

The main GUI (Figure 22) has been divided into two main panels. The one on the top shows all concluded SLAs, while the one at the bottom allows to define and request new SLAs. For every desired new SLA a new sub-panel can be created. In these panels the user has to choose a location out of the locations list, he has to choose an usage, that is the main application of this connection, a start and an end time and date and the maximum price he is willing to pay per minute. By choosing an usage, the data stored in the configuration object of the TA is loaded into the target SLA of the desired connectivity. By switching the SLA panel with the expert button to the expert mode, the user can edit the boundary values of the five QoS Values and the average velocity value for this particular connection. If he wants to change the other parameters or the weighting of them, he has to switch to the preferences GUI and make the changes there. He can only change this values directly in the panel, that may change from connection to connection. It was tried to keep the GUI as simple and user-friendly as possible.

The user can fill out as many new SLA panels as he likes. If he is satisfied he can order to negotiate for a desired SLA, or he can try to get all the filled out panels to be negotiated. If some values like start or end times are missing, the user is informed about it and has first to make the corrections, before he can let it be negotiated again. If the NA could not negotiate with any ISPs, or if it could not negotiate with all available ISPs, because the user has not all available technologies or the ISP does not support his payment methods, the user is informed on the status line of the accordingly sub-panel, too.

If the NA could successfully conclude a SLA, the accordingly panel vanishes and the contracted SLA appears as a sub-panel in the top panel. In the sub-panel of the concluded SLA is information written about the SLA like the ID and configuration data (only in the demo-version), then the start and end times and the price. The user has the possibility to change into an expert mode to see the contracted QoS values in detail. Since this implementation was made for a demonstration, the user has the possibility to connect directly to his contracted access point. This point would naturally not be available in a version, where the CT manages the connections. There is the possibility to remove unwanted SLA panels in the top as well as in the bottom panel, too.

7.5.3 The Negotiation Task

For every demanded SLA the NA starts a negotiation task. This task starts a conversation task with every eligible ISPA. After all ISPAs have responded, the negotiation task makes an evaluation of all offers with the decision function. If at least one offer qualifies, a contract is concluded with the ISPA of the best offer. It is described in the next section how the best offer is determined. If the ISPA agrees, the NA then receives the configuration data and the task finishes. The negotiation agent gets from the TA a list with all demanded SLA, so if there is more than one entry in this list, for every desired connectivity a new negotiation task is started. The negotiation task then starts for every eligible ISPA a conversation task.

This way, if the user needs five different SLA and if for every SLA 5 ISPAs can be contacted, the NA starts simultaneously twenty-five conversations. This way the user has not to wait too long, because all negotiations are simultaneous. If the desired list of SLAs is not to big and stays reasonable, there is no problem for the system with the number of conversations, at least on 800 MHz Windows 2000 box.

7.6 Decision Function

The negotiation agent, that is its negotiation task, weights all proposals it got from the contacted ISPs and the best offer gets the award. This weighting is done with the decision function. A weighted distance function has been chosen. It is simple to implement and suffices for this project. The task of making a decision in the sense of the user is a huge field of its own, which quickly gets very complicated. More intelligent alternatives to a weighted distance function in the field of artificial intelligence would be Bayesian nets or fuzzy logics based approaches.

The weighted distance function proposed by Joan Morris and Pattie Maes in their Sardine Project[44, 45] was taken for this project. It calculates for all parameters distances and sums them up for every proposal, the minimal sum being the best.

$$dist = \sum weight_i \left(\left| \frac{ideal_i - actual_i}{range_i} \right| \right) \tag{1}$$

Equation 1 shows this distance function. They propose that the user indicates for every parameter a flexibility rating. The user can tell with this flexibility rating how important the sole parameters are to him. Furthermore, he can tell how much he is flexible on the sole parameters if the outcome differs from his goal value.

With this rating the weight and the range are computed for each parameter. The weight expresses the user's interest in that particular parameter and the range normalizes the difference between the ideal value and the actual value (the range means the range of the acceptable or normally occurring values for a parameter).

The ideal value is the value entered by the buyer (the value the buyer thinks is ideal for him) and the actual value is the proposed value from the seller. Summing over all distances from the sole parameters computes the distance for every proposal. By minimizing this distance function the best offer is evaluated.

7.6.1 Adaptations to the Formula of Morris/Maes

For this project, the formula of Morris/Maes can not be implemented. The main problem is the ideal value. If looked at all parameters, three groups can be made. Group one includes all parameters with an absolute ideal value given by the customer, like start time, here of course the function can be applied without problems. The second group includes all parameters with an ideal value of zero like cost, packet loss, delay etc. In this group a user given ideal value does not make much sense. If e.g, a user enters an ideal value for the price at 10 cents, a seller however, offers a price for 5 cents and another seller offers 11 cents, then the second one will be awarded, yet it is clear that for the user the lower the price the better it is. In the third group the ideal value would be infinity, that is, the more the better, like bandwidth. Here, a value entered by the user does not make sense neither, because if one offer offers more, the function might not value this right (similar to group two). While for group two the ideal value could be set to zero, for group three it can not be set to infinity, this would not allow the function to indicate differences between offers, since the result would tend towards zero for all offers.

So, one change to the original formula is the differentiation into three cases and the calculation of the ideal values in two of this cases.

The second big change to the formula is about the ideal value, too. As explained before, an user-entered ideal value does not make much sense in many cases. The user has to give some additional information in addition to his weighting of the different parameters, that are some guideline values, or acceptable values. With ideal values, the user can say what he would like to have, but he does not say what he does not like or can not accept. Therefore it is proposed that the user enters minimum or maximum acceptable border values, depending on the parameter.

