28. Business Applications and Revenue Models

Thomas Hummel (Accenture Technologie Labs)
Steffen Muhle, Detlef Schoder (University of Cologne)

28.1 Introduction

With the maturation of its technology, Peer-to-Peer applications have come increasingly into the focus of business. The promise to do business in a faster, more cost-effective and flexible way has lead to the rise of various start-ups as well as the engagement of established market players. They have created a broad variety of Peer-to-Peer business applications and services that have come to the market. However, there has to be a viable revenue model behind these business applications and services to lead them to economic success. In other words, applications should not only be highly successful in terms of adoption rates, like Instant Messaging, but also in terms of revenue generation. This chapter contains a discussion of revenue models for Peer-to-Peer business applications.

It is certainly questionable whether there is a need for Peer-to-Peer revenue models at all. Peer-to-Peer applications often focus on the principle of reciprocity, i.e., they realize barter structures like the early version of Napster did. But this idea does not always hold, e.g., Peer-to-Peer applications, like Instant Messaging, are provided as an infrastructure. Although indirect revenues from ad space and cross-selling efforts are possible (a driving force behind the free availability of AIM is to strengthen the online community), direct revenue models are still of interest. For example, when digital file exchanges deal with the property rights of third parties and want to reimburse these third parties plus recover their operating costs, a viable business model is required.

Throughout this chapter, potential direct revenue models for Peer-to-Peer applications and the issues they face will be presented. An abstract view will be given that represents a generic Peer-to-Peer interaction and that allows the evaluation of the potential for revenue generation for the parties involved. Specifically, there will be a discussion of the following points:

1. Who are the parties that participate in a Peer-to-Peer interaction, direct and indirect, in other words, who are the players that need to be part of a revenue model so that they can recover their costs?
2. What are the open issues that Peer-to-Peer revenue models have to resolve?

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 473-489, 2005.

3. How can these parties potentially recover their cost and earn a margin of profit?

The notion of a service and application style will be introduced, different revenue models will be revealed and their relevant properties will be identified. The abstract view of Peer-to-Peer interaction will be the reference in the analysis of the application/service styles. The discussion at the end of the chapter suggests ways of dealing with the shortcomings from which current revenue models suffer.

28.2 Definitions

28.2.1 Peer-to-Peer Applications and Service Styles

From the user perspective Peer-to-Peer business applications can be classified into four different categories:

- **Instant Messaging**: applications and services for the direct exchange of messages between two or more interacting parties (humans and/or machines).
- **Digital Content Sharing**: applications and services for the exchange of digital content. Compared to simple messages, the generation of digital content is more costly and complex. Furthermore, additional functionality like digital rights management is often connected with the content.
- **Grid Computing**: applications and services that allow customers to send computing tasks to a peer that temporally hosts a server application, which manages the distribution, analysis, integrity checks and security of the data sets to other computers that can offer some processing capacity.
- **Collaboration**: applications and services to work or play in ad hoc groups that do not necessarily include organizational hierarchies. Unlike Instant Messaging, Collaboration applications support working groups particularly with regard to their coordination and cooperation.

All business applications can be provided in an application or a service style. In this chapter, Peer-to-Peer *application style* is defined as the use of software that is either provided as a packaged solution (such as Lotus Instant Messaging, Groove, etc.) or through a set of common definitions and methodologies, i. e. programming models like J2EE and .NET, protocol definitions like Gnutella and the like. The user buys the software and runs it on his own, i.e., no third party is involved. A Peer-to-Peer *service style* is a service provided to third parties which is based on a Peer-to-Peer interaction model. The software is not bought, but its functionality is leased by the user. Therefore a Peer-to-Peer application provided in a service style is not a once-bought-used-forever model.

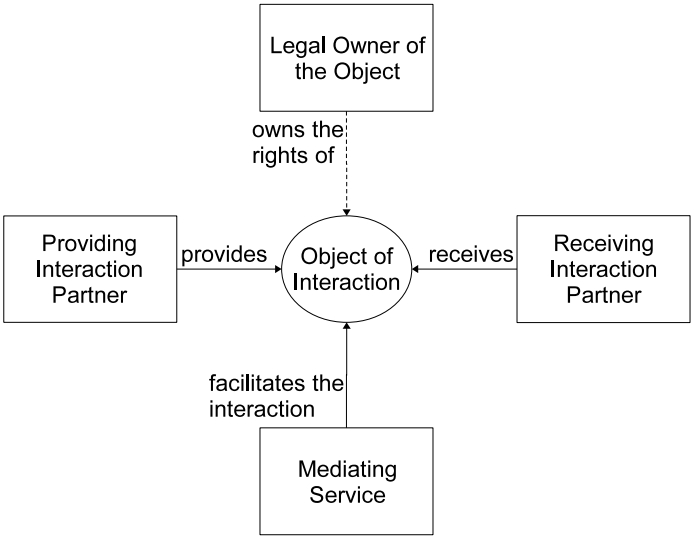

Fig. 28.1: Referential View of Peer-to-Peer Interactions.

