Doing Business in the Information Marketplace

— A Case Study —

Jörg P. Müller*, Markus Pischel
John Wiley & Sons Ltd.
International House
Ealing
London W5 5DB

Abstract

Fueled by the rapid development of the Internet and the World Wide Web, electronic commerce has become one of the fastest-growing areas within computer science and within intelligent agents research, in particular. In this paper, we approach electronic commerce from the perspective of an industrial vendor of electronic information services. The underlying model is that of an *information marketplace*, with producers, vendors, facilitators, and consumers.

The contributions of the paper are fourfold. First, we describe a novel model of electronic commerce that is particularly geared towards operationalization of vendors' business models. Second, we describe the architecture of a vendor service in the information marketplace that implements the electronic commerce model. Third, the paper provides a case study of a large industrial online information service developed based on this architecture. Fourth, we outline requirements and possible benefits of the usage of intelligent agent technology in a commerce-enabled digital library.

1 Introduction

The advent of the World Wide Web has provided the technological basis for the large-scale, ubiquitous deployment of information services and, subsequently, of electronic commerce. Increasingly, end users can buy a wide range of information services, but also "hard goods", such as CDs, computer hard- and software, groceries, directly via the Internet using nothing more than a web browser. At the corporate level, companies do business with departments, partners and suppliers through Intranets and Extranets.

The open, distributed, and dynamic environment within which these developments take place, creates a wealth of technological challenges: On the end users' side, tools and techniques are required that facilitate robust and personalized resource discovery, matchmaking [KH95] [Fon97], information retrieval, and information filtering [MZ97]. Producers and vendors of information services need tools that enable them to define flexible business models, to provide a basis for negotiating and setting up business relationships with their customers, and to provide end users with more user-friendly and personalized shop fronts.

Over the past few years, two research directions within the agents community have covered different aspects of these problems. Research in *Digital Libraries* [LMPS95] [Bir95] [FMPW97] [AD98] investigates extendable, scalable multi-agent architectures that support the dynamic definition of new *library services*. The University of Michigan Digital Library group (UMDL) is the most prominent example of this type of research within the NSF Digital Library initiative [NSF98]. Durfee et al. suggest the notion of an information marketplace, consisting of different types of agents (producers, brokers, consumers) requesting services from each other. In the spirit of *market-oriented programming* [Wel95], UMDL use the market-based approach for dynamically allocating service requests to service providers, rather than as a framework for enabling electronic commerce.

A second research direction in the agents community has developed models and systems for agent-based electronic commerce. Some approaches (e.g., AuctionBot [WWW98], Tête-à-Tête [GMM98], and KASBAH [CM96]) investigate scenarios in which sellers and buyers negotiate by using classical auction protocols (e.g., Dutch or English auction). Others, e.g., SHOPBOT [DEW97], provide comparison shopping agents that analyze web pages in order to find out about and compare the prices of specific classes of goods[1].

We were looking at these approaches from the perspective of a publisher involved in electronic publishing and electronic commerce. From this perspective, we found most of the above approaches either too general to be used as the basis for a commercial electronic commerce site, or based on ad-hoc developments. In particular, most agent-mediated electronic commerce models focus on the consumers side (see e.g., the CBB model described in [GMM98]); while this is clearly important, it does not provide a micro model of electronic commerce that can be used for a vendor of library services to define criteria for accessing, charging, and billing for library services on the World Wide Web.

In this paper, we describe a model and framework to address this topic; in particular, we start from the model of an information marketplace and extend it by a vendor service. This service fulfills three purposes:

1. It encapsulates a general electronic commerce model, as well as tools that allow a vendor to flexibly set up different business models;

*Email: jpm@wis-dev.wiley.co.uk

Autonomous Agents '99 Seattle WA USA

[1] For a recent survey on agent-mediated electronic commerce, see [GMM98].

2. It provides the basis for providing a human user with personalized shop-front services;

3. It provides a uniform interface that is the basis for interaction with other agents in the information marketplace, such as automated negotiation and comparison shopping.

This service is the basis for the development of the commercial electronic publishing / electronic commerce site WILEY INTERSCIENCE. Currently, this site provides millions of end users (most of them organized in institutions like universities and large corporates) with personalized access to about 400 scientific journals. From the business perspective, access to journals is to be defined according to a set of business models.

While it seems obvious that that the overall value of an electronic commerce system for the information marketplace should increase with the number of vendors involved, it is a practical requirement that the system must add sufficient value even if only a single vendor is using it. Thus, we develop the system in a bottom-up fashion: We start with a micro model, i.e., a single vendor scenario, while keeping macro requirements for the information marketplace in mind.