With these border values the ideal value and the range can be calculated and it is guaranteed to the user, that no contracted SLA exceeds his perceptions in any parameters. With the original formula, there is no such security. A seller could easily win a contract with in some parameters unwanted values, or even for the user unacceptable values. If these boundaries are hard or soft is not so important, but they are strongly recommended. Hard boundaries were chosen for this project. Now, before the decision function values the proposals, all proposals that do not fulfill all the minimum or maximum values are sorted out, so only eligible offers are weighted and therefore considered for the final decision.

This algorithm takes two inputs for all parameters from the user, the boundary value and the weight. The parameters from group one, that is such parameters that have an ideal value already given, this ideal value is also taken as input, e.g. start time. First, the algorithm selects all eligible offers with the help of the boundary values. Then the ideal values and the range are calculated where necessary with the help of the boundary values and the weight. The actual values are given by the proposal. Now, all proposals can be evaluated and if at least one offer is eligible the best offer is determined.

7.6.2 Determining the Ideal Value

For the calculation of the ideal value it is proposed to look for every parameter separately at all offers and take the value of the best offer for this particular parameter. This best value is then lowered or raised. This prevents, that the offer, that this value is from, has a too big advantage against the other offers, which would bias the final result to much.

First it was considered to lower/raise this determined ideal value ten percent, but the function did not perform to well. It was decided that the weight should play a more important role (see below), therefore it is proposed to lower or heighten the from the offers calculated best value with a fraction of one over the weight times itself (see formulas 3 and 4).

Since the weight can be zero, one is additionally added to the weight, to prevent divisions trough zero.

7.6.3 Determining the Range

The range is calculated, too. It is proposed to take two times the difference between the (calculated) ideal value and the user given boundary value.

7.6.4 Determining the Weight

There are some changes proposed to the original formula towards the weight, too. One change is, that the weight can be zero. This simply means, that a parameter with weight zero is not considered into the final evaluation (the minimal/maximal boundary values are still controlled although). With this it is possible to turn off all for the user not so important parameters. This is important since every additional parameter could bias the final result.

The second change is about the weighting of the weight itself inside the formula. A big problem to such weighting functions is that the different parameters are difficult to take into consideration equally. To make them comparable they are being standardized, but here lies the problem. How should the range be calculated to make the parameter standardized? It was tried with the inclusion of the weight in the range to make the range more fitting. Still the user given acceptable values can make a big difference into the outcome. Some parameters can be easily overrated. While looking at different calculated values, which are all in a range between 0 and 0.5 (before the multiplication with the weight), they could be differed greatly, even if the actual compared parameters would be similar compared to each other. Now, when a user enters for a to him neutral value a weight of five, and for a more important value a weight of eight, it is possible, that the less important parameter makes the decision. To prevent this, it is proposed to give the weight more power in the formula. Thus it is proposed to square the weight. With this change it is believed to make the decision function more stable and better user-representing.

7.6.5 Problems of Weighted Distance Functions

There a two big problems with weighted distance functions. The first is, as already discussed above, that it is difficult to compare the different parameters and therefore take them equally into the decision making, because they may be very different in their nature. Surely they are made standardized (with the division trough the range), but the ranges and the values itself can differ too much and therefore bias the final result. By looking for example at the parameters price and bandwidth: A price that is as twice as much from another proposal might be in-acceptable, while with bandwidth ten times the value offered by the competitor might not be valued by the user as ten times as well. With an increasing numbers of values to be considered, the possibility of a biased result increases, too.

With more weighting on the weight, the consideration of the boundary values for the range and the possibility to decrease the number of considered values, the function should be enhanced enough to be usable for most cases. Furthermore, the user can help by setting reasonable boundary values and to him uninteresting parameters to zero.

The second problem, which at least applies to this projects distance functions, is the "hijacking" of negotiations. This could be done by a seller that offers for a to the user less important parameter an unreasonable extremely good offer. Thus, this sellers offer might win an elsewhere balanced competition, even if this offer has otherwise to the user worse values than the competing offers. For example, in this project the parameter of credits (in the case of not fulfilling the SLA) could be abused this way. Prevention of this case could be in the planning the elimination of unnecessary parameters, specially ones with an ideal value of infinity, or the inclusion of a detection of standing-out-values, or as a user the consideration of this by the configuration of the weights.

For the decision function exits the dilemma, that on one side, to fulfill the users preferences optimally, it is wishful to include as many parameters as possible, on the other side, the more parameters involved, the chances increase of an unwanted result. With the adapted decision function it is believed to get a suitable and in most cases satisfying result. For a further enhancement, it is believed it would be the best to have for every parameter a curve instead of just the boundary values. This could help to better represent the user opinion of good, not so good, acceptable or non acceptable (e.g. fuzzy logic).

7.6.6 Roundup of Changes

The changes to the formula of Morris/Maes are the following:

- The introduction of boundary values and the preprocessing of all offers to filter out none-eligible ones
- The distinction of all parameters into three groups and the calculation of the ideal value for two of these groups
- The calculation of the range out of the ideal value and the boundary value
- More focus on the weight by squaring it and the inclusion of the weight into the calculation of the ideal value

7.7 Decision Function Formula

Input: User given values and at least one offer.