28.2.2 A Referential View of Peer-to-Peer Interaction Styles

Looking at the broad variety of different application/service styles that fit the concept of Peer-to-Peer, it is likely that there is no one-size-fits-all revenue model for all these styles. Differentiated considerations are needed which should be guided by a common pattern. In this chapter an abstract reference view of the different Peer-to-Peer interaction styles shall be given, which can then be mapped to different application/ service styles.

At the center of the reference view (Figure 28.1) is the object of interaction. Depending on the relevant application/service style, the object can be a message, a file, a data stream, a transaction, or a data object. Different roles can perform operations on this object, which is provided from a providing interaction partner to a receiving interaction partner. The rights to the object are the property of its legal owner. A mediating service facilitates the interactions involving the object. The following overview outlines this view.

To facilitate the reading of this chapter, the elements of this abstract reference view will be referred to as the Object, Provider, Receiver, Mediator and Owner throughout the remainder of this text. It is important to note that this view refers to roles not to participants. As a matter of fact, a participant can play several roles at the same time. For example, the role of the Provider can be taken by the same participant as the role of the Owner. In specific situations some of the roles can be omitted altogether, e.g., in the case where no Mediator is required to match the Receiver and Provider. Provider and

Receiver can reverse their roles throughout an interaction sequence, i.e., the Receiver can become the Provider in a following interaction step and vice versa.

28.2.3 Business Models and Revenue Models

The terms "business model" and "revenue model" are often used interchangeably. However, for clarification distinctions should be made. A *business model* is the totality of processes and arrangements that define a company's approach to commercial markets in order to sell services and/or goods and generate profits. In contrast, a *revenue model* is part of a business model. It includes all arrangements that permit the participants in business interactions to charge fees that are covered by one or several other participants in order to cover costs and add a margin to create profits.[626]

Revenues can be generated in a direct or indirect way. In the direct way, the receipts come directly from the customer. When an indirect revenue model is adopted, the actual product will be provided for free. The gain is received from a third party who is in one way or another interested in the diffusion of the product. These two categories of possible revenue models can be reassigned (adapted from Laudon and Traver [378]):

- **Indirect revenue models** can be advertisements, affiliate models, or a bundling. When the advertising model is adopted, the third party wants to communicate a sales message that is visible during the usage. It should be considered that the message is influenced by the reputation of the advertising media. In the case of the affiliate model, the vendor receives revenue for passing customers to an affiliate partner, who sells the products or services. For every purchase, the vendor receives a commission. The bundling model is similar onto the affiliate model. The revenue is generated by products or services that are connected to the free offering. Even though the basic service is free, the customer has to pay for the additional offering.
- **Direct revenue models** are sales, transaction fees, and subscriptions. In the sales model the customer pays directly for the object. Two forms have to be distinguished. In the first one, licenses for the Peer-to-Peer business application itself are sold. This model corresponds with the application style. The second form is to use the application as an infrastructure to sell goods or services. If the transaction fee model is used, the company will be assigned the role of the mediating service. It provides a service that facilitates a transaction between the Provider and the Receiver. The Mediator earns a fee for every transaction between them. Unlike the transaction fee model, in the subscription model the fee is paid independently of the actual usage. It is rather paid periodically, e. g. every month or year.

Peer-to-Peer application styles and Peer-to-Peer service styles are not mutually exclusive, e.g., a vendor might sell licenses for the use of specific software and add some specific professional services for which he charges service fees. A Peer-to-Peer software vendor needs to come up with a licensing model, whereas a service provider has to develop an approach for the billing of the service. There are several prerequisites that a viable revenue model needs to fulfil. As an absolute minimum, two important features are required:

- **Differentiated Charging**: A viable revenue model needs to be capable of covering costs and earning a margin of profit. However, in order to be efficient a revenue model needs to provide a differentiated pricing structure, i. e. the capability to charge according to criteria that are tied to the patterns of usage.
- **Allocation Effectiveness**: The capability of a revenue model to create a revenue stream to the appropriate Receiver, i.e., to the party that has incurred the cost, is called allocation effectiveness.

28.3 Revenue Models for Peer-to-Peer Business Application/Service Styles

In the following, the interaction style will be mapped to the reference view. Thus, potential revenue models and issues in the application style setting as well as in the service style setting can be examined.

