Commercial electronic publishing over open networks: a global approach based on mobile objects (agents)

Morin, Jean-Henry

UNIVERSITÉ DE GENÈVE

Faculté des Sciences Économiques et Sociales
Département de Systèmes d'Information

Commercial Electronic Publishing over Open Networks: A Global Approach Based on Mobile Objects (Agents)

THÈSE

présentée à la Faculté des sciences économiques et sociales
de l'Université de Genève
par

Jean-Henry MORIN

originaire de Genève-ville (GE)

pour l'obtention du grade de
Docteur ès sciences économiques et sociales,
mention systèmes d'information

Membres du jury de thèse:

M. Michel LÉONARD, Professeur, président du jury
Mme Solange GHERNAOUTI-HELIE, Professeur, HEC - Lausanne
M. Thierry PUN, Professeur, Faculté des Sciences - Genève
M. Dimitri KONSTANTAS, Maître d'enseignement et de recherche
M. Dennis TSICHRITZIS, Professeur, directeur de thèse

Thèse no 475
Genève, 1999

La Faculté des sciences économiques et sociales, sur préavis du jury, a autorisé l'impression de la présente thèse, sans entendre, par là, émettre aucune opinion sur les propositions qui s'y trouvent énoncées et qui n'engagent que la responsabilité de leur auteur.

Genève, le 28 janvier 1999

Le doyen

Beat BÜRGENMEIER

Impression d'après le manuscrit de l'auteur.

Remerciements

J’aimerais profiter de cette occasion pour remercier et témoigner ma reconnaissance à tous ceux sans lesquels ce travail n’aurait jamais existé.

Tout d’abord, j’aimerais remercier le Professeur Dennis Tsichritzis de m’avoir accepté dans son groupe de recherche, le Groupe Systèmes Objet, et de m’avoir ainsi donné l’occasion de faire un doctorat dans un environnement particulièrement stimulant qui m’a accompagné tout au long de mon travail. A son contact, d’abord comme étudiant puis comme assistant, j’ai énormément appris. Je le remercie d’avoir accepté d’être le directeur de cette thèse.

J’aimerais remercier tout particulièrement le Dr Dimitri Konstantas pour son soutien, sa confiance, ses conseils avisés et ses encouragements tout au long de mon travail. Je lui dois beaucoup dans l’accomplissement de cette thèse. Tout d’abord lors des intuitions initiales qu’il a vivement encouragées, puis au fil du temps par son enthousiasme et le travail commun que nous avons fait, finalement lors de la transformation de ce travail en une thèse de doctorat dont il a supervisé toutes les étapes. Pour tout cela, ainsi que pour avoir été membre du jury, je lui adresse mes plus vifs remerciements.

J’aimerais remercier le Professeur Solange Ghernaouti-Hellie d’avoir accepté d’être le juré externe de cette thèse. Ma gratitude va aussi au Professeur Thierry Pun qui a accepté de faire partie du jury de thèse et au Professeur Michel Léonard qui a accepte le rôle de président du jury. Par ces lignes je tiens a leur exprimer ma reconnaissance pour leur enthousiasme et leur lecture attentive.

J’adresse aussi mes remerciements à tous les membres, passés et présents, du Groupe Système Objet et du Centre Universitaire d’Informatique que j’ai eu la chance de rencontrer et avec qui j’ai eu le plaisir de collaborer et discuter tout au long de mon travail. En particulier j’aimerais remercier le Dr Vassilios Prevelakis pour les précieuses discussions que nous avons eues au début du projet *HyperNews* qui ont abouti avec le Dr Dimitri Konstantas au modèle de distribution de documents. Je remercie le Dr Constantin Arapis à qui je dois de précieux commentaires lors de la rédaction de certaines versions initiales de mon travail ainsi que de nombreuses discussions enrichissantes, le Dr Jan Vitek qui m’a fait partager son enthousiasme pour les agents et le Dr Ciarán Bryce pour la lecture de ce travail et ses précieux commentaires.

Je remercie le Fonds National Suisse qui a financé le projet *HyperNews* (no 5003-045333) dans le cadre du programme prioritaire de recherche, *Information and Communication Structures* (SPP-ICS, 1996-1999).

J'aimerais remercier Bruno Giusanni et José Rossi qui ont été les contacts successifs avec *L'Hebdo* partenaire du projet *HyperNews* dont un point culminant a été la démonstration donnée dans le cadre de l'exposition Computer 98 à Lausanne.

J'aimerais aussi remercier ceux qui m'ont initié à l'informatique car ils ont transformé une passion en une profession rigoureuse qui n'a cessé de me servir tout au long de ma vie. Je garde un souvenir vivace de cette époque de la *Formation de Programmeur Analyste* (FPA) dont la qualité de l'enseignement et le dynamisme en a fait successivement une *Ecole Supérieure d'Informatique de Gestion* (ESIG) puis une *Haute Ecole Spécialisée* (HES) dont la réputation n'est plus a faire.

Enfin je remercie tous ceux qui ont participé de près ou de loin à cette aventure, qui m'ont encouragé à un moment ou un autre, qu'ils m'aient écouté poliment ou participé activement à des discussions ou encore partagé de nombreux cafés; à tout ceux-là qui ne sont pas cités ici mais qui se reconnaîtront certainement, j'adresse mes plus vifs remerciements.

Je tiens à remercier mes parents pour tout l'amour et toute la confiance qu'ils m'ont témoignés ainsi que le soutien et les encouragements qu'ils m'ont apportés pendant mes études et à tous les moments clés de ma vie dont cette thèse n'est pas le moindre. Pour tout cela je les remercie du fond du coeur et je leur dédie cette thèse.

Finalement, je tiens à remercier tout particulièrement ma femme Catherine, qui m'a accompagné et encouragé dans cette aventure avec une confiance aveugle. Elle a partagé quotidiennement et de près toutes les étapes de ce travail, depuis les idées initiales jusqu'à leur concrétisation, en passant par les moments de doute ainsi que les joies que ce travail a engendrés. Pour sa patience, son amour, sa compréhension et son soutien permanent et sans limite c'est tout naturellement que je lui dédie cette thèse.

Acknowledgments

I would like to take this opportunity to thank all the people without whom this work would never have existed.

First I would like to thank Professor Dennis Tsichritzis for having accepted me in his research group, the Object Systems Group, and thus given me the opportunity of doing a Ph.D. in a most stimulating environment that I have enjoyed throughout my work. At his contact, initially as a student then as a research assistant, I have learned very much. I thank him for having accepted to be the director of this thesis.

A special thanks goes to Dr. Dimitri Konstantas for his support, trust, wise advices and encouragements throughout my work. I owe him much in the fulfillment of this thesis. First as he encouraged my initial ideas, then with the passing of the years by his enthusiasm and the common work we did, and finally during the transformation of my work into a Ph.D. thesis which he supervised at all stages. For all this and for being part of the thesis committee I am very grateful.

I would like to thank Professor Solange Ghernaouti-Hellie for accepting to be the external examiner of my thesis. My gratitude also goes to Professor Thierry Pun for accepting to be part of the thesis committee and to Professor Michel Léonard for accepting the role of president of the jury. I hereby express my gratitude for their enthusiasm and their thorough reading.

I would also like to thank all the members, past and present, of the Object Systems Group and the Centre Universitaire d'Informatique that I have had the chance to meet and with whom I enjoyed collaborating and discussing throughout my work. In particular I would like to thank Dr. Vassilios Prevelakis for the valuable discussions we had at the beginning of the HyperNews project that led with Dr. Dimitri Konstantas to the document distribution model. I am also grateful to Dr. Constantin Arapis who contributed valuable comments on early versions of my work as well as numerous enriching discussions, Dr. Jan Vitek who shared with me his enthusiasm on agents and Dr. Ciarán Bryce for reading this work and providing valuable comments.

I would like to thank the Swiss National Science Foundation who financed the *HyperNews* project (no 5003-045333) in the scope of the Swiss Priority Program, *Information and Communication Structures* (SPP-ICS, 1996-1999).

I also want to thank Bruno Giusanni and José Rossi who have been the successive contacts with *L'Hebdo*, partner of the *HyperNews* project, where a high point was the demonstration given during the *Computer 98* exhibition in Lausanne.

I would also like to thank those who initiated me to computer science for they have turned a passion into a rigorous profession that has served me ever since. I keep a lively memory of this time of the *Formation de Programmeur Analyste* (FPA) whose quality of teaching and dynamism made it in turn an *Ecole Supérieure d'Informatique de Gestion* (ESIG) then an *Haute Ecole Spécialisée* (HES) whose reputation is well established.

I want to thank all those who have taken part closely or by far in this adventure, those who have encouraged me at one time or another, those who have kindly listened to me or actively participated in discussions or even shared many cups of coffee. To all of those who are not cited here but will surely recognize themselves I address my warmest thanks.

I would like to thank my parents for all the love and trust they have shown me as well as for their support and encouragements throughout my studies and at all the key moments of my life of which this thesis is not the least. For all they have done I thank them from the bottom of my hart and dedicate them this thesis.

Finally, my very special thanks go to my wife Catherine, who accompanied and encouraged me throughout this journey with faith. She shared closely and day by day all the steps of this work from the initial ideas to their materialization, going through moments of doubt as well the rewards this work has created. Therefore, I naturally dedicate this thesis to my wife Catherine for her patience, her love, her comprehension and for her unlimited and constant support.

to my wife

and my parents

Contents

Chapter One | *Introduction.*

The field of information technologies is currently witnessing an unprecedented phenomenon of globalisation and acceleration. This is mainly due to the success of the World-Wide Web which marked a capital turning point in the history of the Internet, by redefining the very role of the computer in the context of a global network. This growth of information technologies is all the more significant since the commercial potential which accompanies it is considerable. From this point of view, our economic system grew rich by a new dimension by the means of electronic commerce. Consequently, a commercial transaction, composed of a payment as the counterpart of goods and / or services, can take an electronic form in either or both its two components. This raises however significant questions which do not concern solely data processing and computer science but also disciplines like law, economics, social sciences, ethics etc.

In this perspective of an “all-numerical interconnected world”, we chose to focus on the field of Electronic Publishing. Commercial electronic publishing over open networks such as the Internet, represents both a great opportunity and a significant challenge for editors, publishers and content providers in a competitive global information market. The information industry is on the way to become a major foreground industry at the dawn of the third millennium. It potentially represents a market which amounts to billions of dollars. However, to date this industry still opposes justifiable reluctance with respect to this type of distribution in spite of the generalization of the means of computerized edition and production and the emergence of global information networks. Hence, the publishing and content industry is still largely dominated today by the production of “physical” documents (in opposition to electronic documents) dispatched through traditional distribution channels. This also applies to CD-ROMs which are considered the same way as books. These reserves are primarily related to the inherent problems in this transition towards an electronic world. In particular with respect to copyright protection and enforcement, intellectual property rights, remuneration of authors rights, electronic payments, security, confidentiality, privacy, information unit granularity, electronic document properties and business models, etc.