This paper is structured as follows: In Section 2, we illustrate our domain by a number of scenarios and outline requirements on an electronic commerce model for the information marketplace. In Section 3, we describe a electronic commerce model that meets these requirements. Section 4 gives an overview of the WILEY INTERSCIENCE system built on top of this model. In Section 5, we discuss how agent technology fits into our work, and where it can provide benefits. The paper ends with conclusions and an outline of future work in Section 6.

2 Electronic Commerce: Scenarios and Requirements

In this section, we outline requirements for an electronic commerce infrastructure or system enabling the information marketplace. We start by illustrating four example scenarios from our domain; then, we analyze the resulting requirements.

2.1 Four scenarios

Scenario 1: Subscription model Today's business model in scientific publishing is very much subscription-centric. Customers or organizations, e.g., university libraries, subscribe to specific journals for a price, and the customers or organization members take away and read the subscribed journals. Translated to electronic publishing, this provides a basic on-line subscription model where customers or organizations purchase the permission to access the journals on-line, and where the customers and organization members can access articles, reviews, abstracts etc., via a web browser. At the time this article has been written, most existing commercial sites of scientific publishers are instances of Scenario 1.

Scenario 2: Information channels In this scenario, publishers provide diverse channels of information, and customers can subscribe to these channels. Information channels can be subject-specific and will contain multimedia content. At the end users' side, the presentation of this information will be supported by a variety of tools. The information channel model is very similar to PAY TV. Today's push channels such as MARIMBA can be seen as a first step towards this business model.

Scenario 3: The virtual journal The second scenario dilutes the strict subscription-based business model of Scenario 1. In particular, it provides a more fine-grained definition and flexible packaging of entities that a publisher might wish to sell. In this example, a publisher offers a customer access to ten articles from up to five different journals, thus defining a *virtual journal*. This access can either be prepaid (so-called *metered access*) or on a pay-per-access basis. Currently, Scenario 3 can be found in financial publishing, where customers pay on-line for individual reports; it is not common in the scientific publishing community.

Scenario 4: The personal journal The previous scenario enriches the basic subscription model by more flexible packaging. However, the packaging was static and hand-coded by publishers. As the content of a virtual journal changes based on who is looking at it, it turns into what we call a *personal journal*. The degree to which this is automated may vary from customer selection (the customer explicitly selects the journal content) to automated selection and presentation of articles based on a user profile. Today's scientific publishers' web sites offer rudimentary personalization services, such as simple personalized homepages. Publishers have recognized that providing better, added value services to their customers will provide them with a crucial competitive advantage.

2.2 Requirements

Scientific publishing in its present form is very much Scenario 1 with a trend towards the more complex scenarios. It is an open question in the publishing industry how Internet electronic commerce will develop and what business models and services will be required to make a business in the information marketplace. Basic requirements on electronic commerce systems include flexibility, extendability, scalability, and performance. In addition, there are some specific requirements from the perspective of a vendor in the information marketplace:

Support roles and relationships: To represent the various players in the information marketplace (consumers, facilitators, producers, vendors) as entities, and to represent their relationships;

Effective access control: To ensure that authorized users will get access to documents, while access is denied to unauthorized users.

Usability: To provide vendors with a competitive advantage through added-value, highly usable shop front services, such as electronic sales assistant, service brokers and matchmakers, and smart search, or personalized presentation;

Decision support: To provide producers and vendors with access to information and tools that allows them to validate their current sales and business models, and to find out what might be beneficial changes, while maintaining users' privacy;

Automation: To facilitate automated generation of product offers and of the negotiation of prices and terms of service between vendors and services.

3 The Electronic Commerce Model

In this section, we present a model for electronic commerce that complies with the requirements identified in Section 2. The primary purposes of this model is to enable vendors in the information marketplace to package and sell information and information services, and to describe interactions between vendors, customers, and end users. We define the basic entities and roles in the model, and define the operations that it provides.

3.1 Basic concepts

Our model is based on four basic roles, i.e., *vendors, publishers, customers,* and *and users,* as well as four fundamental entities, i.e., *resources, actions, products,* and *licenses.* While these entities describe objects at the micro-level of our electronic commerce system, roles describe abstract entities at the macro-level; they denote sets of abstract behaviors that can be adopted by players in the information marketplace. A particular player can take more than one role at a time: e.g., a publisher can both be producer and vendor of information services, or it can only take the role of a vendor, and re-sell services produced by third parties. Figure 1 shows the relationships between these roles and entities.