28.3.1 Instant Messaging

Instant Messaging (IM) refers to applications that enable two participants to exchange information in real time. Typically this is done through typed text messages, but various Instant Messaging applications have extended the functionality to include voice messaging. Although the capability to exchange files has practically become a standard feature in most messaging applications, it will not be discussed here but in the following section on Digital Content Sharing.

Applying the reference view introduced above, the Object is a message and the Owner is identical with the Provider, i.e., the sender of the message. The software usually provides some specific functionalities, like a "buddy list", which provides presence services, i.e., a list of currently logged-on and thus reachable interaction partners. This service, that often includes various other functionalities like finding people based on their profile, does not have to be central. Actually, three primarily different topologies exist which influence possible revenue models for instant messengers. The first one is not really Peer-to-Peer from the technical point of view: In this topology the

buddy list service as well as the actual messaging service is provided by a server. However, the communication takes place between two autonomous peers. In the more decentralized design the buddy list is managed by a server but the communication is self-governed by the communication partners. In both cases the Provider of the service acts in the role of the Mediator. As long as the central software is used within the boundaries of a corporate network, the service infrastructure Provider is the internal IT department (or the outsourced equivalent). Once the software is used across organizations, connectivity between the users is required and this can either be an ISP or a Provider of an IM service. If the pure Peer-to-Peer topology is adopted both (the message exchange and the buddy list service) are organized by peers, so that no server is involved. One example for the latter is Skype.

The revenue model for an application style is rather straightforward and can be built around license fees as well as around optional professional services (integration support etc.). There are various models for license fees (e.g., base fee and yearly maintenance fee, upgrade fees, etc.). The vendor faces the challenge of opting for the best one to generate sufficient revenues – always assuming that the application provides enough value to attract buyers and that it generates sufficient revenues to offset costs plus provide a margin of profit.

The situation becomes significantly more complicated when IM is delivered as a service style. Currently the most popular IM services on the Internet are AIM, ICQ, MSN, and Yahoo!. Today these services are available for free. It is sufficient to download and install a client application and attach it to the network of the respective IM service provider. Highly successful services like AIM claim to have 200 million users [411]. Although these figures refer to the number of accounts rather than the real users, Instant Messaging has unquestionably been a huge success in terms of adoption rates. It is clear that a substantial infrastructure is needed to provide the users a stable and convenient service, and obviously there is cost associated with that. Providing the service for free is a viable revenue model only if an indirect one is adopted (one of the main reasons for AOL to provide AIM is to attract users to its ISP service).

An undifferentiated subscription fee for using the service is the simplest possibility. However it is not a very efficient one because it can hardly account for differentiated charging. Heavy users and occasional users would not be discriminated and in the end the difficulty is to strike a pricing scheme that satisfies the occasional as well as the heavy users. Various modifications to this approach could be suggested like a fee per log on, a fee per log on and day, or a fee for the time during which a user periodically connects the mediation server. A periodical connection is necessary for the user to keep his buddy list up-to-date and to notify the server that he is still online. It is clearly more interesting to take a transaction fee and bill for the actual usage, but that is anything but simple. Where IM services are really built

as Peer-to-Peer applications (e.g., ICQ [414]), the communication between two participants is direct – i.e., there is no application server which connects them for the message exchange. The only possibility for a third person to find out about a message transfer is by monitoring the network providers' hubs (on the network level). This makes charging per usage quite difficult – a billing system would need to be able to get hold of, count and attribute network packets to specific users. If the architecture of the services is realised in pure Peer-to-Peer style, which is totally without a mediation service, even the realization of a pay per time model is problematic. However, where IM services are built in a client-server model (like MSN [414]) this problem is minor, because the application server has control over the conversations and because it can log the usage.

New problems arise with the emergence of interactive agents. Although these agents have not found a wide diffusion yet, they can provide additional services for the user of Instant Messaging. To put it simply, interactive agents are pieces of software accessible through the IM infrastructure as if they were real users. Advanced agents such as Active Buddy even provide interaction with a rather sophisticated natural language dialogue [130]. Such interactive agents enable the execution of transactions over the Instant Messaging infrastructure in a way which is very convenient for the user [439]. These agents deliver for example important news or act as a text based service hotline, which can help to solve customer problems. It is possible that they will play a role in the currently emerging web services. IM services today have only weak possibilities to deal with interactive agents on their networks. Even if they succeed in detecting them (which is already only possible if an interactive agent does not behave in the similar way as a real IM client), they face the challenge of understanding what exactly the conversation is all about. The trouble is that a charging approach that builds on the amount of information exchanged is not a good fit with these agents. Human-human interaction is often chatty - a human-application interaction which aims at performing a transaction should be expected to be fairly brief. As long as the IM service provider has no means of differentiating between a transaction and a conversation, it has no means of performing any type of differentiated billing either. Appropriate charging is only possible if the IM infrastructure service provider owns the interactive agent or has at least an alliance with the Provider of the agent. Microsoft has good reasons to weave its MSN Messenger deeply into the .NET services infrastructure and build it on a client-server model. It remains to be said that three popular IM services go into the market of the application style IM or have already failed in doing so: MSN already bundles the MSN client with Windows XP and sells an application named "Live Communications Server", whereas AOL announced that it would stop developing "AIM Enterprise" in order to cooperate with IMlogic. Yahoo! dropped its messenger because instead of buying the business version, the free version was being used by companies, as Yahoo! stated [345].