Electronic publishing has been approached in the scope of many projects and studies each with particular emphasis on specific issues such as digital libraries, scholarly publishing, typesetting, internationalization, definition languages and grammars, copyright management systems, collaborative work, distributed systems, etc. However, most fail in addressing major issues among which we can identify: copyright protection, privacy, anonymity, flexibility of distribu-

tion models, freedom of choice with respect to commercial partners and electronic payment schemes, dynamics of author rights properties linked to documents, content type and formats or applications for which these documents are intended, etc. The major drawback often being that in the end, once the document has reached its destination, it falls out of control and can thus be reproduced and distributed in an unlimited way at extremely low cost due to its electronic nature.

The general problem that we address aims at studying commercial electronic publishing over open networks such as the Internet in terms similar to traditional publishing, i.e., by using existing technical and legal means in order to promote electronic publishing in a global electronic market while protecting its actors. In other words, the transition towards an "electronic world" of the publishing industry should not be guided solely by technology but also by the inherent needs of an electronic metaphor of traditional publishing. Consequently, the approach must be as open as possible, in order to offer maximum flexibility to content providers as well as to the information consumers and thus set the foundation to enable new heuristics.

It is necessary to consider the Electronic Publishing term in its broadest possible meaning. For example, it can be as well a matter of newspaper or magazine articles as books, and even official or legal documents such as income tax or digital libraries. All things considered, any value added information service, to the full extent, is a potential candidate for the field of electronic publishing. Notwithstanding the diversity of the touched fields, it seems reasonable to postulate that in most cases, it is necessary to consider common needs. Among those, let us raise copyright protection, compensation of authors rights and royalties at any level of added value, security, integrity and authoritativeness of both the content and the involved parties, increased flexibility with respect to commercial and personal policy choices of the information providers as well as of the information consumers.

It is our opinion that electronic publishing is to be understood at a very abstract level as a broad concept that includes many dimensions ranging from legal to technical up to social concerns. Therefore, focusing on the infrastructure enabling the trade seems to be a promising direction thus offering through a framework a common abstraction to classes of electronic publishing applications sharing similar minimal requirements.

In the scope of this thesis, it is our goal to propose a global approach to commercial electronic publishing by the means of a minimal infrastructure through a framework, based on the mobile object (agent) paradigm. Such an approach should make it possible to take into account the needs that are common to the field of electronic publishing in a commercial global networked market which is competitive and untrusted by essence. The use of mobile objects or agents in this context represents a new and original direction whose major advantage is to give a total autonomy to the electronic document while guaranteeing and enforcing a "self-controlled" safe and secure usage of its content at all times. Such an infrastructure should consequently support and encourage the commercial exchange and dissemination (i.e., superdistribution) of electronic documents in the broad sense on global information networks such as the Internet.

1.1 Motivation: The Publishing Industry Paradox or Discrepancy

Publishing is an honorable well established industry which has gone through many (r)evolutions over the centuries from the ancient times of papyrus to Gutenberg and up to current computer assisted publishing. Nowadays, the publishing industry could simply not do without computers just as our whole society.

There is however an interesting paradox or discrepancy that can be witnessed in today's publishing industry. Although the upstream publishing steps of *composing* and *editing* are now fully electronic and computer assisted, the downstream steps of *production* and *distribution* are still dominated by the production of *hard copy* printed documents following traditional distribution channels to reach the consumers. Furthermore, this is all the more astonishing since global networks exist and information is already in digital form thanks to the processing that occurs upstream.

It seems clear now that we have gone beyond the simple initial question of whether electronic publishing represents a threat to the publishing industry or not. Now, the publishing industry commonly agrees that electronic publishing represents a true opportunity provided a number of problems be resolved.

These problems represent the major obstacles hampering the progress and full endorsement of commercial electronic publishing over open networks such as the Internet.

- Weak or non-existent copyright and intellectual property rights protection
- No standard, wide spread, secure and truly open ended electronic revenue collection infrastructure

These issues among the most prominent, are the major reasons why electronic publishing has not fulfilled yet this, rather simple in appearance, end-to-end process. Consequently, commercial electronic publishing is currently often achieved through CD-ROMs and is thus bound to existing intellectual property and copyright laws but without taking any advantage of an existing global networked market. As a result, electronic publishing on the internet is often considered to be untrusted, insecure, non commercial, unrefereed and of low value.

The inherent nature of information, in its broadest sense, make it a prime candidate for a total one hundred percent networked transaction (i.e., including all the steps of a commercial transaction from order to payment up to delivery).

1.2 The Universal Library: The Vision

Coming back to this broad notion, that we have claimed electronic publishing was, earlier in this introduction, it is of prime importance to mention here the vision of *The Universal Library* as described by a distinguished researcher Dr. Raj Reddy, in the mid-nineties. In a paper [1] presented at the Forum on Research & Technology Advances in Digital Libraries, held in May 1995 (ADL'95), the author presents his vision of what he has named *The Universal Library* and discusses a number of issues and challenges it raises. The author concludes by commenting on the possible driving forces to attain such a goal. Namely, through international collaboration and the potential billion dollar revenue that such an industry could generate.

Both driving forces have proven to be valid. First with the emergence of the World-Wide Web, the Internet gained its status of Global Information Infrastructure (GII) and generated the required political interest which turned out to be in some way a major issue of international collaboration. Second, also through the Web, the Internet gathered almost immediately the industry around it, thus revealing the potential profit that such a technology was about to enable.

Nevertheless, it is amazing to see that what was considered a vision a few years ago is about to become, at least in parts, our daily bread. But without further notice, let us go through this most refreshing description (excerpt) of an after all not so unrealistic scenario that we would all be ready to live today:

> *A Future Scenario*
>
> [...] You are in bed after a hard day's night. You pick up your personal ThinkBook running Win-Warp-99. It weighs 24 ounces, roughly the weight of a hardback book. It has a 6x8 inch screen with color and gray scale with a 300 dot per inch resolution. It communicates with your home computer using a home wireless local area network. This in turn is connected to the The Universal Library. The Thinkbook sells for around $99, roughly the cost of manufacturing the device, following the analogy that you give away the razor to sell the blades.
>
> You select a romance novel from your Sony Disk-of-the-Month Club for a single reading privilege. A secure co-process in your ThinkBook queries your home information system and tells you that the cost for read-once privilege is 25 cents. You say "okay" (remember you are in bed - you don't want to type on a keyboard or write with a pen). The secure co-processor then decodes the encrypted novel using a Public-Private key decryption algorithm.
>
> You start to read it, but you get bored and ask for a refund. You get 17 cents back. The Digital-Back Book version of the romance novel self-destructs.
>
> You are trying to decide what might be fun to read. You call up the New York Times book review. The abstracts of the book reviews are free, but you get the ads along with the best seller list. A full review of a book costs you 5 cents to read.
>
> You notice that Tom Clancy's new book has been on the best seller list. You check availability in electronic form. It is not yet available on any disk-of the-month clubs. You ask for availability of an electronic version through the Hard-Back Electronic Book Channel from Random House. Indeed, it is available for a charge of $9.99. You don't want to spend that much and decide to wait until a Paperback Electronic Book Channel version from Bantam with multi-use privilege will become available for only $1.99.
>
> You go back to the New York Times book review and look at the old best-selling paperback list. You find Michael Crichton's Disclosure, which is available for 99 cents. You download it from a local server which has been caching all the frequently accessed books from the Paperback Electronic Book Channel. You get a special discount of 25 cents when you purchase a cached book on a local server rather than accessing it from an international database.

> You start to read this book and decide you would like to hear the background music from sound track of the movie Disclosure. You access the Digital Music Library Channel from Time Warner and pay 35 cents for a single use of the Disclosure sound track. Your ThinkBook transmits the music to the Bose Acoustimass speakers on either side of your bed. You read on to the background music.
>
> You then come to a section where there is a description of Virtual Navigation within a data base, and are curious how it is portrayed in the movie. You access the Digital Video-on-Demand Channel from TCI and request the movie Disclosure for a view-once privilege for $1.99. You set your ThinkBook aside and fall asleep while watching the movie.
>
> Now for a few business details. That night, about 0.lpopulation used The Universal Library. The total value of the service's provided was $17 million for that day. The Universal Library kept $5 million, paid $3 million to the phone companies, and the remaining $9 million went to a producer's clearing house, which in turn distributed these proceeds to authors, editors, producers and agents in agreed upon proportions. IBM, Apple and Sony sold 50 million intelligent information appliances that year at prices ranging from $99 to $9,999 at a total market value of $30 billion. That year revenues of The Universal Library were about $10 billion and expected to grow to a trillion dollars by the middle of the century.
>
> The above scenario provides an illustration of a future world of Thinkbook, Win-Warp-99 and The Universal Library. Such a vision raises a number economic, technical and legal issues which need to be solved before the vision can be realized. The rest of the paper discusses a number of these issues. [...]
>
> Raj Reddy
> The Universal Library: Intelligent Agents
> and Information on Demand [1]

1.3 Contribution

This thesis addresses the issue of commercial electronic publishing over open networks such as the Internet. Our claim is that many electronic publishing applications share common needs from an infrastructure point of view enabling the safe and secure trade of content. To support this claim, our objective is to describe a framework offering such an abstraction to classes of electronic publishing applications sharing common needs in terms of security, copyright, intellectual property, authoritativeness, untampering, etc. The mobile agent paradigm will be used in two ways. First as an abstraction of a priori untrusted hosts interconnected through a network. Second, as a secure content wrapper where the agent is responsible for its own security, thus releasing its content only upon successful fulfillment of payment and other specific policy requirements bound to the content. Finally, in order to assess this work a prototype commercial electronic newspaper environment is implemented based on this framework.

The contribution of this work are as follows:

- A set of requirements for the commercial exchange of electronic documents over open networks and from there on for the electronic publishing field in its broadest sense from an infrastructure point of view based on mobile objects (agents). These requirements are derived from the issues raised.

- A model for the commercial exchange and distribution of electronic documents over open networks such as the internet.

- A general framework for commercial electronic publishing over open networks such as the Internet, based on the mobile object (agent) paradigm and the document distribution model. This framework represents an abstraction of the electronic publishing field from the point of view of a common infrastructure shared by different classes of electronic publishing applications. Thus considering these applications as classes of applications belonging to an application layer sitting on top of a common infrastructure.

- The implementation of a hypermedia electronic newspaper environment, HyperNews, as a specific class of commercial electronic publishing application used to assess and evaluate the HyperMedia Electronic Publishing framework and model.