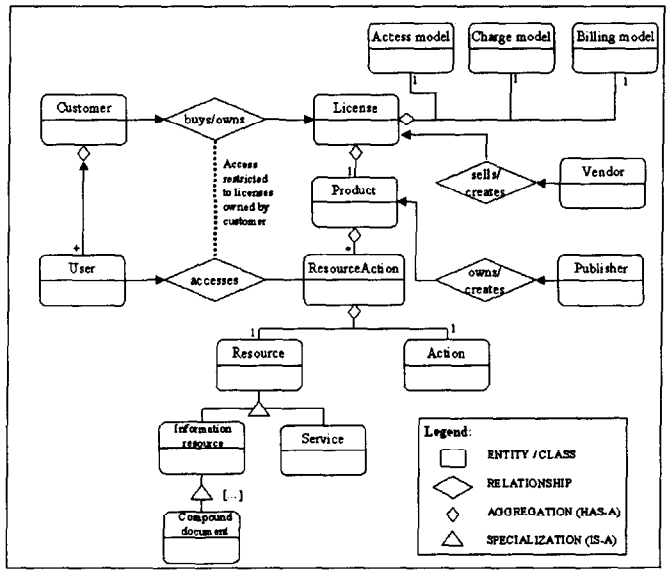

Figure 1: Entities and relationships in the electronic commerce model

Resources are the elementary goods traded in the information marketplace. Resources can be documents or services. Essentially, services are programs that dynamically create output, while documents are static. Documents can be compound, i.e., consist of multiple files. Resources are grouped and refined into products by publishers. Products constitute the smallest sellable entities. In particular, a product is defined by a set of (action, resource) pairs. An example for an action on a resource is "View" or "Download".

Vendors then define business models on products by defining licenses. A license embodies a specific business model, e.g., a one-year prepaid subscription to a journal. Customers can *purchase* licenses by paying a price to the vendor offering the license. Also, a price can be associated with *using* a license. Prices as well as license terms and conditions are subject to negotiation between vendors and customers. Customers are business entities that represent organizations or individual users. Organizations can be hierarchically structured. Individual users can be members of organizations.

In essence, a user u can access (use) a resource r if u is a member of a customer c who owns a license that provides access to at least one product containing r. In this case, we say that *user u uses license l* to access r.

In the remainder of this section, the concepts of our model and their relationships are discussed in more detail.

Resources Resources are elementary goods that are exchanged in the information marketplace. This includes hard goods (e.g., books, groceries, CDs) as well as soft goods. While there is no reason to exclude hard goods from our model, our focus is on soft resources.

There are two types of soft resources: *Information resources* such as documents, images, audio, or video, normally stored as files, and *services.* Services can be thought of as programs that generate information resources. For example, a (chargeable) service offered by a vendor can be a domain search, or a personalized homepage. A particular aspect of information resources is that they can be compound, i.e., a document can consist of multiple physical parts (e.g., an HTML file with images). Resources are associated with metadata. This metadata describes properties of the resource, e.g., creator, creation date, type, subject, or keywords.

Products Products are sets of resources, organized and bundled by a publisher. The notion of a product provides a valuable abstraction, as it allows producers to bundle resources from different repositories (and different intermediate producers). For instance, a product can be an issue of a scientific journal, consisting of a selection of articles.

A product in our model can be defined in three different ways:

- As an explicit list of resources (e.g., identified by URLs);
- As a list of products;
- Through a query expression based on metadata. This is called a *dynamic product definition.*

For instance,

$$P1 = \{http://www.foo.com/docs/doc1.html, P_2, \text{"}TITLE = AI\ Journal \text{ AND } YEAR = 1998\text{"}\}$$

defines a product consisting of the resource denoted by the above URL, an existing product P_2, and all titles of the AI Journal published in 1998.

As resources can be stored in different repositories, and be described using different metadata schemata, we will require mappings from different metadata schemata to the metadata schema underlying the query language used.

Dynamic product definitions are a source of considerable complexity; determining whether a given resource is in a product requires analyzing the product definitions at runtime, or maintaining a resource-to-product index structure which can be computed off-line.

Licenses A vendor's business model determines how it packages and sells products. In our model, a specific combination of a set of products under a business model is denoted by a *license*. I.e., a license provides its holder with access to a set of resources determines by a product definition, under the terms and conditions defined in the license. Vendors sell licenses to their customers; terms and conditions of a license can be subject to negotiation between vendor and customer. The terms and conditions of a license consist of three parts, each of which represent a different aspect of the business model incorporated by the license:

- *Access model*: defines the terms of access to the resources defined in the license;
- *Charge model*: defines how much customers are charged for purchasing and/or using the license;
- *Billing model*: defines how and when the customer is actually billed for the charges made.

Customers can purchase licenses from vendors, giving their users access to the resources covered by the license, under the terms and conditions specified in the license. The terms of a license are subject to negotiation between (human or automated) vendors and customers.