Variables, for every parameter:

- Weight : Weighting of parameter, user given
- Ideal : Ideal-Value of parameter, calculated or user-given
- Actual : Proposed-Value, given by offer
- Boundary : Maximal or Minimal Acceptable-Value, user given

Algorithm:

1. Sorting out of offers with unacceptable values with the help of the boundary values
2. Calculation of the ideal value where necessary
3. Evaluation of every offer with:

$$dist = \sum weight_i^2 \left(\left| \frac{ideal_i - actual_i}{2(ideal_i - boundary_i)} \right| \right) \tag{2}$$

   The offer with the minimal evaluation value being the best

For the calculation of the Ideal values, all parameters are grouped and distinct into three groups :

Group 1: Ideal Value Absolute and Non-Zero Parameters:

- Start-date and Time
- End-date and time
- Availability (here ideal value is 100 %)

The ideal value does not need to be calculated, since for this group it is user given or absolute.

Group 2: Ideal Value Zero Parameters:

- Price
- Payback 50 %
- Payback 100 %
- Jitter
- Latency
- Packet Loss

Calculation of the ideal value for this parameters:

$$Ideal = Minimum - Minimum * \left(\frac{1}{Weight + 1} \right) \tag{3}$$

Minimum: The minimal value of all offers for this parameter.

Group 3: Ideal Value Infinity User Input: Ideal Value, Boundary Value, Weight

Parameters:

- Start-date and Time
- End-date and time
- Availability (here ideal value is 100 %)

Calculation of the ideal value for this parameters:

$$Ideal = Maximum + Maximum * \left(\frac{1}{Weight + 1}\right) \tag{4}$$

Maximum: The maximal value of all offers for this parameter.

Figure 23: Demonstration Setup Overview

8 Realizing a Demo in a Wireless LAN Environment

A goal of the diploma thesis was not only to realize a software system, but also to realize a demonstration of it in a laboratory environment, to not only show the negotiations, but also to show the result of them, the ability of connecting to the awarded ISPs access point.

A simple scenario was chosen with a hot spot situation with two ISPs and a public accessible reception access point. As access technology 802.11b WLAN was chosen, because it ideally fulfills the requirements to be able to connect from a mobile device to different access points, and the cards and access points have been available. In Figure 23 is an overview of the network and demo situation.

A small network was built with four Linux boxes, running on Intel Pentium processors with tact rates between 300 and 525 MHz, three access points with a transmission rate of 2Mbps and the portable device was a 800 MHz Intel based notebook with Microsoft Windows 2000 running on it.

Three independent subnets were build, two representing an ISP and one the reception area, each consisting of an access point and a computer. The computer of the reception area also acted as a router that connected the three subnets and the forth computer, that acted as a correspondent in the Internet with a http server running on it. Since the access point did not offer dhcp services, a dhcp server was installed on all three computers that were connected to the access points. On these three computers a Java 1.3 runtime environment for the agent-platform, the agents also FIPA-OS as agent-platform has been installed.

Subnet 1 represented ISP 1, a ISP offering 2Mbps max data throughput at according prices. On the computer the ISP agent representing this ISP was running. On the access point the ESSID was set but no WEP-key, because the used WLAN-cards did not support WEP-keys. The values for the agent were set accordingly to the role of this ISP, these are prices, bandwidth, other QoS values, and the correct ESSID.

Subnet 2 represented the public reception area, on this access point the standard ESSID (any) and no WEP-key was set, so everybody could connect to it. On the computer behind the access point a firewall was set-up, so that only connections to the local agents could be made. Furthermore the firewall had to be configured that it blocked unwanted traffic when the user was using mobile IP, because the Mobile-IP home-agent was running on the same machine (this was done to keep the scenario simple). The marketplace agent was situated on this computer, too.

Subnet 3 represented the third ISP. This ISP was cheaper than the other ISP, but did offer much less bandwidth. The limitation of bandwidth was done with a token bucket filter at one of the interfaces of the Linux box behind this access point. The limit was set to 200 Kbps to show a clear difference. Elsewhere, the computer was set up like the one of ISP 1, except the access point itself had another ESSID and the values of the ISPA running on the box were adapted.

The demonstration scenario firstly was to connect with the portable device to the reception access point. Then to make a negotiation with the preference of a good price. So, the NA should conclude a SLA with ISP 2. After that switching over to the access point of ISP 2 and to start a download of a file from the server. With 200 Kbps this would go slowly, so the demonstrator would make a new negotiation, this time with the preference on a good bandwidth. The NA should now negotiate a SLA with ISP 1. After switching over to this ISPs access point, the download should speed up clearly.

For this demonstration also Mobile IP was needed, so the connection for the download did not crash while changing over to a different subnet and the agents could continue to communicate and negotiate after an access point change. For simplicity reasons the computer behind the reception point was chosen as Home Agent. Dynamics Mobile IP[46] was installed on it.

To make things easier the Portable Office Client was chosen, which supports Mobile IP, too. This way just the Portable Office Client had to be installed on the portable device and it was no need to install a foreign agent on the subnet of the access points, like it would have been needed by using the Dynamics Mobile Client. Since the Portable Office Client was only available for Windows 2000, this operating system was run on the portable device instead of Linux.

This again was nice because it could be shown that the system works well in heterogeneous environments.

8.1 Problems

The setup of the system took some time, but was without big problems. The demonstration system ran and the demo could be showed as planned, but there have been some problems with two parts of the system.

- SecMIP client on the portable device: For the demo a beta version of the SecMIP for Windows 2000 was used. This version did not run dependable.

- Fipa-OS platform: The Fipa-OS platforms are not that easy to handle. To start up the whole system every platform has to have an entry of the platforms it is going to be cross-registered with. Then the DF has to be started and cross-registered with another platform. This takes some time, everything has to be done step after step, the java based platform run slowly. By overburden them or overhasty behavior the whole system can crash, and the whole set-up has to be started again. Furthermore, if the system crashes while running, it is best to shut-down everything and start over.