1.4 Thesis Overview

In this introduction we have tried to set the general context and the major driving forces in the background of this thesis. In other words, we tried to state how we came to address this very challenging issue of electronic publishing.

The second chapter presents an overview of the various domains in the background of this work basically as a set of enabling technologies, major influences and more recent achievements.

The third chapter presents the issues raised by electronic publishing in a commercial environment. The discussion is centered around four main lines: the information consumer, the information provider, the information goods and the infrastructure enabling their trade. From the issues raised, a set of requirements are identified towards a framework for commercial electronic publishing.

The fourth chapter describes the Hep framework based on the requirements identified in the third chapter. The use of the agent technology is discussed. Then, a model for the commercial distribution of electronic documents is presented. These are then used as the basis of the core environment for the commercial exchange of electronic documents as the resulting Hep framework.

In chapter five, the Hep framework is used to implement a hypermedia electronic newspaper publishing environment called HyperNews. The specific application domain is described and the resulting prototype is presented before evaluating and assessing the results of this work.

Finally, chapter six concludes this work. Open issues are discussed and future research directions are considered before some concluding remarks and discussion.

Chapter Two | *Background*

Our purpose in this chapter is to sketch the landscape in the background of the numerous issues that have influenced, contributed and enabled commercial electronic publishing over open networks such as the Internet. Although still being far from having reached its full commercial potential and cruising speed, it carries a tremendous number of expectations for a very large number of people, communities and activities. It is about to turn the Information Age into reality upon entering the third millennium.

Although *Electronic Publishing* is now a well established term it conveys as many meanings as the wide number of communities it spreads over. People talk about electronic publishing in a variety of disciplines where it has different meanings, implications and definitions, for example: hypertext, typesetting, printing, databases, information retrieval and data mining, multimedia, linguistics and language processing, human computer interfaces, distributed systems, artificial intelligence, law, business, etc. It is thus difficult and almost impossible to find a single agreed upon definition of electronic publishing. Reason for which we will not attempt to provide yet another definition that would only suit the purpose of our work. However, we find it useful, to put our work into context and define our angle of attack on the domain.

We have found extreme difficulty in identifying “a” background and organizing this chapter as you probably expect to see as many issues covered as the great number of domains and communities this subject is directly and indirectly linked to. For this reason and as a notice to the reader, we do not claim that the content of this chapter and /or its structure describe thoroughly and to its best a clear cut background of the field, but rather provides an insight and a snap shot of the author’s point of view in trying to sketch the background of this work.

Without going back to the abacus, Babbage and Lovelace, the history of computer science and the papyrus, there are a number of issues and contributions that are worth mentioning in trying to set the foundation of our work, explaining where we come from and acknowledging previous contributions, enabling technologies and more recent achievements without which this work would never have existed.

The rest of this chapter is organized the following way. We will start by describing some fundamental contributions and enabling technologies for the field. Then, more recent achievements will be presented and discussed through existing systems and current technology.

2.1 Memex

Towards the end of World War II, Vannevar Bush who had coordinated the American war oriented research as director of Scientific Research and Development, published an article in The Atlantic Monthly of July 1945 under the title “As we may think” [5]. With this article, and like many visionaries that coined great ideas ahead of time, Bush is probably the father of a number of still very active research areas such as hypertext, digital libraries and electronic publishing among the major.

The vision of Bush was probably motivated by his notice of how research was done before the war compared to how it was done towards its end. Now that scientists and researchers had experienced to cooperate and thus learned team-work through a common goal, Bush urged to capitalize on this huge body of emerging knowledge, cooperation and know-how towards a task he identifies as “making more accessible our bewildering store of knowledge”.

Most importantly in his paper Bush sets forth a vision which is twofold. First he elaborates on the idea of *selection by association* claiming that it follows the mode of operation of the human mind which he claims can be mechanized. This in turn leads to the notion of associative trails. In doing so, he sets the foundation of what will later emerge and be called *hypertext*. Second, in order to support his claim, Bush imagines a mechanical device he calls the “memex” in which every individual would be able to store data, books, communications, annotations and most importantly: associations. The later being the “essential feature of the memex. The process of tying two items together is the important thing”. He describes the memex as “an enlarged intimate supplement to an individual’s memory”. Following, in his article, is a comprehensive description of the memex device relying on microfilm, dry-photography, mechanical levers, etc.

The memex device resembles stunningly what would much later (almost fifty years) be known as the World-Wide Web [2] implemented in the form of a piece of software composed of a rendering language HTML [3], a communication protocol supporting hypertext HTTP [4] running on hosts interconnected by the Internet. Similarly, memex is also in the background of what is now known as digital libraries.

2.2 Superdistribution

The concept of superdistribution, as coined by Ryoichi Mori in 1987 and described in a paper published in 1990 [6], is probably among the most prominent pieces of work in our background. Initially, this idea was first invented by Mori in 1983 and was known as the “Software Service System (SSS)” [7].

This idea was originally aimed at solving the crucial problem of software distribution enforcing fair compensation to software producers and protection of the software against modification with the least possible burden from the user’s point of view. Mori observed that while trying to detect whether software was copied (i.e., software piracy) was particularly difficult, it was easier or almost trivial for a program to detect and monitor its use. From there on, Mori proposes a model where programs are encrypted prior to their release, thus enabling and allowing wide and uncontrolled copying and distribution without any problem of piracy since payment be-

comes bound to usage rather than to acquisition of the software. Mori describes a set of four desirable properties that must be satisfied for software superdistribution:

- Software products are freely distributed without restriction. The user of a software product pays for using that product, not for possessing it.
- The vendor of a software product can set the terms and conditions of its use and the schedule of fees, if any, for its use.
- Software products can be executed by any user having the proper equipment, provided that the user adheres to the conditions of use set by the vendor and pays the fees charged by the vendor.
- The proper operation of the superdistribution system, including the enforcement of the conditions set by the vendor, is ensured by tamper-resistant electronic devices as digitally protected modules.

The resulting proposed superdistribution architecture relies on three principal functions:

- Administrative arrangements for collecting accounting information on software usage and fees for software usage.
- An accounting process that records and accumulates usage charges, payments and the allocation of usage charges among different software vendors.
- A defense mechanism, utilizing digitally protected modules, that protects the system against interference with its proper operation.

In Mori's design, computers are equipped with devices he calls Superdistribution Box (S-box). Computers equipped with such devices become S-computers. These boxes are to be understood as tamper resistant devices embodying microprocessors, RAM, ROM and a real-time clock intended for storage, processing and management of sensitive elements such as deciphering keys and other aspects of the superdistribution system. It is noteworthy to mention with respect to this specific issue that current trends in electronic commerce and security still follow this interesting idea of tamper resistant secure device for smart card readers and cryptographic devices. The resulting encrypted software together with its usage terms and conditions is called an S-program. Its permanent encrypted state has the very nice property of enabling it to be transmitted over untrusted and insecure communication channels. This is the exact property which is needed for communicating over today's open networks like the Internet. Furthermore, since programs are encrypted they can be copied and distributed by anybody without causing any prejudice.

From an operational point of view, the S-box holds a metering program called the Software Usage Monitor (SUM) in charge of enforcing usage terms and conditions set by software vendors and of tracking the fees (i.e., software usage units called S-credits) owed to each vendor. The S-box then generates payment files which are encrypted and sent through the network to collection agencies which in turn transmits payments to vendors. A clearinghouse, which may be a credit card company, keeps track of funds transfers in the superdistribution system.

In doing so, Mori turns a major drawback into a major asset. Namely, the inherent nature of software that allows it to be copied and distributed in a marginal, cost-effective way, turns out to be a real asset. In this scope, users become themselves “legal” re-distributors of software they like and use most. Based on this work, two prototype S-box systems were built. The first one based on a NEC9801 personal computer in 1987. The second prototype built as a co-processor for a Macintosh in 1990.

It so happens that apparently Mori was not the only one working on similar ideas. Brad Cox claims to also have been a pioneer in the field of superdistribution. In 1984, Cox came up with a similar design without knowledge of Mori’s work. This design was documented in a notarized patent workbook which he never filed for reasons explained in his book [9]. Cox gave it the name of CopyFree Software in the sense that software could be copied and distributed for free, but revenue collection would be based on usage. Later on, in an article published in Sept. 1994 in Wired Magazine [8], Cox describes superdistribution (meterware) as a possible foundation of a new networked economy. In 1996, Cox published a book on Superdistribution, “Objects as property on the Electronic Frontier” [9] comparing the challenge faced by electronic goods in the information age to the pony express days of the Wild West of America. He calls this challenging process of hauling goods made of bits rather than atoms in an emerging networked economy: “taming the Electronic Frontier”.

2.3 Secure Content Encapsulation or Binding Policy to Content

More recently, coupled with the general advent of the Internet and emerging electronic commerce, some commercial systems based on work done by Mori and Cox on superdistribution have appeared. Among the major, we find IBM’s cyptographic envelopes: Cryptolope [10] [11] and InterTrust’s digital box: DigiBox [12] [13]. Later, other systems appeared such as SoftLock of SoftLock Inc. [14] and Folio4 products of OpenMarket Inc. [15] which also address similar issues. These systems have a strong emphasis on content commercialization, copyright protection and usage metering. Unfortunately, they have major limitations in that they bind their users to proprietary systems or commercial partners and networks. The main characteristic of such technology is to bind the usage policy to the content in a secure way. This approach of “boxing up bytes” is commonly known under many terms such as cryptographic content wrappers, boxology, secure content encapsulation, etc.

2.3.1 Cryptolope

Cryptolope is a Java based software relying on three components. First, the *Cryptolope Builder* can be thought of as a packaging tool allowing to build the cryptographic envelope holding both the content and the business rules for its use. This tool is basically to be used by content providers. The second component which is intended to be used by information consumers, the *Cryptolope Player* is the interpreter for accessing the Cryptolope content. It uses a trusted HTML viewer and interacts with the *Cryptolope Clearing Center*, which is the third component of the architecture. It is basically a trusted third party providing key management, payment system and event logging / usage metering. The major problem faced with their approach was that it was a closed proprietary system. Users were forced to use IBM’s InfoMarket infrastructure for the clearing center acting as a trusted third party thus binding them to IBM. This is probably the

reason for which Cryptolope has not encountered the anticipated success. In fact, a key factor of success for this type of technology relies in how open it is to integrate other commercial partners be they clearing centers for copyright and/or usage, financial institutions or content providers.