The flexibility and expressiveness of the electronic commerce model largely depends on the underlying license model, i.e., on which access, charge, and billing models it supports. Therefore, our approach is to provide an initial, sufficiently expressive model, and to specify APIs for the submodels, allowing application programmers to define customized access, charge, and billing models.

3.2 Business models and licenses

3.2.1 Access models

The access model part of a license allows one to specify restrictions on who may use the license. Our initial access model supports the restriction of access to a resource through a license by (combinations of) a number of criteria, such as the number of concurrent users of a license; network protocol information such as IP addresses, domain names, or LAN machine names; session information, such as the maximal duration of a session, periods of data access (e.g., *off-peak* hours only), or metered access parameters (restrict number of times a resource is accessed via a license, download data volume, number of available "cyber credits"); user authentication (login name / password, private certificates)

Access model API: a simple access model API provides two functions (listed below in Java notation), deciding whether a specific user has access to a specific resource:

```
boolean hasAccess(UserCredentials uc, Customer[] c,
          ResourceProperties rp)

License[] hasAccess(UserCredentials uc, Customer[] c,
          ResourceProperties rp)
```

uc is a property list containing user identification information; *c* is a list of customers of which the user accessing the resource is a member; *rp* is a property list describing the resource to be accessed by the user, including the specific action performed on the resource (see Figure 1). The first function just returns a boolean indicating whether the user is authorized to access the resource; the second function in addition returns a list of licenses through which a user can access the resource in question.

3.2.2 Charge models

The charge model encapsulates the core of an electronic commerce site, i.e., knowledge about and strategies for the pricing of products. Technically, the charge model specifies specific events that lead to charges being made to a customer account. There are two types of events relevant for charging[2], i.e. *license purchase* and *license usage*. A charge model specifies the following charging information:

On purchase of a license by a customer: Specify how much to charge (amount and currency), and when to charge (on purchase of the license (≡ NOW); at a specified date; on first usage of the license; Δ time units after purchase;)

On usage of a customer's license by a user: Specify how much to charge (amount and currency) and when to charge (on each usage, or each number of time units (daily, weekly, monthly, ...)); further (optional) restrictions, such as the date of first and last charging.

Charging API: There are two essential usages of a charge model covered in the initial charging API; one is to determine the conditions (price to pay, terms) for purchasing or using a license; this is encapsulated in the concept of an *offer*; the other is to actually accept or reject previously made offers (e.g., asking for the price of a document versus actually downloading the document). The charging API provides the following functions:

```
PurchaseOffer[] computePurchaseOffers(Customer c,
          ResourceProperties rp)

UsageOffer[] computeUsageOffers(User u, Customer c,
          ResourceProperties rp)

UsageOffer computeUsageOffer(User u, Customer c,
          ResourceProperties rp, License l)

boolean commit(User u, Customer cu, Offer o)

void discard(Offer o)
```

Essentially, an offer is a data structure that contains the relevant information about the offer, such as a unique identification, price, originator, the party to which the offer has been made, and the offer expiry date. Offers need to be forgery-proof, which can be achieved by using *digital signatures*. Authorized parties can extract the content of the offer and validate it. This data structure can be used to generate human-readable descriptions of offers and as a basis for automated negotiation of the offer terms. They also provide a receipt that can be used by the recipient of the offer to actually accept it (using the `commit` function specified in the API), or to discard it (using the *discard* function). In order to facilitate an automated auction, we would need to extend the API by functions to enabling e.g., making counter offers. An alternative route would be to switch from the traditional API approach to a speech-act-based model of negotiation as e.g., described by FIPA [Ste98]. While currently we see no necessity to do so, we may pursue this approach in the future.

3.2.3 Billing model

While the charge model determines what and when to charge a customer account, the billing model determines how and

[2]The same distinction can actually be made for access control and billing; however, we do not pursue this issue in this paper.

when payments are actually made. E.g., in the case of a phone bill, charges are made for each phone call; however, the customer is billed only once a month. The billing model allows one to specify the method of billing for transactions carried out under a license (e.g., per invoice, direct debit, or credit card), as well as the time and frequency of billing (on each charge, on charge level, monthly, yearly, ...); it also records billing details and customer-specific discounts. For online transactions, the billing model is used to decide e.g., whether to offer a customer the possibility of online credit-card payment, or payment using SET or XML/EDI protocols. It also plays an important role for integration with traditional fulfillment and invoicing systems. So far, we have not developed a dedicated billing API.

3.3 Examples

In the following, we show how combinations of the three components of a license can be used to implement different business models.