  Fipa-OS builds a database with all the information about the other platform in it, to allow recovering without problems. This database was more a burden, because before every new start-up, it had to be cleaned in order to guarantee a trouble-free running, specially if something changed in the system, but it proved to be more secure to do a new start-up every-time, even if the system stayed the same.

  If the system was running, and one of its participants got problems, the other platforms and agents running on them would also get problems, when trying to contact a crashed or overloaded platform. During a demo even the whole Java virtual machine running on the portable device crashed, after a participating platform was shut down by accident.

  Fipa-os platforms need some time to be started up and do not run robustly. Hopefully, the critical points are improved in future releases.

9 Agent Performance Tests

The project demonstrator was rather small with only one buyer, two sellers and a marketplace agent. It showed that the negotiations and the idea of the system work. But if such a system was implemented in reality, it would have to be much bigger and consisting of hundreds of agents to be reasonable. But where are the limits of such a system in terms of performance and participants? To explore some possible problems and bottlenecks, some simple tests have carried out. It has to be noted that the current implementations of agent platforms are not suitable for commercial use. They are just implemented for research, and are therefore aimed at testing agents and not meant to support a big number of platforms. However, the results of the test may give some hints concerning the problems in commercial mass agents systems.

The idea of putting several platforms on one computer to test e.g. the communication speed between platforms linked in a chain was not possible because the current implementation of FIPA-OS does not support multiple platforms on one host (they could be started but could not be connected) and even if it would, the performance of the available Linux computers (330 and 525 MHz) would not have been sufficient. The computers got already performance problems by to many agents running on them.

The setup of the participating computer, that is the start and configuration of the platforms and the cross-linking of them together, was very time consuming. Furthermore the available computers were limited and the default implementation of FIPA-OS allowed only a reachability scope of two platforms in the deep (linked in a chain) from one platform, so it was decided to limit the tests to maximum three platforms.

By looking at this projects system, there are two different ways of communication between the platforms and therefore two different possible performance problems (there is actually a third kind of communication between the platforms, the cross-linking of them at the start-up, but this is not considered here). The first is between platforms and goes through the system the way the platforms are linked together (the way they were cross-registered at each other, not physically), the second is direct agent to agent communication, which goes directly from the sender platform to the receiving one.

9.1 DF-Search

When an agent needs to know what other agents are registered at the system it can search the Directory Facilitator (DF) of the platform (The DF is itself an agent, which has yellow-pages functions). If the platform is linked with other platforms, the DF forwards the search on to the other DFs of the platforms recursively (until a defined recursion depth, on a standard FIPA-OS platform only two). Every agent can search the DF for all agents or for a specific kind of. In this diploma project, e.g, the NA needs to know what MAs are around, so the user can choose between all the available hot spots. Therefore the NA starts at his start-up a DF-Search for all MAs (the ISPA makes a DF-Search at startup to find all MAs, too). The structure of how the platforms are linked together

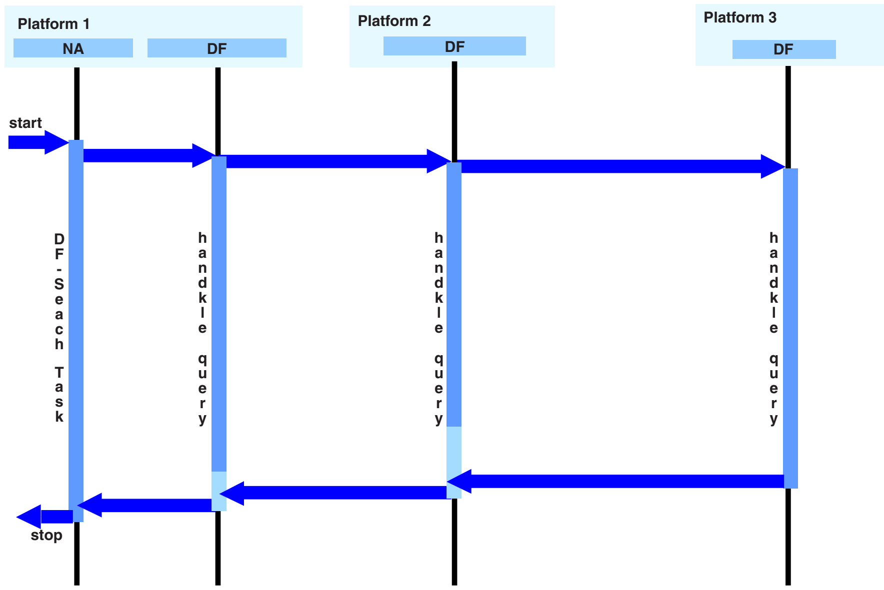

Figure 24: DF-Search with 3 Platforms (Test C)

may play a role only for this test, because for the other tests the platforms can contact each other directly and must not recursively travel through the whole FIPA-OS network. The limitation for only three platforms is not really sufficient for testing a DF-Search. It would be interesting, e.g, to compare the times of a DF-Search of ten chain-linked platforms versus ten platforms in a star-like or combined structure. On the far side, for this system the DF-Search is not time critical. Research results in the field of peer-to-peer networking, which face similar problems, may help advance future agent platforms. Furthermore, it has to be noted that FIPA-OS changed from a central server based approach to a decentralized one in recent versions.