Judging from the criteria that have been defined for viable business models above, the application style allows for a differentiated charging through various licensing models and fulfils the criteria of allocation effectiveness (as long as the software is not illegally copied). However, when it comes to a service style, revenue models for Instant Messaging face quite significant challenges with respect to a differentiated charging structure. Allocation effectiveness is not a problem for centralized and hybrid Peer-to-Peer architectures since IM clients need to log on to the service provider's network. However, this becomes a problem in pure Peer-to-Peer architectures. In this case it is rather difficult to trace the communication between users in order to bring it to account.

28.3.2 Digital Content Sharing

File sharing has become infamous through the quick rise and fall of Napster. Although Napster collapsed with its first model, its place has been taken over by others such as KaZaa, Morpheus, Grokster, and eDonkey. The music industry is amidst a ferocious fight against the free exchange of music by submitting empty music files or files that contain only part of a song plus some affixed commercial content. Beside these technical weapons, the music industry threatens users and service providers with lawsuits. By means of rigorous penalties the users should be discouraged from sharing illegal media files. For example, the RIAA made a settlement with four students in the USA who ran a service for searching MP3 files in their college network. The students pay penalties of between $ 12,000 and $ 17,000 each [83]. In Germany, a trainee settled to the amount of EUR 8,000 [478].

Whatever the fate of the various file sharing applications and the communities behind them will be in the end, they have shaken up the value chain of the music industry and they might well lead to changes in business models [310]. Although such digital media exchanges have become the best-known examples, they should not be mistaken for the only possible instantiation of this application/service style. Two points are important here:

1. The exchange of entertainment media files is only one specific type of content exchange. Any other digital content can also be exchanged, e.g., design documents, training documents, reports etc. Thus, Digital Content Sharing can be used as a decentralized form of knowledge management. If an adequate index service exists, the documents containing the knowledge of the network participants can be accessed without forcing them to save these documents on a central server.
2. The definition needs to be broader than just file sharing. It should include streaming content as well since this type of content can also be recorded and exchanged in a Peer-to-Peer manner (such an extension to include streaming content clearly leads to specific challenges in the technological

implementation, but it does not change the considerations for revenue models).

In terms of the reference view introduced above, the roles of the Provider and Receiver are obvious. But the most important point is that the Owner is not necessarily the Provider or Receiver – he can be a third party. Depending on the specific implementation of the file exchange, there can be a Mediator providing a central catalogue of contents available as well as providing additional services, even though that is not a necessity for digital content exchanges. Napster, for example, was essentially a Mediator; it ran a central directory to which the client's software connected in order to match the Sender with the Receiver. In this case, only the exchange of the files as such was performed in a Peer-to-Peer manner. Peer-to-Peer Digital Content Sharing applications that are built on the Gnutella protocol do not need a Mediator. They work on a pure Peer-to-Peer model even for matching Sender and Receiver [396].

As an application style, software of Digital Content Sharing can be implemented in corporate networks, e.g., for the facilitation of knowledge exchange between employees. Peer-to-Peer collaboration tools also typically provide such Peer-to-Peer file exchange functionalities. Again, the vendor of such software can adopt various license models and provide consulting services for implementation, training, and operations to create additional revenues. The application style is not limited to the borders of a single corporation. Various Digital Content Sharing communities can be connected across several organizations in a more or less ad hoc manner. Furthermore, a mediating service Provider could act in such intercorporate Peer-to-Peer exchanges by providing catalogue services for the content supplied by different organizations as well as other value added services. This can be the software vendor, the participating organizations, or a third party. It brings us to a service style type of Digital Content Sharing. If the Provider is also the Owner, then the revenue model is fairly straightforward and the Provider can charge a fee for the transfer of the object.