Although Cryptolope made a noticeable appearance on the market in early stages, it now seams to be in a frozen state. The Cryptolope web site has been down for many months, as IBM's InfoMarket to which it was linked. Initial examples, downloads and demonstrations are no longer available. Now, only a few pages are left on the IBM web site describing very briefly the Cryptolope technology and its components. Moreover, these pages have not evolved or been updated for months. However, this technology as a separate product has not been abandoned by IBM but rather used within several IBM / Lotus products for e-Business or Digital Library, etc. They do consider licensing the technology including source code for a fee on a case by case basis. But in any case, it seems very unlikely for the time being and according to them, that this technology will have any future as a product per se.

2.3.2 DigiBox

The DigiBox [12] technology (by analogy to the idea of a digital box) is probably the leader in the field currently. This technology developed by STAR Lab [16] (Strategic Technologies and Architectural Research Laboratory) is also a secure content wrapper technology which is the foundation of a commercial product, Commerce 1.0 and Enterprise 1.0, of InterTrust Technologies Corp.

The DigiBox architecture is a secure content wrapper. In their approach content is called *properties* and the policies defining their usage is called *controls*. A DigiBox can hold one or many properties as arbitrary data. The controls can be delivered in the same DigiBox or independently in a separate DigiBox. Controls are linked to properties by cryptographic means. Here is an example taken from the paper [12] showing how it works: a DigiBox (#1) holding two properties P1 and P2 as well as a regular control set CS1 linked to P1 would not allow its user to access property P2. However if the user acquires another DigiBox (#2) holding two control sets CS2 and SC3 linked to properties P2 and P1 respectively (of DigiBox #1), the user would have the choice of using either CS1 or CS3 to access P1. However with this second DigiBox, the user can now access property P2. Figure 2.1 shows this example. To be noted that controls can apply to arbitrary parts of the properties such as bytes, frames of a movie, etc.

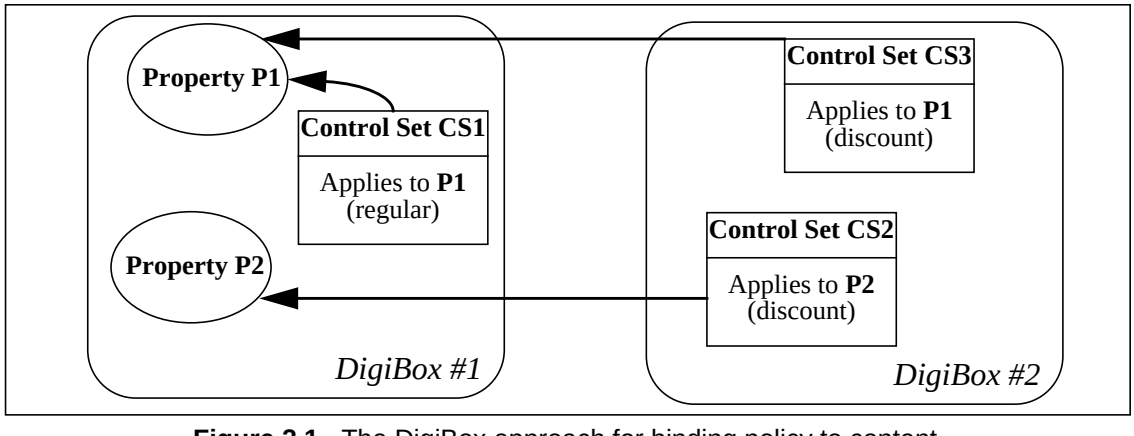


Figure 2.1 The DigiBox approach for binding policy to content

In a DigiBox, high level elements such as headers and general information are encrypted with a transport key. Properties are encrypted with other keys which can be delivered separately if needed. The transport key is composed of two parts. One which is included in the digibox and will be combined (XOR) with an other one stored locally in protected storage where the DigiBox is to be opened. The part included in the DigiBox is encrypted with a public key algorithm. The main advantage of this is that it protects against the threat of having any of the two keys compromised. However this approach requires to distribute the keys among the participating parties (i.e., key management). Moreover it requires on every host a secure storage which is called an InterRights Point. The cryptographic algorithms used are Triple DES and RSA and integrity verification is done with a cryptographic hash function [17].

Once the DigiBox is about to be opened according to the controls governing this process two different flows of information can occur. The first one towards the financial clearinghouse for billing purposes. The second one, if required within the control set of the DigiBox, towards the usage clearinghouse for collecting usage and metering information to be returned. Hopefully provided that the user is aware and has agreed on such a feedback loop.

The key benefits of the DigiBox approach is its support for both superdistribution (provided the controls are within the same DigiBox as the properties they are linked to) and separate delivery of properties and controls. The architecture directly supports off-line transactions due to its key management policy. However this has a cost in terms of key management which must be distributed among the participating actors by means of special DigiBoxs called “Directed DigiBox”. Applications that want to use the DigiBox architecture must be certified by InterTrust. Within the InterTrust system, all participants have unique IDs. DigiBoxs are assigned unique identifiers throughout the whole system. Thus it would also be possible to use content identification schemes such as Digital Object Identifiers (DOI) [18].

InterTrust released in June 1998 its initial version 1.0 (FCS) to licensing partners. Current issues of interest at STAR Lab include Commerce Control Languages, tamper resistant devices targeted towards better security for the InterRights Point. Watermarking techniques are also of interest as any piece of content could be disclosed from the digital world back on a support made of atoms. In which case if the released content holds a watermark identifying uniquely information such as the copyright holder and to whom it was last revealed could eventually serve in a court of law. Expanding on the idea, Jim Horning (Director of STAR Lab), also considers the issue of encoding as well the terms and conditions (i.e., controls) within the watermark. This would have the nice property of allowing legal digital re-use of material which would otherwise have simply been disclosed forever.

Recently, InterTrust has allied with NatWest [19] probably towards financial clearing house services. Trial is to take place towards the end of 1998 and the global launch is expected for early 1999. A pilot program has been launched by Rights Exchange Inc. [20] based on InterTrust’s Commerce 1.0 system.

2.3.3 Other Systems

SoftLock [21] of SoftLock Services Inc. is basically a password based locking mechanism for software and documents. Products automatically re-lock upon moving them from one machine

to another. Passwords are acquired by purchasing them over the phone from a toll free number on a 24 hours a day basis or from their web site where you are proposed the product description and price based on the product number. Upon acceptation of the terms and conditions the client is billed and the password delivered to the user by e-mail.

SoftSEAL [22] [23] of Breaker Technologies Ltd. is a plug-in based system for on-line license acquisition. Content and access policy (i.e., licenses) are separated. Licenses are purchased from a license server on the Internet. Once the license is held, viewing the content is like viewing free content. However, the Web site seems to be "abandoned" and holds many dangling links.

Folio4 products and *Folio SecurePublish* [24] of Open Market Inc. provides a whole set of tools for content management and publishing where information is secured but access is based on the purchase of the access rights. They claim future versions will provide "pay per document" facilities.

2.4 Of Bits and Atoms: The Copyright Hassle

These systems described above are now known under many terms such as secure content encapsulation, boxology, cryptographic content wrappers, etc. They are becoming more and more popular as copyright and intellectual property laws have trouble being adapted to the digital nature of content on open networks. This is probably the reason for which the debate on public policy or legal enforcement versus technology arose recently in the scope of a global market where goods are made of bits instead of atoms for which copyright law and intellectual property rights provide a legal framework such as the Bern Convention [25], WIPO treaties [26] and constitutional rights such as in the US constitution[1].

Actually, the debate is two fold and its second aspect depends on the first one. Namely, the first aspect is more a social aspect where the debate is centered on whether "digital information / content wants to be free"? We will not discourse upon this issue as it is out of our scope. The Cover story of the September 1998 issue of The Atlantic Monthly: "Who will own your next good idea?" by Charles C. Mann [27] provides an in depth description on the issues raised and which are at stake. Moreover, an excellent roundtable on this issue can be found in the September 1998 on-line publication of The Atlantic Unbound [28] where Charles C. Mann debates such questions as freedom of information with other prominent panelists such as Mark Stefik (principal scientist at Xerox Parc), Lawrence Lessig (professor of law at Harvard) and John Perry Barlow (co-founder of the Electronic Frontier Foundation). However if the answer is to be that information is to be free and evolve in a chaotic information world, which we think is most improbable, then the whole issue centered around technology to enforce copyright and intellectual property becomes non pertinent.

The second aspect of the debate, assuming that copyright and intellectual property is a pertinent issue, is more a techno-centric issue facing the problem of finding the "right" equilibrium between public policy, law enforcement, privacy and technology as a service to the issue of copyright and intellectual property. In this respect, the right question was asked by Pamela Sam-

1. US Constitution, Article 1, Section 8, Clause 8

uelson (Professor at he University of California at Berkeley) in a plenary lecture given at the HICSS-31 conference in January 1998 [29]: “What information society de we want to live in?”. In her talk she was fairly optimistic for a “cool information society” and concluded by urging computer scientists to come up with ideas hoping that technical protection systems “will do the thing”. It is indeed our opinion that technology will not be the only means by which all our problems will be solved. However finding this equilibrium between technologically enforced mechanisms and *ethico-legal* issues is “the” challenge keeping in mind constraints such as privacy, fair use, true market competition, protection of all the economic actors but most of all the less possible public policy (i.e., political intervention). Commerce is a very old and mature discipline. Law has a role to play in commerce from the point of view of a framework. However, within this framework, it is the market through its actors that regulates what happens. Thus working on digital equivalents of century old concepts such as the “terms and conditions of a contract” is a very interesting issue. Work is being done in this field for example: Mark Stefik’s Digital Property Rights Language [30] or work under way at InterTrust’s STAR Lab on Commerce Control Languages according to Jim Horning, Director of the Lab.

2.5 Electronic Commerce

Electronic fund transfers and electronic data interchange (EDI) over financial and private networks have been around for some time now. These infrastructures are expensive and the Internet has paved the way for wide scale electronic commerce over open networks. But, many issues still need to be addressed before achieving this goal in a safe open and secure way. This section describes the background and the issues of electronic payment technologies over the Internet through existing systems in order to draw the requirements for an electronic document distribution architecture.

In most electronic commerce systems there is a trade-off between efficiency, security and cost. In a general way, electronic commerce systems can be classified in different categories depending on how they deal with the following issues which impact on this trade-off:

- **Microtransactions** : this issue is important specially in the scope of our work since the goods are information goods and their costs may be as low as a few cents or even fractions of a cent. Credit card and alike payment systems aren’t suited for these kind of deals since the transaction cost would be much higher than the amount to be paid. However, it can be anticipated that an increase of the transaction volume is likely to reduce significantly the threshold above which such credit card based transactions would become commercially viable.

- **Security** : the main concern of security is to provide, throughout the whole chain of the commercial transaction, the means to ensure the trustworthiness of all the parties, the efficient encrypting and digital signatures of protocol dependent authorizations, authentication requests, digital receipts and alike. In this context key length and lifetime of encrypted content are important issues.