Annual subscription: A license encapsulating the basic subscription model described as Scenario 1 in Section 2.1 can be defined as follows:

- *Product:* all 1998 issues of a journal;
- *Access model:* five concurrent users + restriction to a specific IP address mask (e.g., `194.72.55.*`);
- *Charge model:*
  - *On purchase:* 495 US$, charged to customer account at the end of a one month free trial period;
  - *On usage:* no charge;
- Billing model: per invoice, including 5% discount for the customer.

Virtual journal with off-peak–only access: A model that implements the *virtual journals* business model outlined in Scenario 3 of Section 2.1 can be defined as follows:

- *Product:* list of ten articles;
- *Access model:* a single concurrent user, allow maximum of 10 accesses per month, authentication by user name / password combination, restrict to off-peak hours EST;
- *Charge model:*
  - *On purchase:* 50 US$, charged on purchase;
  - *On usage:* 5 US$ monthly, first charge three months after purchase;
- *Billing model:* via direct debit;

Pay per access: This example shows how our model can be used to describe a pay-per-view business model in which charges are made according to user requests. In the example, this is done by defining a one-off license.

- *Product:* all articles in a journal;
- *Access model:* allow a single access to authenticated user;
- *Charge model:*
  - *On purchase:* 20 US$, charged on purchase;
  - *On usage:* no charge;
- *Billing model:* on-line payment using SET.

3.4 Operational model

So far, we have described the structure of the electronic commerce model. Now, we illustrate its operational aspects. The most important operation carried out is to decide whether and under what conditions a user who requests access to a resource (e.g., by clicking on a hyperlink on the shopfront web page) is allowed to access the resource. This operation involves the following steps:

1. Intercept a resource request made by a user (e.g., an HTTP request);
2. Identify the resource that the user wishes to request and the credentials of the user;
3. Authenticate the user; if authentication fails, EXIT;
4. Check whether the user has access to the resource via a *valid license* using the *Access Model API*; if this is not the case, compute a return code, EXIT;
5. By using the Charging API, compute the cost of access;
6. If the business logic requires, ask the user for confirmation. If confirmation is denied, EXIT;
7. Grant the request and perform the necessary transaction, providing the user with access to the resource, and making the charges as specified in the license (using the Charging API).

The first three steps involve the usual web-based authentication and access control mechanisms such as login name and password, or a private certificate. The crucial steps in the above process are steps 4 and 5, which involve testing whether a user has access to a resource and at what cost. For a given user u and a resource r, resource access is determined by the following steps:

1. Compute the set C of customers of which u is a member;
2. Compute the set L_1 of licenses u can access, i.e., the union of all licenses held by customers $u \in C$;
3. Compute the set $L_2 \subseteq L_1$ of valid licenses for user u; doing this involves dynamically checking various conditions specified in the access model of the license, such as the number of users currently using a license against the maximum number of concurrent users;
4. Compute the set $L \subseteq L_2$ of licenses that provide access to a product that contains the resource r;

This sequence of steps returns a set of licenses L, which grant the user u access to the resource. If $L = \emptyset$, the user has no access, and an appropriate action is taken (e.g., guide user to registration or catalog page).

At this point, our model provides sufficient information to decide *whether or not* a user is authorized to access a resource. The second step is to find out the cost of access to a resource. Using the functions of the charge model API, the system can compute a set of offers providing different possibilities (associated with different terms) to access a resource as described in Section 3.2. This information can be presented to the user by other applications, or can be used for automated decision-making (see Section 5).

4 Wiley InterScience

WILEY INTERSCIENCE (*http://www.interscience.wiley.com*) is a web-based online information service. WILEY INTERSCIENCE is used by the scientific publisher John Wiley & Sons to provide their customers with access to approx. 400 scientific journals and, in the future, other scholarly reference works and personalized information services. WILEY INTERSCIENCE has been designed based on a classical three-tier architecture ((i) user interfaces and client applications, (ii) business logic, (iii) data storage, see Figure 3) and on standards such as CORBA and JDBC. In Section 5 we will explain how this three-tier architecture has been extended to support an intelligent agents perspective.

The system has been online since mid 1997; initially, it provided registered users with free access to approx. 160 journals, including basic access control, navigation, search, and personalization features. Figure 2 shows the personal homepage visible to a user of WILEY INTERSCIENCE after logging into the system.