Things become significantly more complicated if Digital Content Sharing applications are operated in a service style and if the Owner is not identical with the Provider. The most prominent case is Napster, which made its market entry as a free service and allegedly intended to create a revenue stream from advertisements. In principle, other revenue models would also have been possible, like membership fees for subscriptions, log in fees, or even a match making fee. However, given the legally precarious situation of the whole approach, such revenue models would have been debatable anyway. Today there exists no serious Peer-to-Peer service for distributing legal content. Whatever the real business idea of the makers of Napster and the look-alikes is, the real problem is that the legal owner of the rights of the exchanged files is not a participant in the transactions and hence he can not cover his costs. BMG, EMI, Sony, Universal Music and Warner Music never participated in

the original Napster transactions, and the BMG Napster did not do more than stop the illegal activity. For digital content exchanges that embrace a Mediator approach, one could think of implementing a billing step into the content exchange, where any or both of the participants are required to pay fees for exchanging content and those could then be paid to the Owner. If, for the moment, it is assumed that this would be technically feasible, then the Mediator adopts the role of an aggregating middleman for the content which can be conveniently searched and compiled.

But it is questionable whether anybody other than the Owner should be involved as a Provider. Why should the content first be bought from an Owner and then be sold to another Receiver when at the same time the Owner needs to be part of that transaction again and he needs to be properly reimbursed? Even though this distribution model could lead to technical advantages, its economical benefits are not clear. It would be easier for the Receiver to buy the content directly from the Provider. One might argue, however, that the intermediary function can add additional value: Today's digital content exchanges integrate usually the recordings from different music groups which would not be the case with separate download sites. But that is hardly convincing – the music industry could run a joint catalogue service without major problems since the artists are regularly bound through exclusive contracts. Finally, there is the question of control. Digital content exchanges perform an unbundling of content and provide a possibility for free reassembly through the user: Rather than buying a complete album, consumers can buy selected titles only and create their own specific albums to their tastes. The music industry and artists alike have good reasons to be reluctant to agree on the unbundling and recompilation of the content. It is rather difficult to determine an appropriate price for popular as opposed to less popular titles. Apart from that, a full CD can be sold for a higher price than, e.g., the three popular titles only [125, 604]. Anyhow, it seems that the music industry cannot surrender the market demand for digital content. Several distribution services have started up recently, e.g., the Apple iTunes store. But nearly all of the serious upcoming Providers are based on a client-server architecture because more control over the distribution process is guaranteed. In short: taking an additional party into the transaction simply because he or she happens to have the digital content at hand does not add any clear economic value. It is more reasonable if the Provider once again becomes identical with the Owner, in other words, the record companies sell the content themselves or with the help of a few centralized licensed sellers. Then it is likely to look more like iTunes, which follows a client-server-based approach rather than a Peer-to-Peer exchange.

Finally all these considerations will only hold for Peer-to-Peer exchanges if a *billing scheme* can be built into the digital content exchange and if the fees can be allocated accordingly. If no Mediator is involved, e.g., if a digital content exchange is built on the decentralized Gnutella protocol, the enforce-

ment of a payment will be very difficult to achieve since interactions are not centrally managed in the first place. Furthermore, there is always the possibility to copy and forward the contents once a Receiver has them. A billing scheme, even a well-built one, is not a content protection scheme, and even if contents are bought legally once, they can be illegally copied infinite times thereafter. Consequently, any (legal) revenue model will need to provide for the *protection of digital rights*. In addition, it will need to *enforce the payments* to the appropriate legal owner. As simple as this may sound, there are difficult technical challenges to solve, and the question of whether digital content can be protected at all is still open [267]. The failure of the SDMI initiative is quite an instructive example. The music industry asked the Multimedia Telecom Italia Lab to create an encryptable and thus a safe digital music format. The lab agreed and promised the amount of $ 25,000 to those who managed to break the code. When the group of people that had managed to break the code wanted to publish their accepted paper, the music industry filed for a court injunction and the paper was banned from publication. Prof. Edward Felten from the University of Princeton put the paper on the Internet, where it can still be downloaded [136]. Another problem with digital rights protection systems is that the Providers always have to make tradeoffs between the security of such systems and the usability. If the protection system is too restrictive, then it can deter potential customers from buying the media file. Whether the music industry will win the fight against digital content exchanges is an open question. In the end, it might be wiser for the music industry to adopt a new business model instead. Rather than trying to crush the online exchange communities, the music industry could try to develop them and own the communities [310]. This way, through word of mouth advertising, marketing affects can be achieved. This applies especially to smaller labels and independent artists.