- **Anonymity** : this issue faces the problem of hiding the customer’s identity in a way similar to the use of cash. Different techniques can be used like certified tokens, blind sig-

natures, pseudonyms. Moreover, for low cost goods and every day transactions one might not want these transactions to be traceable for many reasons.

- **Privacy and secrecy** : this issue is more concerned by the protection of the content of information goods such as copyrighted documents and the privacy of users. It could also to some extent concern the aspect of certifying that the received document is conform to what has been sent.
- **On-line and off-line payment models** : On-line payment models refer to systems involving a third party during the commercial transaction for authentication and authorization reasons. Whereas off-line payment models only involve the customer and the merchant. The first situation prevents easily problems of double spending and dishonest transactions. In the second situation, these problems could be solved by secure hardware components such as smart cards.
- **Payment scheme** : there are many different possible schemes depending on who creates the “electronic money”, when and how customers and merchants are debited or credited with real money, who initiates the commercial transaction. Basically, there are three models: debit, credit and cash like.
- **Repudiation of transactions and dispute handling** : how the systems deal with these problems in order to enforce trustworthy trade from end to end even in critical cases. What is the legal value of a digital signature, how does it bind the commercial parties, how are litigations handled and who has the authority to settle them etc.

The *Millicent* protocol [31] developed at the DEC SRC, is best suited for micropayments over the Internet (i.e. less than a cent). The model followed by this protocol is that of a trusted third party called a *broker* who’s role is to serve as an accounting intermediary between customers and merchants. A *scrip* is a piece of digital cash that is only valid between a given customer - merchant pair. Scrips are obtained from brokers which themselves obtain them from merchants. The scrip contains a value and when a customer makes a purchase with it, the amount of the sale is deducted from the scrip’s value and sent back to the customer as change. A scrip can be considered as an “account” between the customer and the merchant which is set up, used and closed. A scrip can not be spent more than once, can only be spent by its initial owner and at a specific merchant, has an expiration time and can be regenerated upon expiration. Different Millicent protocols offers various levels of security and privacy. Namely, *Scrip in the clear* offers no security and no privacy. The *private and secure* protocol uses a shared secret between the two parties to establish a secure communication channel. And the *secure without encryption* protocol does the same without the privacy aspect in order to achieve better performance.

The *NetBill* [32] system from Carnegie Mellon University is a set of protocols for micropayment of information goods on the Internet. It is composed of a set of protocols involving customers, merchants and a NetBill server (i.e. an account server which is linked to conventional financial institutions). This enables the aggregation of many small transactions in to larger transactions. In this model, the NetBill server acts as a trusted third party ensuring the atomicity of the transactions (i.e. payment and delivery of the information goods). There are three phases in a NetBill transaction. In the first phase, the customer requests a price offer for the desired item.

At this stage a bid for that item can also be included as well as personal information in order to qualify for special prices (e.g. student ID, frequent buyer ID). The merchant replies with a price quote. The second phase is initiated by the customer acceptance. The encrypted information goods are then sent to the customer but the decryption key will only be sent after completion of the third phase, namely the payment. In this third phase, the customer sends a digitally signed payment order to the merchant. The merchant includes the decryption key to this order, digitally signs it and sends it to the NetBill server. Then, upon completion, the NetBill server sends back to the merchant a digitally signed receipt including the key. The merchant forwards a copy of it to the customer. The system allows also for easy use of pseudonyms for anonymity but in any case, the NetBill server knows both parties identity and transaction amounts. However, it can ignore everything about the content of the goods through simple encryption techniques.

The system developed by *Digicash* [33] relies on a software based electronic money system called *Ecash*. Both clients and merchants are given an Ecash software which can be thought of as an electronic wallet in which withdraws from a bank can be put in or deposits to a bank can be made to. The system relies on the bank who's role is to act as a back-end to the system to certify that the electronic payments are valid (i.e. that coin signatures are valid) and to serve both the customers and the merchants. The system allows also for person to person payments and for customer anonymity through "*blind signatures*".

The *CyberCash Secure Internet Payment Service* [34] is based on a metaphor of physical payment. Customers are given CyberCash Wallets. Merchants use the Secure Merchant Payment System (SMPS). The CyberCash Gateway Servers are operated by CyberCash and in the future by banks. This system relies on the use of existing financial networks which are totally independent of the internet for communicating between the CyberCash Gateway Server and the banks or credit institutions. For the time being, it supports only credit cards. In the future, it will support electronic checks, electronic cash and micropayments. A transaction is initiated by a customer clicking on a merchants "PAY" button. This action generates an electronic order form at the merchants site which is sent to the customer. The browser opens the CyberCash wallet window thus allowing the customer to select a payment instrument (credit card). Upon confirmation of the amount by the customer, the information is sent encrypted to the merchant which appends his own identification information and passes the payment request to the CyberCash Gateway Server. After validation of the payment request an acceptance or denial information is sent back to the CyberCash Gateway Server which is forwarded to the customer as a digital receipt. The system uses 56 bit DES private-key to encrypt all the messages between wallets, merchants and CyberCash Gateway servers. The DES key, which is unique for each transaction, is then encrypted using RSA public key technology. The length of the key is currently 768 bit but 1024 bit has been approved by the United-States government and will be eventually. Finally, a digital signature is appended for source authentication and non-repudiation purposes. CyberCash is following the credit card institutions and will be compliant with the Secure Electronic Transaction specification (SET) [38] defined by the major credit card institutions namely Visa and MasterCard.

NetChex [35] is an electronic check system on the internet. Consumers must be registered at the *Net 1 Inc.* A local software (Windows based) is responsible for secure communication with

the NetChex system which then processes the transaction through traditional banking systems and networks. The issuer of a NetChex check is notified upon completion of the transaction by e-mail.

CAFE [36] (*Conditional Access For Europe*) is an ESPRIT project that developed a secure electronic payment system that protects the privacy of the user. It is based on smartcard technology for electronic wallets. These electronic wallets look like pocket calculators or PDAs (Personal Digital Assistants) in which smartcards are inserted. The system acts like a prepaid offline payment system. Users have to "load" electronic money from an issuer prior to spending it at points of sales. The system is multi currency and there is no need to contact an issuer or third party during a payment transaction. Communication between PDAs and other devices is made by an infrared channel.

The *Mondex* [42] system is also based on similar smart card technology. It follows the electronic cash wallet paradigm by storing the electronic cash on an encrypted microchip. The security scheme relies on a digital signature which is generated by the chip on the card. This digital signature can only be recognized by other Mondex enabled participants in a transaction. It has been designed to support multiple currencies (five) and to allow person to person transactions. Various trials and pilot have been undertaken involving banks (NatWest and Midland in the UK, HongKong Bank outside the UK) and British Telecom for the telephone infrastructure.

Internet Keyed Payment Protocol (*iKP*) [37] is a credit card based Internet payment system developed by IBM. It was part of the ACTS project SEMPER. It contributed in 1995 to MasterCard's SEPP specification which led to MasterCard and Visa's joint SET specification. The system relies on the following assumptions: both customers and merchants must have an existing relation with a financial institution. The customer must have a credit card. The merchant must have a contract with an acquirer accepting the client's card. The system uses traditional encryption algorithms (i.e., RSA). The iKP system has three variants. In the *1KP* variant, only the acquirer has a public key. In *2KP*, both the acquirer and the merchant have public keys (i.e., both can sign). Finally in *3KP* all three parties have public keys.

The *Secure Electronic Transaction* (*SET*) [38] specification is the common standard that is developed by two of the major credit card institutions (Visa and MasterCard). This common effort is the result of an agreement which took place early in 1996. American Express has also announced that they would support the SET specification. Some of the major partners in this agreement include GTE, IBM, Microsoft, Netscape, SAIC, Terisa Systems, and Verisign. In the background of SET there are two competing individual efforts, namely SEPP for MasterCard and STT for Visa. The current specification does not account for microtransactions nor for smartcard technology. SET is based on both asymmetric (e.g., RSA public key can be used for digital signatures) and symmetric (e.g., DES) cryptography.

The *Java Electronic Commerce Framework* (*JECF*) [46] developed by JavaSoft is a promising solution for the integration of existing payment instruments such as credit (using the SET protocol) and debit cards, as well as emerging solutions like electronic cash, electronic checks and smartcard technology. The system will support microtransactions, frequent buyer style advantages, procurement cards, coupons, etc. Currently, the JECF is now called Java Wallet. It is

version 1.0 is shipping in its early access 2 while the final implementation is expected to be released any time soon.

The overall feeling that prevails in the field of electronic commerce on open networks is that many proprietary protocols and trials have been set. Each one of them addressing some specific issues. There is a clear need for standards addressing all the issues and providing well defined requirements for API developers integrating in a unified way existing payment schemes and technologies. These standards should be flexible enough to allow easy integration of current and future payment schemes and technologies. For example U-PAI [48] (Universal Payment Application Interface) which is part of the Stanford InfoBus provides such a layer as an abstraction to payment protocols. An other similar approach called the Generic Payment Service Framework (GPSF) [49] is also oriented towards a unified interface to electronic commerce protocols and systems. It is also part of the work done within the SEMPER project [50]. Moreover smartcard technology in this field is still in its infancy but could be a promising solution for wide spread safe and secure electronic commerce in both networked and real world. It is gaining credibility and smart-card manufacturers are releasing Java based smart-cards that comply with the java Card API from Sun.

2.6 Mobile Objects and Agent Technology

While the client server model has received significant attention in recent years, a whole new domain of research has emerged from the combination of progress achieved both in object oriented research and networks. Mobile objects, mobile computations, mobile agents or simply agents [52] [53] [54] have reached the level of being a research area raising a set of issues dealing with security, distributed systems, networks, etc.

There are basically two agent communities. First, intelligent agents or multi-agents and second, mobile agents. The former, rises from artificial intelligence and is focused on knowledge representation, collaboration, behavior, avatars, etc. The later stemming from object oriented, network and distributed system research address the issue of moving behavior towards the source of data. An agent in this context can be considered as an object (compound or not) in the object oriented terminology having the following two characteristics: persistency and network awareness. Persistency in the sense the object holds state in a persistent way. Network aware in the sense the object has knowledge of network and is able to migrate between participating nodes of the network. This represents a major change for network applications thus leading to a new paradigm. Moreover, it represents a significant opportunity in the field of electronic commerce since agents can be considered as natural metaphors of commercial actors be they consumers, providers or even intermediaries (i.e., brokers, facilitators).