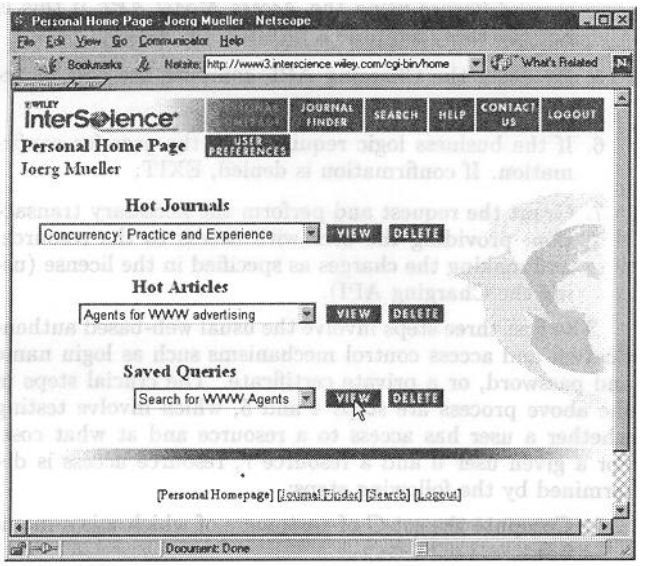

Figure 2: WILEY INTERSCIENCE

Since its commercial launch in January 1999, the system provides improved search and navigation features as well as the restriction of access to print subscribers. In this phase, the performance requirement on WILEY INTERSCIENCE amounts to manage approx. 400 journals with five millions of files, and to support up to 500 concurrent users.

Subsequent development phases will add more sophisticated business models; in these phases, we will focus on the development of intelligent agent services such as increased personalization, and the possibility to purchase licenses online. Currently planned are pilot projects exploring multi-publisher coordination (in particular, cross-linking and virtual journals) and agent support for automated setting up of draft price agreements between vendors and customers (see Section 5 for a more detailed description).

5 An Intelligent Agents Perspective

As far as WILEY INTERSCIENCE is concerned, our perspective of agents is that of flexible, intelligent domain services (in our application: library services) that are able to communicate with each other and with humans while in operation. In this context, there are two relevant views:

- WILEY INTERSCIENCE can be regarded as a vendor node in the information marketplace, providing services to other consumer, vendor, or facilitator nodes.
- The internal structure of WILEY INTERSCIENCE is complex enough, and its operation offers sufficient challenges to make the perspective of the system itself as an agent system a useful option.

Both perspectives complement each other, as both look at the system in terms of providing services; the difference is that the former perspective deals with the macro level (i.e., looking at the services WILEY INTERSCIENCE offers to the external world), while the latter operates at the micro-level (describing internal services).

Figure 3 shows the underlying software architecture. As mentioned above, the system is organized as a classical three-tier architecture. The middle layer is the core of the system; it basically implements the electronic commerce model as defined in Section 3, encapsulated by an access control and session management system, ensuring an appropriate level of security and run-time efficiency.

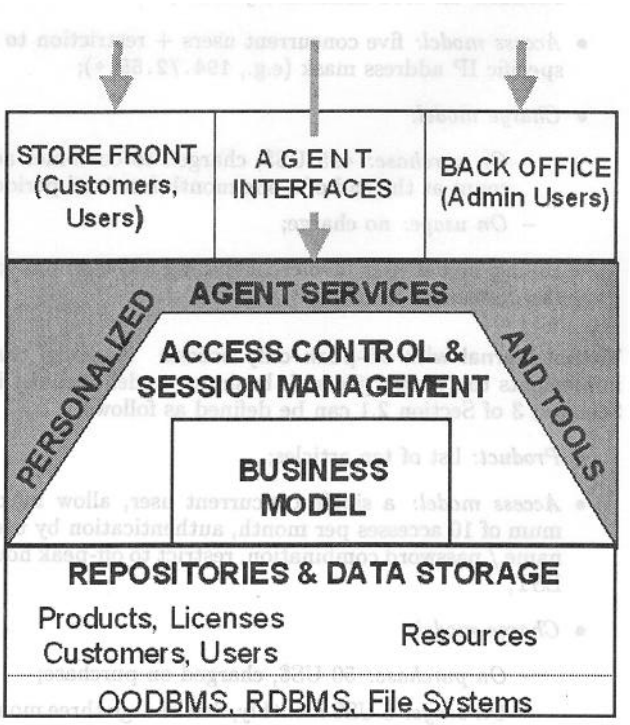

Figure 3: Electronic vendor architecture

The two perspectives mentioned above are dealt with in this architecture by an intermediate layer of *personalized agent services and tools*. This layer can be regarded either as an upward-extension of the business logic or as a downward extension of the user interface layer. Agent services can be accessed from the user interface and application layer and thus be accessible for store-front and back-office applications. In addition, we envisage an agent interface that allows software agents from other sites within the information marketplace to communicate with our agent services.

In building a large-scale commercial system, our approach to developing agent technology is clearly based on Etzioni's *useful-first* paradigm, starting with the essentials and adding the desirable but not so essential features later. Therefore, the agent interface has not yet been implemented, and so far we have only designed basic agent services that are essential for the operation of the vendor node. In the following, we discuss some of the problems we encountered during the development of the system, and list a number of useful (existing or future) services).