If these considerations are put in the context of the criteria for viable revenue models introduced above, it can be stated that there is currently no viable revenue model for Peer-to-Peer digital content exchanges in service style and it is likely that there will never be one. The point is that it is very difficult – if not impossible – to enforce allocation effectiveness when Me is in the game. Digital content exchanges, where the Owner is identical with the Provider, face comparable issues if the Provider does not sell the Object itself but if he only sells limited rights to its usage. As soon as digital content is provided to a Receiver, it is vulnerable to copying and handing over to third parties. Differentiated charging, on the other hand, is easier to achieve since digital content exchanges could be priced individually. The real issue to resolve for digital content exchanges is its protection at such a level that the customers are restrained from infringing the copyright. If protection needs to be given up, new approaches will have to be found for the business models.

One example is the approach of Marillion, a group that has a long history of releasing records the traditional way. It has built up a faithful community

of fans over the years. When Marillion's contract with their record label expired, they decided not to renew it. Instead they promoted the new record using the Internet. The band members wrote to their fan base and asked if they would be willing to pay for the CD in advance in order to finance the making of the record. The response was overwhelming. Some fans even offered to pay for two CDs "if that would help them". When the production was finished, the CD was offered through the band's web site – not for download, but for ordinary purchase through a secure web link. It remains to be said, however, that there are other examples of such approaches that failed, e.g., Stephen King's experiment with an online book. A recent remarkable approach without protection is addressed by the Potato System [493]. This system tries to induce the users to license the media files by offering them the possibility of earning a commission. When a user registers a media file and hands this file to a friend he gets a commission if his friend buys a license as well. So the users are motivated to license and to recommend the file. But whether this model will work is questionable. Due to the absence of copyright protection, the Owner has no chance to enforce his rights. It is likely that this will be the main obstacle for music labels participating in this model.

28.3.3 Grid Computing

Grid Computing refers to the coordinated utilization of distributed computing resources [544]. Of all Peer-to-Peer applications, Grid Computing is the one that least fits the definition of Peer-to-Peer computing in its contemporary realizations that are suitable for industrial use. Their computational model is regularly client-server-based rather than decentralized. None of today's models work without a special server that manages the distribution, analysis, integrity checks and security of the data sets. Peer-to-Peer in this case refers to the idea of having complex problems broken down into parts that are solved more or less independently by peers and that are then put together to one solution. In doing so peers act autonomously to a large extend. They can choose, if and when they are willing to provide their computing power. Examples for applications that use this computing model are SETI@Home [28] and MoneyBee [426]. However, if the vision of Grid Computing in its pure understanding should become true, peers should not only be able to provide computing power, but to demand the resources available in the network.

The point of Grid Computing is more about the work getting done by peers than about the organization of the work [178]. Grid Computing has not been very commonly used so far. Early applications in this area have focused on the aggregation of computer processing cycles to solve complex mathematical problems. Future applications can be expected in various areas, such as Biological and Chemical Engineering, Economics and Financial Modelling. It is interesting to see that IBM's patent on distributed comput-

ing technologies was ranked as one of the five top patents to watch by the MIT technology review in 2001 [583].

If Grid Computing is mapped to the reference view, the Provider(s) can be interpreted as the one(s) providing available computing power and the Receiver as the one using this computing power to solve complex problems. The providing interaction partner can as well be seen as the one who provides the task to be processed. In this chapter, the focus lies on the computing power because of the interest in its payment. The Mediator is the central server application, which manages the distribution, analysis, integrity checks, and security of the data sets.

Dougherty et al. have distinguished four different revenue models for Grid Computing: the enterprise software sale, the public Internet exchange, an application service provider (ASP) and a B2B exchange model, though the ASP and B2B models "have not yet developed and may never develop" [178, p. 114]. The enterprise software sale model is identical with the view of an application style, i.e., the revenue model is about selling distributed computing software for installation behind the firewall or "enterprise grid". The rationale is to provide more control over the contributing resources which will lead to higher availability and better security. Apart from that, the LAN/WAN capacities typically allow the transport of much larger data sets. A revenue model is typically straightforward and consists of license fees and professional services for implementation.

The public Internet exchange or "mixed grid" approach is of the service style. The idea is to provide access to vast computing power on a worldwide scale. An example is Moneybee where Grid Computing is used to predict stock prices and exchange rates. Participants download and install a free screen saver that uses the idle PC resources when the screen saver is on in order to perform complex operations that are downloaded from a Moneybee server. Thereafter results are uploaded to the Moneybee server [426].