From these research issues and driven by market demand, a number of prototype systems and languages supporting this agent paradigm have been implemented (*Emerald* [55], General Magic's initial implementation: *Telescript* and current Java based technology: *Odyssey* [56], *Obliq* [57], *D'Agents* [58] formerly known as *Agent TCL*, IBM *Aglets* [59] [60], Object Space *Voyager* [61], *Mole* [62] [63], etc.) Some of which have already become commercial products which are now available on the market. However such systems represent a considerable chal-

lenge with respect to security issues. These have been identified and discussed thoroughly in [64] [65] and a prototype system called *Seal* [66] is currently being implemented.

2.7 Remark

As presented above, the background on which we base our work is relatively vast and wide spread. However, in the scope of our work we sit at the crossroads of many of these issues targeting a technology integration towards a framework for the commercial exchange of electronic documents over open networks based on mobile objects.

Chapter Three

Requirements for Commercial Electronic Publishing

Throughout the remainder of this work we will use the term *electronic document* or *electronic content* as a generic name identifying electronic information in a broad sense regardless of its type (i.e., text, images, sounds, movie clips, value added services, arbitrary data, etc.). Likewise, the term *electronic publishing system* will identify the infrastructure enabling the trade of information goods over open networks.

At a very abstract level, our general issue can be characterized the following way: information providers and information consumers participate in a commercial transaction where at least one of its component is in digital form and as such can be carried out over an electronic infrastructure. We will now consider the issues centered around these four main lines and draw their corresponding requirements towards a framework for commercial electronic publishing.

Our discussion in this chapter is centered around the consumer, the provider, the information goods and the infrastructure enabling their trade. The requirements presented in this chapter are summarized in tables that can be found in Appendix A and Appendix B.

3.1 Information Consumer Issues and Requirements

The information consumer is the end user of an electronic publishing system and as such, he is entitled to be provided with a high quality service. From this point of view, we discuss below the issues relevant to the design of such an electronic publishing system and identify the corresponding requirements.

3.1.1 User Interface

- *information consumers should be able to use any widespread rendering interface as an electronic publishing reader.*

Current trends in user interface are oriented towards simplicity and consistency among applications. Taking a very popular example based on currently available technology, the World-Wide Web has become a very popular media and browsers are now available on most platforms. Moreover, the user interface of Web browsers is intuitive enough to be used even by children. Thus, it would be an advantage to enable the users to access in a seamless way both their electronic publishing applications and popular browsing, rendering applications through the same unified interface. Based on current technology, this could be a Java enabled World-Wide Web

browser. Note that it is not mandatory but when considering this issue it can be seen as a key factor of success specially when targeting general non technical users.

3.1.2 Free Choice of Information Providers

- *information consumers should be able to choose their information providers.*

We assume that the information consumer is aware of the existing supply in any given information domain. The needed information sources, content and service providers, whether traditional or electronic, are chosen accurately, based on habits, experience and what is done in their environment. Thus selecting suppliers, providers in an accurate way is not a random process but rather a conscious decision of the information consumer.

Note that this approach of knowing in advance and selecting accurately providers does not compete with a traditional browsing approach which is different and closer to a “going shopping” metaphor where the consumer does not have a precise idea of what he is looking for. However even this later case can be argued to be a service in case the user chooses a specific information broker.

3.1.3 Customized Information Selection

- *information consumers should be able to specify their information interests for every information provider (information profile).*

In order to illustrate the notion of customized information selection, we consider how an information consumer accesses a printed document. It is quite common to access the same information source for different reasons (e.g., in a newspaper, finance, arts, sports) or to read only a specific section of a number of documents (e.g., executive summary, white papers, bibliography). Currently with paper based information sources the whole document is bought, although only a subset of it is used. The needed sections are accessed almost directly and can be then browsed or scanned. Implicitly an information consumer has his mental representation of the structure of a given information source. This allows for direct access to the needed section of the information source depending on the context of work or use.

For an electronic publishing system, the *information profile* specifies an exhaustive set of information sources and information interests for each source. Or in other words *what* and *where from*.

3.1.4 Customizing Presentation

- *information consumers should be able to specify the general structure of their electronic publishing application (presentation profile),*
- *information consumers should be able to specify the content of each structure element of their electronic publishing application (presentation profile).*

Finding and physically showing together related information from different information sources is time consuming and error prone. However it can be very useful to bring in-depth understanding of an information through different views. Such cross information source *linking* is currently done “manually” by the reader. However, for an electronic publishing system, this can

be achieved by providing means for personalized presentation of information. The goal is to offer both a consistent layout over time and physical proximity of related information independently from the source. This in a way decided by the user. For example, on the first page of an electronic newspaper environment, one might want the closing values of the Dow Jones, the Nikkei and the CAC40 from one provider, the weather forecast of the day from another provider and the up to date version of a business report, etc.

For an electronic publishing system, the *presentation profile* specifies how information is to be presented and can be thought of as a wire frame of the user's electronic information application that will be instantiated at retrieval time. It defines the general structure as well as the content of the structure elements. In other words it defines *how* it is to be presented.

3.1.5 Context Dependent Electronic Publishing

- *information consumers should be able to have multiple instances of an electronic publishing application (information contexts).*

The situation in which information consumers hold concurrently different positions and responsibilities is rather common. It is also common to be involved in different projects, especially as responsibilities grow. Thus, depending on the role played by the information consumer, different information sources, sections, topics and their corresponding presentations are needed.

For an electronic publishing system, we call the elements defining the information needs and their presentation *information context*. An information context is equivalent to an instance of an electronic information application. An information consumer can have multiple information contexts (e.g., for an electronic newspaper environment this would represent multiple personal newspapers, for a library bibliographic environment this would be different perspectives of the library).

An information context includes two elements: first, a list of information providers along with information interests for each source and second, the description of how the information should be presented to the consumer. The two parts comprising the information context are the *information profile* and the *presentation profile* as presented earlier. For example, a user can have a work context, a private context, a context on a number of specific topics or projects, etc.

In summary, the presentation profile together with the information profile are bound to an information context (i.e., an instance of an electronic publishing application). They provide the information consumer the way to specify where information is to come from, what information is of interest and how it is to be presented at retrieval time for each information context.

3.1.6 Active Information or Notification of Update Availability

- *information consumers should be able to be notified when information updates are available on issues of interest (active information).*

Information is dynamic by essence. It can only be considered static when reflecting a state of fact at a given point in time. We provide below two examples on this issue. The first one is taken from the newspaper and magazine like publishing industry whereas the second example

is taken from the book publishing industry. These two industries are significantly different from the point of view of the type and attributes of their content.

Currently, newspapers, magazines and alike are published on a regular predefined time interval basis. The information contained in a given edition is considered static in the sense that it was valid when printed. Nothing however guarantees that it has not evolved by the time it is read. Any evolution of the information will only be traced in the next edition. For immediate information updates other means and media have to be used. For example on-line information feeds such as Reuters, Bloomberg, news casting channels, mailing lists, etc.

In the book publishing industry it is also common to find different editions of the same book reflecting evolution of the matter over time or simply its continuation. Here again when accessing a given edition nothing will tell the reader that a new edition of the book exists and is available. The reader will have to either be registered in the editor's mailing list or follow the specialized publications or even take an educated guess based on the date of publication and his knowledge of the field.

In an electronic publishing environment, means are needed to offer the information consumer the possibility of being notified upon availability of information updates (i.e., active information). This follows a "pseudo push" model of interaction where the information consumer can request to be "fed" with or notified upon information updates on issues for which he has expressed interest.

3.1.7 Information Evolution and Referencing

- *information consumers should be able to access easily any point of the historical evolution of an information,*
- *information consumers should be able to access directly any referenced material both inside and outside the domain of the information provider.*

There are situations where consumers are either not able or not willing to access their information for many reasons including vacation, illness, sudden high priority events, etc. In such cases, newspapers, books, reports, information in a broad sense will piled up for later access. In order to catch up, the information consumer might jump to the most recent edition / version or just start with the first unread one. In any case, for tracing either backward or forward the evolution of a topic the consumer will have to browse through all the editions building his own information evolution with risks of missing something important. Such a task is time consuming, error prone and cumbersome.

An electronic publishing environment can very easily provide links between each step of the life-cycle of an information or a piece of content (events happen, evolve and terminate) through hypertext links since information are related one to an other. For example, when writing an article, a columnist can easily include a "follow up on" link to relate it to a previous article. A book can have a link to its previous edition, etc. Moreover, hyperlinks can also be used to refer to archives, other encyclopedic information or related material both on an internal (i.e., bound to the same information provider) and external basis (i.e., referring to material of a different information provider).

3.1.8 Anonymity

- *information consumers should be able to retain full anonymity when desired.*

In the traditional publishing industry, the information consumer has the choice of revealing or hiding his identity to the publisher. By revealing his identity, through for example a subscription or a membership, the consumer can enjoy several benefits such as lower price, frequent buyer bonus, personalized information, special issues, etc. However, by buying an edition from a shop the information consumer can retain full anonymity. Accessing information in an anonymous way is very important since information providers can easily build profiles based on consumer habits, often leading to unwanted solicitations and targeted advertisement. Besides, the piece of information read is itself information which the reader might not want to divulge. For example, knowing that a well known investor is suddenly interested in some small venture company could lead to unpredictable speculations having an impact on an IPO[1] or consequences on the stocks of that particular company. Furthermore, anonymity and privacy can be crucial when considering new job opportunities or even desirable when reading “sensitive” documents. Finally, usage metering of consumers through hidden feedback loops is highly undesirable unless the consumer is aware of it, possibly against advantages of some sort. Such information is intellectual property and there is a market for that also.

In an electronic publishing system the information consumer should retain the possibility of revealing or hiding his identity. The electronic information consumer should be able to pay for information without having to reveal in any direct or indirect way his identity to the information provider. The problem however is that unlike real cash, the provider can easily find out who the consumer is and what is read from the information included in an electronic payment [67]. A possible solution is to either introduce an intermediary acting as a trusted third party between the consumer and the provider and thus bound to contractual agreement, or to use an electronic cash system based on anonymous money or payment mechanisms.

3.1.9 Information Usage and Access

- *information consumers should be able to pay on a usage basis,*
- *information consumers should be able to hold and redistribute freely self contained, self secured electronic documents,*

In the traditional publishing industry, the smallest information unit that can be put on the market is a *document*. Whether it is an *issue* of a publication, a report or a book is irrelevant, its price is fixed accordingly for the document as a whole. The price of a periodical for example does not vary if it holds more or less information than a previous issue. It is also not possible to either negotiate a discount when claiming that only a part of the document is relevant to the consumer, nor is it possible, in general, to ask a refund if the document has not been used at all.