Table 1 lists a number of useful agent services, classified into (external, user-oriented) macro services and (internal, vendor-oriented) micro services. For many of the individual services, techniques and prototypes to implement them have been described in the literature. However, what interests us is their integration into a real-world system.

External services	Internal services
Shopfront assistant	Sales decision support
License purchase advisor	License generation
Adaptive user profiler	License validation/verification
Personalized content presenter	Product management assistant
Seamless authentication	Production workflow mgt.
Seamless, content-based linking	Access control support
Proactive notification	Alerting services
Personalized, multi-domain search	Customer services support
Comparison shopping	Handling access denial
Automated negotiation	Process tracking
Service brokering; recommending	Automated offer generation
Workgroup filtering	

Table 1: Personalized agent services and tools

In the following, we discuss three of these areas which require urgent solutions from our perspective.

Personalization The WWW will change traditional print publishing business models. Even mapping the subscription model into an electronic counterpart is not straightforward, as the meaning of notions such as *concurrent usage* of a resource in electronic publishing is very different from its print publishing counterpart. Even more dramatic, the publishing market is higy likely to change from a producer market to a consumer market. Publishers will no longer be able to dictate business models, and to use one business model for all customers. All of a sudden, it is possible and necessary to tailor business models to individuals. This will lead to developments such as virtual journals, personalized journals, and information channels with notification and alerting.

In this context, adaptive profiling and matching technology developed in the agents community is essential, using syntactic (TF/IDF document analysis, Vector Space model) and semantic (ontology-based) means of analyzing documents, and using feedback learning to extract and maintain user profiles. Also, as we will detail in the following paragraph, the personalized generation of business models for millions of customers cannot be achieved without automating at least some of its aspects.

Decision support If a user of WILEY INTERSCIENCE tries to access a resource stored in the system, finding out the best way of doing so is non-trivial, making decision-support for users an obvious agent-enabled shop-front service. The problem is that a user can have more than one way to access (and pay for) a resource. For instance, a researcher could have access to an article:

- through a prepaid subscription from its university;
- as part of a personal license that gives access to ten articles for a special price;
- via a pay-per-access option.

Although the best solution (choose the prepaid subscription license) seems obvious in this example, the general decision problem of selecting the *best* from a set of applicable license to access a resource is interesting, as the utility function may depend on multiple attributes. We have developed a simple license selector agent that implements a simple strategy based on filtering rules, which seems sufficient for most cases:

Rule 1: Prefer subscriptions to pay-per-view or metered access licenses;

Rule 2: For pay-per-view or metered access licenses: if the specified charges are comparable (i.e., they are the same units, or there is a known translation), select the ones with the minimal charges;

Rule 3: For all licenses that are applicable after applying Rule 1 and Rule 2, the system should be configurable to either choose an arbitrary license, or to present the user with a choice, deferring the selection to her.

A similar type of agent can be used to solve a related problem: To advise a user who does not have the required license to access a resource for the best deals to purchase a license. Also, such a *sales advisor agent* can explore the user's reading properties for patterns and e.g., suggest the user a change from a pay-per-view option to a subscription, which may be advantageous both for user and vendor.

Seamless authentication, charging, and linking Looking at the macro-level, the usability of a system such as WILEY INTERSCIENCE within the information marketplace can be improved by increasing the level of transparency of services across vendors. For instance, users should be asked for their identification and charging information only once. Also, a user should not have to deal with various charging and billing systems. Aspects of this can be supported by a software agent who knows about different authentication schemes, and about the users consumption behavior. Such an agent can interact with different vendor nodes to authenticate the user transparently, and to generate one charge based on multiple commerce transactions at different sites in the information marketplace. The development of such an agent will have to take into account emerging electronic commerce and communication standards (SET, wallets, XML/EDI) that are currently being developed to deal with these problems) as well as privacy issues.

A related problem is that of publisher-transparent linking, i.e., linking from documents managed by different vendor nodes. One route of dealing with this problem is to hope for the emergence of resource naming standards (e.g., Digital Object Identifiers). A second, complementary route is content-based linking: literature reference are treated as queries that in most cases uniquely identify a resource, and to model traversing these links as a distributed search process, which implies posting the query to different vendor nodes. This process can be supported by distributed agents. As a first step in this direction, we are currently building up a reference gateway service that enables dynamic resolution of the location of the resource identified by a given reference.