With respect to revenue models, Grid Computing is a quite different situation than the other Peer-to-Peer interaction styles. As far as the Mediator is concerned, there is a need to distinguish whether the service manages the Grid Computing tasks on behalf of a third party or whether the mediating service is identical with the Receiver. If the work is done on behalf of a third party (which corresponds to the ASP model in [178]), the cost for the mediating service plus a margin will need to be charged. If the mediating service is provided by the Receiver, then the business utilization of the grid computation results will have to cover the cost. In both cases, the issue is how to determine the computing cost per Provider and how to compensate the Providers. Mediators currently ask users to donate their excess resources. In exchange, they offer a portion of these resources to non-profit organizations, or else they provide "sweepstakes entries" for prizes. The Providers at Moneybee contribute their resources free of charge. Their incentive is to get part of the results (forecasts for stock prices and exchange rates) that the system

generates. It may certainly be that many other Grid Computing tasks are taken on by Providers in a similar way for free (e.g., when they are bundled with attractive add-ons like the lively graphics of a screen saver hosting the client application as SETI@home does). What, however, if the Providers want to be reimbursed monetarily for offering their excess resources? From the technical point of view, it is not very difficult to employ a pay-per-use model where the client application records resource usage and provides the data to the mediating service which then reimburses based on usage. The real problem is that the price paid for the resource supply is likely to be rather low and (micro-) payments will need to be organized in a very efficient way, if the transaction is not to use up all the benefit. It is doubtful whether financial incentives will be capable of attracting a sufficient number of Providers – at the end of the day a non monetary incentive seems to be the better idea.

In summary, even though Grid Computing probably has the most straightforward revenue model of the core Peer-to-Peer applications, it still faces the challenge of creating enough business to generate sufficient micro payments to attract a sufficient subscriber base. Judging from the criteria for revenue models, there are no problems regarding the allocation effectiveness or efficiency. The questions is whether Grid Computing will create sufficient business value to earn its own living, once Providers want to charge for the use of their resources.

28.3.4 Collaboration

Collaboration supporting Groupware applications have been around for about fifteen years with differential success in providing functions beyond email and workflow. Typically groupware functionality includes some of the functions of email, chat, bulletin boards, calendaring/scheduling, file sharing (push and pull models), and search. Today it is available in different facets of existing client-server tools. Peer-to-Peer concepts can add some flexibility that client-server-based groupware lacks today, such as facilitated personalized categorization of data and information, and the creation of ad hoc working groups across organizations [560]. In the following, Groupware applications that are used in the business context, will be considered. In this context the workgroups are composed of defined and authenticated members. For that reason, generally no problems arise in respect of copyright protection. The files are not shared blindly, but are given to the other group members to enable cooperation.

Like Instant Messaging and Digital Context Sharing, Groupware applications can be built according to three major topology types: Within the first, communication is handled by a server. Thus Peer-to-Peer in its pure conception is not achieved. If the second topology is adopted, the communication will take place between the clients, but a server gives additional services. Within the third, no server is involved at all. According to the reference view, the Me-

diator can be the server which facilitates the communication or which offers additional services, depending on the topology type. The Provider and Receiver are the communication partners of the workgroup, whereas the Object is the message or document that is exchanged between them. As described, the Provider generally is the legal owner of the Object or is at least authorised to hand it on to the Receiver.

The method of selling groupware as an application style is unlikely to change with new architectures underlying the software. Various licensing models as introduced in Section 28.3.1 can be used. A special opportunity arises from the complexity of groupware applications via the aggregation of different functions and their high integration in daily work. It can be assumed that a comparatively high demand for professional services exists.

If groupware applications are hosted and brought to the users in the form of a service style, the above considerations for the core applications of Instant Messaging and Digital Content Sharing can be carried forward. A transaction-based billing can only be arranged if a central instance that can observe the usage of the service is used. In the case where the communication is held by a server, a transaction-based fee can be accounted for by the amount of transferred data or the usage time. Otherwise, only the usage of the services provided by the server can be brought into account, e.g., the catalogue service for members and files, memory to store files of temporary offline peers, or security services such as logging. Whether it is possible to adopt the service style in the case of a completely decentralized architecture is questionable. In this case, the considerations of Instant Messaging apply. It should be added that the Provider of a Peer-to-Peer collaboration service style should consider whether he wants to bill for every user or for a complete group. It is inherent in collaboration that the work between two members can also benefit the other group members. So it seems adequate to leave the choice of accounting method to the customer.

With respect to the criteria for revenue models the danger is that allocation effectiveness cannot be ensured. If the infrastructure employs a real Peer-to-Peer model, the revenue model will face efficiency challenges. It is the bundling of various services (such as IM and File Sharing and other, potentially non-Peer-to-Peer services) that makes groupware interesting. Revenue models for groupware service styles can then be built around various other criteria. The Peer-to-Peer functionality is only one of them.