Every agent can start a DF-Search. To do so, it starts a DF-Search-Task. This task contacts the DF of the platform and queries it. If the platform is linked together with other platforms, the DF forwards the query to the other DFs that are located at the other platforms, and gives the results back after he received them from the other DFs. After the DF-Search-Task receives the results, it hands them over to the agent and terminates itself afterwards. For the tests the time was stopped between the start of the DF-Search-Task until its completion. The query was to search for all Marketplace Agents.

Test A For the first test, test A, a DF-Search was measured on a single platform, platform 1. The platform run on a 333 MHz Linux box. The results are shown in Table 2, column 1.

Test B For test B, two platforms have been linked together. The DF-Search query was started on platform 1. All two platforms were running on a 333 MHz

Test A	Test B	Test C
1 Platform	2 Platforms	3 Platforms
in ms	in ms	in ms
1120	4009	4390
735	3505	2880
757	2402	2741
657	2241	2551
1179	3107	2515
1275	2704	2629
573	3502	2681
731	2771	2219
602	3006	2857
898	3236	2086

Table 2: DF-Search

Linux box. The results are shown in Table 2, column 2.

Test C For test C, three serial linked platforms were used (see Figure 24). The DF-Search was always started on the first platform. The first and second platform ran on a 333 MHz Intel box, while platform 3 ran on a 525 MHz Intel based box. Table 2 column 3 shows the results of the third test.

It was expected that the DF-Search would take longer when done over multiple platforms, which the results proved. Surprisingly, the average search time for three platform chain-linked together compared to the test with two platforms was almost the same. This is probably because the platforms forward directly the search request first. The results suggest, that probably by linking more than three platforms in a chain together, the DF-Search time will stay almost the same and will not grow too much. In the case of thousands of platforms, the DF-Search time might be in the range of a search in a peer-to-peer file-sharing network.

It is noticeable that the search times vary largely. This might be due to different states the computers is in. It can be said that the search times vary more with the these particular states the computer may be in than the depth of the linked platforms.

Is it clear that ten tests just show a tendency and are not statistically relevant. But to get a statistic relevant result, the tests must be made in great numbers, because of the great variance. This does not make much sense, because in all the test that were made, the first search or conversation after the start-up of the whole system always took much longer than the following tests (up to 2 times), which in mass tests could not be considered. Furthermore, with the great variety of results, a single average value does not help much, unless the results would be used to optimize the platforms. To show a tendency and the broad range of possible results, ten tests should be enough.

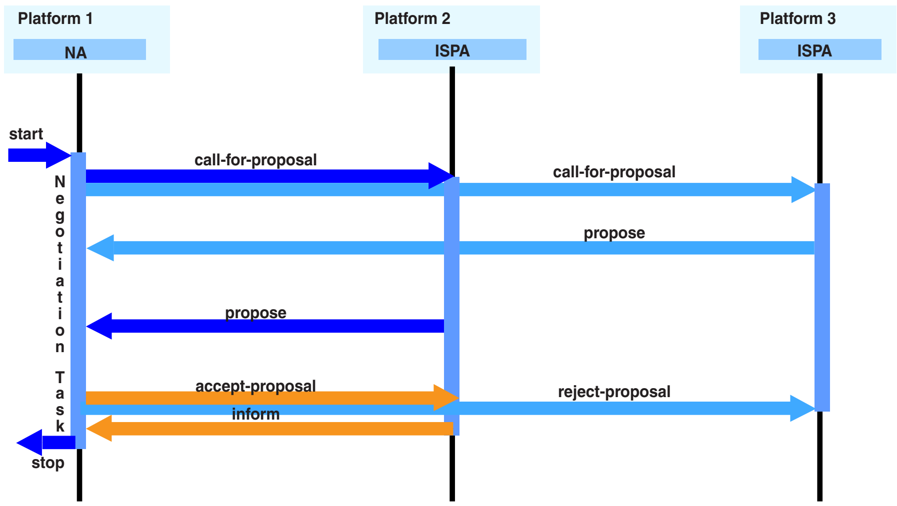

Figure 25: Negotiation with 2 ISPs on 3 Platforms (Test G)

9.2 Negotiation between Agents

While a DF-Search only happens at start-up of the NA and the ISPA, the real challenge of performance of this system lies in conversations between agents. There are three such conversations, while the negotiation between the NA and the ISPA (contract-net protocol) is the most complex one. Negotiations are more time critical and more likely to happen in a major quantity. The other conversations in this system are the request for market information from the NA to the MA and the subscription of the ISPA to a MA.

The next three tests were similar to the first three tests of the DF Search, but this time it was looked at the conversation of two agents, the negotiation between the NA and the ISPA. The time was stopped on the NA between the start of his negotiation task until this task was finished.

Test D The first test, test D, was on a single platform, running on a 333MHz Linux Box. One NA negotiated a SLA with one ISPA. The results are shown in Table 3 column 1.

Test E The second test, test E, was on two platforms. The initiating NA was on platform 1 while the participating ISPA was on platform 2. Both platforms run on a 333MHz Linux Box. The results are shown in Table 3 column 2.

Test F The third test, test F, was again on 2 platforms, but this time on platform 1 and platform 3. The NA was on platform 1 and the ISPA on platform 3 (Table 3, column 3). The difference to test E was, that Platform 3 ran on a 525 MHz Linux box.

The results of the two-platform tests seemed to be a little bit faster than the test

Test D	Test E	Test F	Test G	Test H
1 Platform	2 Platforms	2 Platforms	3 Platforms	3 Platforms
1 ISP	1 ISP	1 ISP	2 ISPs	3 ISPs
in ms	in ms	in ms	in ms	in ms
1012	1682	1059	1931	1191
641	462	417	1340	818
420	589	387	730	1164
966	519	372	701	828
448	619	385	666	789
489	455	520	974	691
418	568	640	634	1015
672	473	343	628	701
528	396	423	604	697
462	496	420	1052	642

Table 3: Negotiations

on one platform. This is because the processing load is now distributed. The negotiation between platform one and three was in average a bit faster than between platform one and two, but this due to the faster processor platform three run on.