In an electronic publishing environment however, the granularity of the marketable information unit can be brought down to any arbitrary level such as an article for a newspaper, a definition for an encyclopedia or a chapter for a book. The granularity of the information unit in the

1. Initial Public Offering (IPO): raising equity capital

electronic world depends on the information provider and his commercial policy (business rules). This allows for more flexible pricing policies where each information unit has a price and can be sold independently from any other. Combined with a security scheme enforcing payment upon access to the content of the information unit, the information consumer is provided with a *pay per use* system. This offers the major advantage of allowing information consumers to freely acquire, hold and distribute documents without infringing any copyright or intellectual property laws. This is also known under the term of *superdistribution* as described in chapter two.

- *information consumers should be able to access subsequently documents they have already paid for (provided the policy allows it),*

Subsequent access to a document that has already been paid for must be possible by providing a valid proof of purchase. However, only the user that has paid for the document should be granted subsequent access right through the proof of purchase provided it complies with the document policy.

3.1.10Payment and Electronic Commerce Systems

- *an electronic publishing system should be open to existing and future electronic commerce systems.*

Many electronic commerce schemes and protocols exist. They can be classified according to how they cope with issues such as support for micro-transactions, security, anonymity, off-line payment, dispute handling, pricing policies etc. as discussed in the previous chapter. In order to offer maximum flexibility to the information consumers, various payment methods should be accommodated by an electronic commerce system. This could be achieved by providing a common abstraction layer to electronic commerce. Thus allowing the user to specify payment preferences that are to be used and matched against when undertaking an electronic commerce transaction. Work has been done in this direction by SunSoft with the Java Electronic Commerce Framework (JECF)[45] [46] [47], the Universal Payment Application Interface (U-PAI) [48] within the Stanford InfoBus, and the Generic Payment Service Framework (GPSF)[49] within the SEMPER project. These among the major.

3.1.11 Off-line Activity and Information Consumer Mobility

- *information consumers should be able to access documents off-line.*
- *information consumers should be able to move their electronic publishing environment between hosts easily.*

Situations in which people require high geographical flexibility and mobility are common nowadays. Consequently, they often have access to many different hosts like for example a workstation at the office, a desktop at home, a laptop when traveling and a palmtop, PDA or other handheld devices on vacation. This raises two issues: first, off-line activity when a network connection is not available or possible; and second, synchronizing or transferring environments between different hosts. Concerning off-line activity, accessing electronic documents should be possible without any network connection. Of course, in order to acquire the data one must connect to a network but after that, the system should offer means for reading the downloaded in-

formation off-line, for example while going to work, on vacation or even from a CD-ROM. This in turn raises the problem of off-line payment systems. Thus, mechanisms for off-line payment must be considered for an electronic publishing system. For example, smart card technology could offer a clean solution to such off-line problems. This aspect is enforced by the recent release of Java Cards [41] which provides an API for programming smart cards. The current release of the API is 2.1 and is under public review. Concerning the issue of environment mobility, such situations require means to transfer easily a part or the whole environment between the different hosts.

3.1.12 The Information Consumer as an Information Provider

- *information consumers should be able to publish new documents embedding documents of other information providers together with their own added value.*

This last issue introduces in some way the next section discussing the issues and requirements of the information provider. In fact, we consider here the information consumer as an information provider. To illustrate this issue, we provide the following two examples where a user receives a document: (i) the user decides to forward it to a friend together with some free comments. (ii) the user decides to forward it to a client together with some comments that have to be paid for. In the first case, the comments are free. However, in the second case, the user has added value to the original document and will want to collect revenue for this. Moreover, in both cases, the user adding value has the right to protect his own intellectual property and be protected against content tampering. Thus in both cases a new document is created which contains the original document and the comments with their corresponding price and access policy. As a result, each level of added value is protected and authoritative.

The idea here is that an information consumer can become an information provider of his own added value and a reseller of other information provider's material without infringing any copyright or intellectual property law in the sense of superdistribution. This leads to a recursive definition of a document where a document can hold its own added value and other embedded documents provided that each document is able to protect and release its content according to the policy it is tied to. An other example taken from the real world illustrating this issue is the *Courrier International* [51] weekly edition. They publish a selection of the world's best daily newspaper articles. This shows exactly the notion of different levels of added value which all need to be protected against misuse.

3.2 Information Provider Issues and Requirements

The information provider wants to commercialize electronically information content and other value added services that he holds in a way that will secure and protect his intellectual property rights. From this point of view we discuss a number of issues related to the transition from the printed to the electronic media and identify the corresponding requirements.

3.2.1 Copyright Management and Revenue Collection

- *information providers should be assured that content will be protected and secured against illegal access and use thus protecting their copyrights and intellectual property,*

- *information providers should be assured that content access will generate payment to the copyright holder and thus be able to collect revenue from their electronic publishing activity.*

Copyright management is a crucial issue for publishers [68] [69]. A legal base exists for printed material in order to protect intellectual property and rights of author. These laws regulate correct usage of copyrighted material. However, very little has been done yet for electronic documents and data. The problems raised by copyright and rights of author in the electronic world represent a blocking factor for the publishing industry. Information providers need to be assured that electronic information items will be paid for, used as they should be (i.e. view only, authorized copying, distribution, etc.), retain copyright information both as a whole or part, etc.

As far as electronic documents are concerned, a scheme is needed to protect their content from being hacked and tampered with, while enforcing payment to the copyright holder. Protection against hacking and tampering can be achieved by using encryption and digital signature techniques. The security scheme used does not need to be 100% secure but secure enough to prevent generic attacks or failures, for example, the breaking of one document key leading to the access of all the documents. Moreover, breaking a key should involve sufficient effort for it to be either too costly or simply too long. This follows the real world model where photocopying a document is labor intensive and time consuming. Thus buying a new copy is often the preferred solution. Note that the security level to be used should be proportional to the lifetime and / or the value of the document. For example, newspaper or a magazine article almost become public domain after a day or so whereas books are likely to last decades.

The access to the content (i.e., decryption) should not occur before payment has been successfully made. This means that the client has been debited and the copyright holder credited with the right amount of money. This raises the issue of atomicity and fairness of transactions where either the transaction as a whole is successful or not at all. Work has been done in this direction addressing the issue of fairness in electronic commerce [70].

It should be noted that all these security aspects should not prevent copying, holding and free distribution of the documents since they embody the necessary mechanisms for protecting, paying and releasing their content. Thus, when one finds that a document could be of interest to a friend, it can be directly copied and forwarded to him since any access to the content will generate a new payment to the copyright holder. This forwarding action must not involve the information provider.

Accessing a document content off-line without being tied to a network is an important issue. However it requires the use of tamper resistant external devices such as secure smart card readers for off-line electronic commerce in order to undertake the role of a financial institution. This raises however the problem of correctly dispatching the revenue collected off-line to the right information providers. This issue will need careful attention when considering off-line usage.

Currently revenue collection for electronic material does not support the pay per use scheme. If one is interested in only one section of a publication, it will not be possible to pay only for that specific section but one will have to buy the whole document or issue or pay a full

subscription price in order to be granted access rights. Whereas in the electronic world having every piece of information attached with a price associated to authorized actions (policy) results in a highly flexible *pay per use* scheme.

It is noteworthy to mention that issues considering whether or not information is to be subsidized by advertisement is a policy decision of the information provider. Free information can be considered as paying information with price zero. Free does not mean loosing copyright and intellectual property. Thus these issues need careful attention even in a “free information” digital world trend.

3.2.2 Information Classification

- *information providers should be able to have their own classification schemes.*

Each publication has its own way of structuring and classifying its content. For example, a newspaper has a front page with headlines and is composed of columns, sections (e.g., politics, finance, sports, arts, classified, etc.) and sub-sections (e.g., international, national, local, etc.). Thus, the classification scheme used by an information provider for a given publication can be easily derived from the structure of that publication. From there on, and in the scope on an electronic publishing environment, a content element belonging to a given *classification element* can be assigned a classification identifier by the publisher. This is done at the time the content is published and is similar to assigning the content to an element of the classification scheme. Moreover, due to the electronic nature of the content and the possibility of dynamically processing it, it is possible to assign it alternate classification identifiers together with “weights” reflecting the level of relevance of that particular content to the alternate classifications. For example, a newspaper article can be classified as national politics and also be relevant to social issues, etc.

Publications are different and their structure depends on many parameters such as field or domain, theme, periodicity, target audience, etc. It is important to maintain this diversity of information sources and not to attempt to unify information classification globally. A global unified classification scheme of information would result in a difficult to manage structure requiring centralization, which would in any case end up being an unsatisfactory common denominator. Thus, it is capital that each electronic publication keeps its own classification scheme from which the users will be able to make accurate and fine grained decisions about their interests. There is a problem however when the classification scheme changes due to reorganization of the publication. In the printed media, changes are simply reflected in the new editions together with comments. However, in the electronic world, careful attention needs to be given to this issue in order to ensure synchronization of the classification scheme used by the information consumer with the *official* (i.e., up to date) one of the information provider while still preserving history.

3.2.3 Transition from Print and Web Publishing to Electronic Publishing

- *information providers should be able to easily transform existing electronic material into autonomous electronic documents.*

Today, most if not all publishers use computer assistance in their publishing process. It offers two major advantages. First, as opposed to the printed media, the marginal cost of publishing electronic documents once the initial copy exists, is close to nothing. Additional copies can

be produced in a cost effective way. The second advantage is that almost all the material is already in some sort of digital form. Thus, when a new electronic publishing standard appears, it only takes the time to build the corresponding converters for using it. For example, when the Web appeared, it was relatively easy for the publishing industry to generate html files from the different electronic formats, databases, etc. they already used.

3.2.4 Hypermedia Electronic Publishing

- *information providers should be able to establish historical evolution links between content elements,*
- *information providers should be able to insert links in content elements to reference other material inside and outside their own domain.*

Hypermedia electronic publishing is the result of combining hypertext techniques for information navigation with multimedia components. Hypertext systems offer a navigation mechanism allowing users to reach a target object through the referencing link also called an *anchor*. Thus, an electronic publishing system must offer the information provider means to relate (i.e., link) content elements to one an other for historical evolution and access to archives as well as access to other resources. This issue is closely related to of the information consumer's requirements on information evolution and referencing discussed in section 3.1.7. There are two different types of links. First, the internal link for which the target is bound to the same information provider. Second, the external link for which the target can be anything outside the domain of the information provider (e.g., a World-Wide Web address or a piece of content of another information provider, etc.)

3.2.5 Using Standards for Information Composition and Rendering

- *information providers should rely on a standard widely accepted formatting language for electronic content composition and rendering.*

Information composition and rendering are issues which depend highly on standards. Thus one should always be very careful when addressing such issues to try to use available and wide spread standards and by no means attempt to modify them as standards evolve rapidly. Within an electronic publishing environment composition and rendering are of paramount importance as it represents the visible core business of this industry.