28.4 Discussion

Summarizing the considerations, it can be concluded that revenue models for Peer-to-Peer application styles are not different from revenue models for traditional application styles. Revenue models for Peer-to-Peer service styles face important challenges, especially in the areas of Instant Messaging and Collaboration (where differentiated charging is difficult to achieve) as well

as in digital content exchanges (where allocation effectiveness is difficult to achieve). Because Instant Messaging and Collaboration Providers only make infrastructure available and because digital content exchange affects copyrights belonging to third parties, these three applications require an accounting centre. However such a centre is difficult to build into Peer-to-Peer architecture. That is not the problem with adopting revenue models for Grid Computing, which least faces issues of the core Peer-to-Peer applications. The transaction only takes place between Provider and Receiver, so that a central instance has not to be paid. But still, the overhead of administering (micro-)payments is likely to diminish the attractiveness of these revenue models.

The open question is what kind of strategies the different parties (Mediators, Owners, etc.) could employ to increase their revenue. Until now, the focus has been on direct revenue models for specific Peer-to-Peer interactions, in order to understand what kind of issues charging and payment mechanisms they will face. The focus on a revenue model for just one specific interaction, as initially considered, is a rather tight restriction. As a matter of fact, real world examples of Peer-to-Peer applications follow more a bundling approach. Examples of service models are AIM as a means of fostering the AOL online community and the close integration of MSN in the .NET infrastructure. The benefits of bundling digital information goods as well as price discrimination are well known [49, 50, 604] and the various Peer-to-Peer areas could very well take advantage of this approach.

Instant Messaging services could be bundled with services which use the existing infrastructure and which extends the basic services. One possibility is to provide additional services through interactive agents. IM services would then give access to agents that provide various services. These services can be rather rudimentary, like information services (access to news, stock quotes, and the like) or advanced, like a secure access to bank accounts, hosted personal calendars etc. This does not mean that the IM service provider controls the bank accounts or calendars. It is sufficient that it provides a ubiquitous infrastructure that conveniently integrates with applications of those providing the services in the first place (e.g., banks). These services will become even more valuable if they are combined with contextual information. For Example, mobile users could be given instant information based on their current location through an interactive agent. These services fall under the label "Location Based Services". IM service providers would need to cooperate with the companies offering these services to set up the required applications and infrastructure as well as to provide convenient yellow pages so that the IM access can be found easily by anybody using the IM services. There are interesting opportunities for price discrimination as well. Another possibility for bundling is shown by Skype. The Luxembourgian company offers a fully decentralized Peer-to-Peer Instant Messenger providing a VoIP infrastructure that allows making telephone calls between peers for free. The beta-version

of Skype was made public at the end of August 2003. Due to the easy installation process, the easy-to-use user interface and the good sound quality, Skype could – as it stated – adopt 10 million users in the first year, with more than 600,000 users having logged on on average [199]. So, it was successful in adopting a huge user community. This community gives Skype a user base to establish a revenue model based on a costly service called SkypeOut. The latter started at the end of July 2004 allowing users to make prepaid calls in conventional telecommunication networks. It is conceivable that Skype could use its VoIP infrastructure for further services, e.g., for a radio program or for music distribution. For both IM bundling revenue models, the very basic and rudimentary IM services could still be free (not the least since even a small fee can put communities that have grown accustomed to free-of-charge using at the risk of breaking apart). Additional services could be charged on a pay-per-use basis. Premium services for secure access (e.g., for connections to bank agents) could have a base subscription fee.

When it comes to *Digital Content Sharing*, it is currently unclear how the battle between file exchanging and the music (or other, e.g., film) industry will finally turn out. But as described above, it might be a better strategy to try owning the communities. Clearly, owning a community would hardly be possible if the participants of that community were required to pay for something that they could get for free somewhere else. Once again, the way to make such communities work would be to bundle the digital content with other information goods that are not easily available through illegal content exchanges. Examples would be reductions for concert tickets, fan articles that could be ordered exclusively through the community, chat services with the artists, competitions where authentic belongings from the artists can be won.

Bundling is not really a remedy for *Grid Computing* revenue models. If the transfer of the (micro-) payments generates too much overhead, then revenue in the sense of a pecuniary compensation might not be the way to go. However, providing information goods as a reimbursement could be feasible. For example, the screen saver running the distributed computing task might not just be made of lively graphics but it might provide an information channel, such as Moneybee does (even though Moneybee is not Peer-to-Peer) [426]. These information channels can report parts of the grid-wide computed results news independent of the computing task. At the end of the day, that approach brings the revenue model back to barter-like structures.

Services for supporting *Collaboration* have possibilities for bundling similar to IM. Basic services like document handover and communication can be provided for free. Additional services, which fall back on central components, can be charged per use. These central components can be especially catalogue, buffering and security services. These services are not required for smaller workgroups and Collaboration tasks, but they might become essential with rising requirements.