But what happens if the NA negotiates with more than one ISPA? In the next two tests the NA negotiated with two ISPAs and then with three ISPAs.

Test G For this test (Figure 25), the NA was on platform 1 (333MHz), and the participating ISPAs where one on platform 2 (333MHz) and the other ISPA on platform 3 (525MHz). The results are shown in Table 3 column 4.

Test H Same scenario as in test G, but this time, there has been running an additional ISPA on platform 2. Thus, one NA negotiated with three ISPAs. The results for this test are shown in Table 3 column 5.

Not surprisingly, the times went up from negotiating with one ISPA to negotiating with two ISPAs to negotiating with tree ISPAs. This is explainable by the fact, that the NA has to wait until all ISPA answered him, and then can make his decision and finish the conversations. The NA always has to wait for the slowest reaction of his participants. The more participants there are, also the probability rises, that one of the participants reacts a little bit slow.

For the next test, it was looked at the negotiation times of a NA with a busy ISPA. For this test the ISPA was being queried simultaneously by five automated NAs. These NAs were negotiating non-stop, that is, as soon as they good a negotiation finished, they started a new one.

Test I For this test, all participating agents (6 NA and 1 ISPA) run on the same platform (platform 1, 333MHz). Five NAs negotiated in a loop with the ISPA. A sixth NA then made one negotiation with the same ISPA , and this

Test K	Test L	Test M	Test N	Test O
1 Platform		2 Platforms		NA to 2 ISPAs
1 NA	5 NA	1 NA	5 NA	5 NA
number of negotiations per 10 sec				
28	30	35	49	17
31	31	36	56	20
30	29	37	50	18
33	30	39	58	19
33	31	38	53	17
32	29	41	56	18
31	32	38	50	19
34	30	40	58	17
31	32	39	54	18
30	31	38	56	19

Table 4: Capacity of the ISPA

negotiation was measured. The resulting times (from 10 tests) range from 2109 ms - 2448 ms.

Test J For this test, it was the same scenario as in test I, but this time the agents where distributed over 3 platforms. On one platform, the ISPA was running, on the other one the NA that made the test negotiation and on the third one the five automated NAs. In this scenario the results ranged from 1715 ms - 2405 ms.

The negotiation time (compared to the tests D-F), as expected, slowed down with a busy ISPA.

9.2.1 Capacity of the ISPA

But, how many negotiations can an ISPA handle? To find this out, first one and after that five NAs made non-stop negotiations with an ISPA, i.e., as soon they finished, they started the next negotiation, and the ISPA counted the number of negotiations it could finish in 10 seconds.

Test K This test was done on one platform (333MHz). One NA negotiated as many times as he could with one ISPA. Table 4 column 1 shows the number of negotiations the ISPA could finish in 10 seconds.

Test L The same scenario as test K was used, but this time five NA made negotiations simultaneously with the ISPA. Table 4 column 2 shows the results.

Test M For this test the same test scenario as test K was used, but this time the test was distributed on two platforms. On platform 1 (333MHz) the NA was running, and on platform 2 (333MHz) the ISPA. The results are shown in Table 4 column 3.

Test N For this test, the same scenario as in test L was used, but the five NAs and the ISPA were situated on two different platforms, both platforms running on a 333MHz Linux box.

Again, the distributed system was faster than the one running on a single platform. Surprisingly the ISPA was able to have more conversations with five simultaneous NAs than with just one, even on the same platform. This is due to the fact that negotiating with five agents simultaneously, the ISPA has less idle times (when it has to wait for the response) and can therefore serve more agents. It can also be said, that the more different NAs negotiate simultaneously with an ISPA, the more NAs an ISPA can serve, but also the more time a single NA has to wait for a response.

But what happens to the efficiency of the ISPA, if the served NAs, like in a normal case, negotiate with more than one ISPA.

Test O For this test three platforms have been used. On platform 1(333MHz), five NA were running. They negotiated non stop simultaneously with an ISPA that was running on platform 2 (333MHz) and a second ISPA that was running on platform 3(525MHz). The ISPA on platform 2 counted the negotiations he could finish in 10 seconds. The results are shown in the last column of Table 4.

It was expected, that the number of served NA would be smaller than in the four tests before, because the initiating NAs now had the double amount of tasks running, but that the added waiting time of the NAs would bring the performance of the ISPA down like this was a little surprising. Thus, the amount an ISPA can handle depends on the number of NAs negotiating simultaneously and the reaction time of these NAs, which depends on the number of ISPAs the NA contacts itself and the processing power of the NA.

In this system the reaction time of the NAs will remain short, because the number of ISPAs at a marketplace is limited to just a few and the user is not likely to make to many contracts at once. The main factor of the performance of the ISPA (and the NA too) is its own processing power and the available memory. This is because the biggest fraction of the duration of a conversation take up the Java tasks that handle and interpret the conversation. The Linux boxes used in this test with a 333 MHz Intel processor are on the verge, and likely participants of such a system should have better hardware. So, the implementation of such systems need at least some powerful hardware.