6 Conclusions and Future Work

In this paper, we have described a novel model for electronic commerce, developed to serve the needs of a vendor of information and library services. We have embedded this electronic commerce in a service that is based on the paradigm of an agent-based information marketplace. A simplified version of the model has been implemented in the web-based online electronic commerce system WILEY INTERSCIENCE that serves the needs of the world's second-largest scientific publisher and millions of "information consumers". Finally, we have outlined requirements and opportunities for employing agent technology in this domain.

The model satisfies most of the requirements stated in Section 2. The requirements that are not yet fully satisfied are *decision support* and *automation*. However, we believe that the system so far provides a basis that will allow us to collect usage and demographic information upon which we can build *decision support* applications. Also, the general underlying electronic commerce model serves as a basis for *automation* of business transactions and negotiation between software agents. Externalizing the model by mapping it into an appropriate data interchange standard (e.g., XML) will be one of the immediate next steps.

An important lesson learnt from this project has been that agent technology is not the first thing to worry about. Far more important are effective requirements gathering and development processes, and, in the end, the human factor. To put this in terms of Winston's "raisin bread" model of AI: it appears that the raisins are often the first thing to go over board (or the last thing to be added) in the case of a real-world software development project.

An issue that we have hardly touched in this paper is *security*. In fact, the development of our system is based on standard security technologies (firewall architectures, web server URL auditing and access control, encryption), to deal with the lower-level aspects of security (see [Sem96] for an overview). The access model, for instance, should not be used to reinvent standard web server security, but rather be used on top of it, to provide more fine-grained and expressive control over library services and other information resources, where such additional control is required.

Maybe the two most thrilling questions for the future of WILEY INTERSCIENCE are which business models will finally prevail in Internet publishing, and how we can manage the bottom-up development of a multi-publisher *Digital Library*, starting from a vendor-oriented micro-model of electronic commerce.

Acknowledgement

We thank the Wiley InterScience development team for ongoing support and discussions: Rosie Altoft, Stuart Bremner, David Derbyshire, Aurora Diaz, Yinka Fagbenro, Aidan Killian, Michael Kolb, Bevis Lloyd-West, Gregory St.John, and Andy Townsend.

References

[AD98] A. A. Armstrong and E. H. Durfee. Mixing and memory: Emergent cooperation in an information marketplace. In *Proc. of the 3rd Intl. Conf. on Multi-Agents Systems*, pages 34–41. IEEE Computer Society, 1998.

[Bir95] W. P. Birmingham. An agent-based architecture for digital libraries. *D-Lib*, July 1995.

[CM96] A. Chavez and P. Maes. KASBAH: An agent marketplace for buying and selling goods. In *Proc. of the First Intl. Conf. on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, London, UK, 1996.

[DEW97] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison shopping agent for the world-wide web. In *Proc. of the First Intl. Conf. on Autonomous Agents*, pages 39–48. ACM Press, 1997.

[FMPW97] I. A. Ferguson, J. P. Müller, M. Pischel, and M. J. Wooldridge. Web economics: A case for agent-based digital libraries. In *Proc. of the IJCAI'97 Workshop on the Future of AI and the Internet*, Nagoya, Japan, 1997.

[Fon97] L. N. Foner. Yenta: A multi-agent, referral-based matchmaking system. In *Proc. of the First Intl. Conf. on Autonomous Agents*, pages 228–236. ACM Press, 1997.

[GMM98] R. Guttman, A. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review*, June 1998.

[KH95] D. Kuokka and L. Harada. Matchmaking for information agents. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 672–678. Morgan-Kaufmann Publishers, 1995.

[LMPS95] C. Lynch, A. Michelson, C. Preston, and C. A. Summerhill. CNI white paper on networked information discovery and retrieval. http://www.cni.org/projects/nidr/www/toc.html, 1995.

[MZ97] A. Moukas and G. Zacharias. Evolving a multiagent information filtering solution in Amalthaea. In *Proc. of the First Intl. Conf. on Autonomous Agents*, pages 394–403. ACM Press, 1997.

[NSF98] WWW page of the NSF Digital Library Initiative. The National Science Foundation, http://www.cise.nsf.gov/iis/dli_home.html, 1998.

[Sem96] C. Semeria. Internet firewalls and security: a technology overview. 3Com Corporation, http://www.3com.com/nsc/500619.html, 1996.

[Ste98] D. Steiner. Industrial applications of multi-agent technology. In *Proc. of the 3rd Intl. Conf. on Multi-Agent Systems (ICMAS'98)*, pages 12–13. IEEE Press, 1998.

[Wel95] M. P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 27(3), 1995.

[WWW98] P. R. Wurman, W. E. Walsh, and M. P. Wellman. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Proc. of the 2nd Intl. Conf. on Autonomous Agents (Agents'98)*, Minneapolis, US, 1998. AAAI Press.