10 Security Issues

What is of great importance for a system proposed as such is security. In the design and the implementation only a few security aspects have been covered, but not all, because the main focus was the realization of a working system. Some of the security risk are not that easy to solve and need further research, e.g., for secure micro-payment. The main security issues are:

10.0.2 Security on Layer 1

The main security risk on layer 1 (of this project's design), the physical layer, is to protect the ISPs from unauthorized active network access. This means to prevent someone from using the network infrastructure over an access point without being authorized to, in most cases would that be without paying for it. Because the access points are wireless, it is easier to use the network without being noticed, than it is with wired ones. But it depends on the used technology. Mobile Phone Networks are very secure to break in, while 802.11b access points are sometimes not secured at all and even if are they can be broken into relatively easily. Authorization methods for 802.11b access points:

- WEP-key and private ESSID. This is only of use if nobody knows them and if this information changes constantly (the user can get this information easily out of his portable device). The WEP-key and ESSID can be hacked!
- Access control with a temporary IP addresses.
- Access control with the MAC address of the portable device (the MAC address can be changed in most wireless LAN cards !)
- Authentication Server (AAA): This server only allows access from users that can authenticate themselves with an code or user name/password couple.

All these measurements are here to protect the ISPs. But, the reception access point has to be secured as well, to prevent abuse. The idea of this access point is, that everybody can connect to it, in order to let his User Agent negotiate a contract. Thus, it has to be secured against abuse. The best way is to have a firewall or some traffic control, that blocks unwanted traffic (i.e. traffic that goes to addresses outside of the hot spot location). If it should also work with SecMIP (that builds up a secured tunnel to the users home network), then it has to be assured, that only agent communication is allowed. To control this, only un-encrypted connections have to be allowed. This can be achieved by allowing Mobile IP only, but without IPSec (this would not conflict with a secured agent communication, which is essential, see in paragraph about 3 layer security below)

10.0.3 Security on Layer 2

It is important on the network layer to secure the connections of the customer, so that nobody can listen to network traffic and gain access to private data. This is mainly the problem of the customer (most ISP declare explicitly that this is the responsibility of their customers).

A possible connection firstly goes over a wireless network and then over parts of the Internet. The part over the Internet is not secured by default, however it is harder to gain access to listen to a connection on it. On the far-side it is easier to listen to a wireless connection, presumed the eavesdropper is at the hot spot location. Most wireless technologies have some built in security mechanism. But again, they vary. While mobile phone networks are considered secure, wireless LANs encryption with the use of a WEP-key is weak an can be broken into. Most of the time it is not even used, too (most ISPs in the wireless hot spot market do not use it). If a WEP-key is used, it has to be held secret.

With un-secure wireless access points and traffic over the Internet, the user should always secure sensitive connections by himself, with a VPN. With the built in encryption (IPSec) of the Portable Office, which is used in this project, there is a nice solution to this problems.

10.0.4 Security on Layer 3

There are two main security concerns on the agent/application layer:

Agent Communications Most agent communications go over unsecured connections. But they contain sensitive date, specially when two agents negotiate a contract. In negotiations banking information is exchanged against configuration data, both are very sensible data. Therefore inter-agent communication must be secured. This is possible with FIPA-OS platforms. The communication between platforms can be encrypted with SSL. This is done with the built in SSL support of the Java RMI. But this has not been tested in this project and there is no information known on how secure it would be.

Trustworthy Participants The second concern is the trust of participants in a market. For the ISP it is important that it gets valuable accounting information from the customer. This accounting information should enable the ISP to get money and it must be ensured that this money really comes from this customer. On the other hand, the customer has to trust the ISP agent, that he really gets access time at his desired location. It has to be ensured, that all ISP Agents are really representing real ISPs, which can offer the sold services, too.

Therefore, all in a market participating agents have to be able to be authenticated. This allows not only trustworthy transactions, it allows to prove that a transaction has been made, too. This could be realized with a Public Key Infrastructure (PKI)[47]. PKI is based on certificates, these are issued by certificate authorities, and a public/private key encryption. With PKI it is possible to validate the identity of each party in a transaction and is ensured, that the messages have not been changed or corrupted during transit. Furthermore, PKI can be used to authorize access and transactions.

In the standard FIPA-OS distribution, PKI elements are not yet implemented, bur there are groups working on these issues, specially to develop new speech-acts for identification and authorization on the level of FIPA, and to implement them in FIPA-OS, too.

11 Conclusions

11.1 Agents

Firstly, the idea of agents has to get used to. However, after a while it becomes a natural way of designing and implementing certain problems. Agents and agent like concepts will for sure play a part in the future. In order to be used in a commercial way, the agent platforms have to be improved. Today, platforms are fine for implementing agent-based concepts for demonstrations, mainly in explorative environments. But the set-up of connections to other agent platforms has to be simplified and the robustness of agents systems have to be increased, too.

In future, there are going to be made some interesting agent-based applications, but that will take some time, this technology is still in very young. Some application areas could be network balancing, others could be optimization and solving matchmaking problems. There will be a trend for user supporting agents, situated on personal computers and portable devices, specially mobile phones and palmtops, too.

11.2 Automated Marketplaces

One of the goals of this diploma thesis, to look at and to implement a competitive automated marketplace, was rather set to high. Such a marketplace is really complex and a lot of things have to be considered. The main challenge is the intelligence in such marketplaces, that is specially the price building on the seller side, the decision-making on the buyer side, and if more complex negotiations are involved, the negotiation strategies.

Non-competitive marketplaces can be implemented much easier. In the next years some experience in this field is going to be gained, which later could be used to establish competitive marketplaces. There is a huge field of possible applications for non-competitive marketplaces, most of them are similar to the aforementioned ones in the section about agents. This is because markets can easily be implemented with agents, probably agent technology is the way to go for markets. On the far side, a lot of optimization problems can be solved with market-like mechanism, there is demand and offer, and an equilibrium has to be found. So, non competitive marketplaces will solve a lot of problems in the future.

For competitive marketplaces, it is proposed to do more research into modeling real marketplaces, that is first on a theoretical level, instead of just implementing some special cases, that can not be used in reality.

11.3 Intelligent Software for Wireless Access

Another goal of this diploma thesis was to use a marketplace to help users get connectivity for mobile Internet access. As for the part of intelligent user support, this is going to be an important issue in the future (see chapter about related work). For the other part, to do this with a marketplace, where the user can get a customized SLA on a short-term basis, things look different.

First, a lot of the technologies used for mobile Internet access could not be used for such a system, like mobile phones, where today's technical systems (network and mobile devices) are not yet suited for such an approach. The only technologies that could be easily used for such markets, are wireless-LANs, specially 802.11b (Wi-Fi). 802.11b is entering the market for public out-of-office Internet access at this very moment (for wireless network inside an office, 802.11b is already very popular).

Second, for ISPs, there is naturally no interest in such systems, specially from the marketing department. A lot of energy is put into binding users to the own firm by companies (a typical example is the way mobile phone service providers are dealing with and apply to customers). Therefore, one of the main interests of ISPs would be to bind customer with long-term contracts.

However, the wireless access providers are even smarter than this, beside long-term contracts, they also start to make exclusive contracts for hot spot locations, only allowing them to offer service at the contracted locations (restaurants, airports, park and places). If there is only one ISP offering service at a location (monopoly...), the system proposed in this diploma thesis then again does not make sense. This exclusive contracts system is not likely to last forever, specially when the 3g mobile phone data service will become available (at a reasonable price), the customer will at least have the option to change over to mobile phone based connectivity.

Until a system like the one described in this work or one similar to it will show up is difficult to say. Sometimes, new technologies can accelerate changes in markets really fast, but on the other hand, the enterprises will care about their interests, too. Hence, for intelligent software in the field of mobile Internet access, it can be said similarly to automated markets, that competitive systems will not come in the near future, but intelligent user supporting systems will surely do.

References

[1] http://www.ethermanage.com/ethernet/faq.html

[2] http://80211b.weblogger.com/

[3] http://www.wi-fi.org/

[4] http://www.bluetooth.com/

[5] http://www.umts-forum.org/

[6] http://www.gsmworld.com/

[7] http://www.teltarif.de/i/gprs.html

[8] http://www.iridium.com/

[9] http://www.inmarsat.com/

[10] http://www.globalstar.com/

[11] http://www.ietf.org/html.charters/mobileip-charter.html

[12] http://www.ietf.org/html.charters/ipsec-charter.html

[13] Marc Danzeisen, "Secure Mobile IP Communication", 2001, Bern

[14] Jim Carr "Service Level Agreements", Network Magazine, 2001

[15] Various "Ip QoS FAQ", Stardust Technologies, 1999, Campell California

[16] Ralf Wagner "Leitfaden Volkswirtschaftslehre", http://www.wagner-berlin.de/leit.htm , 2002

[17] Philip Kotler, Friedhelm Bliemel "Marketing-Management: Analyse, Planung, Umsetzung und Steuerung", Schaeffer-Poeschel, 1995, Stuttgart

[18] Kate Reynolds "Going, Going, Gone! A Survey of Auction Types", http://www.agorics.com/Library/auctions.html , 1996

[19] Alexandra Flicker, Johann Hoeller "E-Commerce based on Internet Technologies in Procurement", January 2001, Linz

[20] Michael Wooldridge, various publications, http://www.csc.liv.ac.uk/˜mjw/pubs/

[21] http://www.ootips.com/agent-orientation.html

[22] James Odell "Objects and Agents: Is there Room for Both?", Distributed Computing, November 1999

[23] Thomas Magedanz "OMG AND FIPA standardisation for agent technology: competition or convergence?", ftp://ftp.cordis.lu/pub/infowin/docs/omg.pdf

[24] http://www.fipa.org/

[25] http://www.tryllian.com/

[26] http://sf.us.agentcities.net/aap/

[27] http://ias.comtec.co.jp/ap/

[28] http://fipa-os.sourceforge.net/

[29] http://www.grasshopper.de/

[30] http://www.agent-software.com.au/shared/home/

[31] http://sharon.cselt.it/projects/jade/

[32] http://www.java-agent.org/

[33] http://leap.crm-paris.com/

[34] http://sourceforge.net/projects/zeusagent

[35] http://www.omg.org/

[36] http://cgi.omg.org/docs/orbos/97-10-05.pdf

[37] http://www.trl.ibm.com/aglets/index_e.htm

[38] http://www.cis.upenn.edu/ bcpierce/courses/629/papers/Concordia-MobileAgentConf.html

[39] http://www.labs.fujitsu.com/free/paw/guide/paw_1.html

[40] http://www.ist-shuffle.org/

[41] http://web.media.mit.edu/˜joanie/sardine/

[42] http://www.aamas-conference.org/

[43] http://www.boingo.com/

[44] Joan Morris, Pattie Maes, "Sardine: An Agent-faciliated Airline Ticket Bidding System"

[45] Joan Morris, Peter Ree, Pattie Maes "Sardine: Dynamic Seller Strategies in an Auction Marketplace", *EC'00*, October 17-20, 2000, Minneapolis, Minesota

[46] http://www.cs.hut.fi/Research/Dynamics/

[47] http://verisign.netscape.com/security/pki/