Taking an example in the present context and given currently available technology, the use of de facto standards such as the html formatting language [3] and the http protocol [4] offer many advantages. Both of them have become widely available and used standards. Html allows the use of any html editor for formatting electronic content. This prevents from using or building yet another proprietary standard for composing and rendering information. Moreover, combining a commercially viable electronic publishing environment and the Internet protocols in a transparent way will benefit to the users. Such an approach is a key factor in user acceptance and is likely to survive at least a few evolutions as ascending compatibility issues are often accounted for in such cases.

3.2.6 Editorial Process

- *information providers should be able to reflect the editorial process within the electronic content element.*

Electronic publishing raises a fundamental question concerning the editorial process. Namely, what is an *edition* or an *issue* in the electronic context? In the traditional publishing industry when a chief editor gives his final consent he triggers the publishing process of an edition. This is done on a day by day basis for a daily newspaper. Thus, the editorial process for the printed media is a discreet notion based on periodical issues. Whereas in an electronic publishing environment, one could say that an edition is triggered whenever a new information is made available (i.e., any time). Thus, the editorial process becomes a continuous notion. Although this issue is in essence a policy decision of the information provider, it needs to be taken into account from a chronology point of view.

3.2.7 Policy and Marketing Issues

- *information providers should be able to accommodate and customize policy dependent issues according to their needs.*

An electronic publishing system must offer full flexibility to the information provider to accommodate any policy issue he chooses. Such issues cover but are not limited to brandware, pricing, payment methods, layout, frequent buyer programs, subscription, advertisement, direct marketing etc. Most of these are business models and as such must be left open. For example, the industry is very keen on logos, presentation styles, layouts and special services they offer. They are often protected by copyright and reflect in an unambiguous way a brand, a product or even a life-style. From a marketing point of view, this is a very important issue.

3.2.8 Service Availability

- *information providers should strive to offer high service availability to information consumers.*

High availability of the information providers is also of prime importance. The expectations facing the electronic publishing industry require a twenty four hours per day, seven days a week availability of the system due to the inherent global nature of the used media. Bad servicing over time will lead to a loss of trust from the clients who will consider other information sources offering better quality of service. To this respect, the Web has already contributed to educate companies within this global round the clock market.

3.3 The Notion of Electronic Document and its Distribution Policy

The issues raised so far in the previous two sections, together with their corresponding requirements, were centered around the users of a commercial electronic publishing system. It is important to also consider which of these issues concern the documents themselves and thus have an impact on the structure and general policy of document distribution within a commercial electronic publishing system.

- *electronic documents should be at all time in a secured state, able to check for authoritativeness and untampering of content* thus allowing them to be freely held and distributed. The document content should be in an encrypted state at all times. The cleartext of the document content should never be stored anywhere. The document key must be kept only for the decryption and immediately destroyed. This is bound to successful interpretation and completion of the policy to which the document is attached. Moreover, the document should be able to check dynamically that it really comes from where it claims it comes from and that its content has not been modified in any way. This can be achieved through digital signature techniques.
- *electronic documents should embed within them their usage policy as dynamic behavior* thus allowing them to be self contained and autonomous in the "superdistribution" sense. The usage policy (terms and condition of use) will have to be interpreted as a piece of code (i.e., program) attached to the document in order to release the content to which it is attached when successfully completed. This will allow usage based payment policies to be used while enforcing fair compensation to rights holders and correct usage according to the attached policy. Moreover, it will offer information providers a way to build and tailor their documents according to their specific needs since they are programs.
- *electronic documents should be able to determine at all time their state and location* since the electronic document is a program and is given network awareness, it will be able to determine at all times its state, where it is and eventually request to move on its own behalf or be insttructed to do so.
- *electronic documents should take into account user requirements* such as subsequent content access (provided the policy allows it), anonymity, support for multiple payment systems (open ended approach for commercial partnerships), legal redistribution, off-line usage, etc. These issues have been presented and discussed thoroughly earlier in this chapter when considering the requirements from the user's point of view.
- *electronic documents should be defined in a recursive way* allowing the electronic document to embed any number of electronic documents at an arbitrary level of depth thus reflecting the different levels of added value.

3.4 The Enabling Electronic Infrastructure

Having considered the issues and requirements centered around the electronic document and its distribution policy we now consider the issues and requirements which will impact on the choice of the enabling infrastructure. In the scope of this work we address a problem relying on open networks which are untrusted and insecure by essence. Today, the Internet offers such a network infrastructure which will be assumed. Although it might and is probably not the ultimate stage of evolution in open networks it is nevertheless reasonable to assume that open networks will continue to exist in the future and that they will be used as communication infrastructures to build network oriented systems. It is exactly at this point that our work starts in trying to provide the necessary levels of abstraction to address more complex problems such as commercial electronic publishing having a full range of interdependent requirements as we have shown up to now in this chapter.

Having assumed a first level of global network connectivity is not enough. In fact, this first level of connectivity addresses the issue of hauling raw data through physical medium which is achieved by higher level protocols enabling peer to peer communication. Based on this, a higher level of network abstraction is needed which is oriented towards hauling behavior together with the data to which it is tied. Much work has been done in this field recently for example with mobile agents thus offering a new paradigm for network oriented development.

- *the infrastructure should be network aware and architecture independent* thus offering a common abstraction of interconnected heterogeneous hosts. Network awareness is a central issue when addressing distributed applications requiring built-in network connectivity. Architecture independence becomes a real asset in distributed applications as such applications are likely to span world-wide over a wide variety of hardware architectures. Moreover, it would be an illusion to assume unified hardware platforms throughout the world at least nowadays.

- *the infrastructure should allow to package together and bind raw data to the behavior regulating the use of those data* in a similar way object oriented technology binds data to code in objects. These objects given persistency and network awareness are commonly called mobile objects or agents. Thus allowing to enforce their use according to a well defined interface where the agent is able to mediate its own use as well as communication from / to the outer world.

- *the infrastructure should account for transport level authoritativeness and untampering* ensuring authenticated communication between participating nodes and guaranteeing that what has been received is conform to what has been sent.

- *the infrastructure should offer basic communication patterns* such as one to one or one to many both on a local and a cross infrastructure basis, in synchronous or asynchronous ways. In any system communication among the participating actors is required. The patterns for such communications can be different depending on a number of factors like: whether there is a single or multiple receivers, whether the communication must be synchronous or not and finally whether the communication is local to the infrastructure or involves a remote peer infrastructure. In any case, the infrastructure must mediate such communications in order to enforce security.

- *the infrastructure should offer hooks to basic services and allow to build higher level services.* Basic services are services provided by the infrastructure for its management and the services it offers. For example mediated device access to local resources such as printer, storage, user interface, smart-card readers, etc. Higher level services are services built on top of the infrastructure which use the basic services to accomplish their tasks. For example an electronic publishing environment.

Chapter Four

Hep - The Hypermedia Electronic Publishing Framework

Based on the requirements identified in the previous chapter, we designed *Hep*[1] (*H*ypermedia *E*lectronic *P*ublishing), a framework for the commercial distribution of electronic documents over open networks such as the Internet. Our approach is based on the so called mobile object or agent paradigm as presented in the background of chapter two.

In this chapter, we provide the foundation of the Hypermedia Electronic Publishing (Hep) framework. In the first section, we discuss the use of the mobile agent paradigm as supporting technology for our framework. In the second section we present a model for the commercial distribution of copyrighted electronic documents. Finally, in the remaining sections we describe the core environment for the commercial exchange of copyrighted electronic documents as the resulting Hep framework.

4.1 The use of Agent Technology

In the scope of our work we have chosen to use the mobile agent paradigm as supporting technology to address the issue of commercial electronic publishing over open networks such as the Internet. The use of agent technology for an electronic publishing system is a choice. However, there are a number of issues relevant to the electronic publishing field for which such a technology can provide solutions or at least the necessary abstraction enabling its design in a very nice way. This section will focus on explaining the reasons of this choice and set forth the basic requirements towards the design of the target framework based on agent technology.

4.1.1 Network Abstraction, Portability and Architecture Independence

From a network point of view, an electronic publishing system is likely to span over a great number of interconnected heterogeneous hosts reflecting market actors in a competitive global market. To this respect, agent technology provides a very clear and clean abstraction of distributed heterogeneous systems by running on each participating node of the network a common agent execution environment also called an *agent execution platform*. Thus, all the collaborating agent execution platforms together form the agent system as a whole sharing common characteristics. This is similar to the Java [74] approach which has proven to be a very successful platform independent language [71] [72] [73] by the means of its Java virtual machine [75] that can execute

1. From Webster dictionary: ***Hep***: 1- *characterized by a keen informed awareness of or interest in the newest developments*, 2- *In the know, having good taste.*

any Java byte code in a platform independent way. Thus, we can reasonably anticipate and make the assumption that at some point in time agent execution platforms will be generalized and readily available on a cross platform basis (built-in).

From the point of view of the underlying technology over which an agent system relies, issues such as architecture independence (i.e., portability) and network awareness are of prime importance. Java meets these two requirements: first, the Java language has become a de facto standard for network aware software development. Second, the generated Java byte code can be interpreted on any Java Virtual Machine and is thus fully portable due to its wide availability. The Java language per se does not support mobile computations in the sense of an agent system. However, it may be extended to support the agent paradigm as in Odyssey[56], Aglets [59] [60], Voyager [61], Mole [62] [63], etc. Thus, we make the basic assumption that a Java virtual machine exists, or can be easily downloaded and installed. This assumption is also supported by the recent release of the JavaOS [76] operating environment, the JavaStation [77] and the picoJava [78] microprocessor.

4.1.2 Security

Security is a crucial issue in the scope of an electronic publishing system since the target is to distribute commercially the electronic information and thus to protect as much as possible intellectual property, copyright of content and revenue collection. From the point of view of the agent execution platform, special attention must be given to three aspects. First, authentication of the communicating parties. In an agent environment, this means that upon migration of an agent, the target platforms can be guarantied that the incoming agent really comes from the announced source. Second, the platform must be assured that the received agent has not been tampered with while migrating. Third, a minimal level of security is required to prevent malicious agents to enter or at least to execute on the agent platform. This involves some way of detecting such agents and defining what “malicious actions” cover and what to do with them.

Most of the issues regarding security and mobile object systems are described in detail in [64]. They are classified in five categories: transfer security, authentication and authorization, host system security, computational environment security and mobile object system security. However, from the point of view of a user of agent technology (i.e., agent programmer), this is not sufficient. Granting access and execution rights to a foreign agent requires more than knowing that the agent is authenticated and not malicious. For example, in an electronic publishing system, it is important for the user not to be bothered by unwanted solicitations and “attention getters” even if they are not malicious and come from identified sources. Thus, the platform must provide a *hook* to a second level of security which can be customized according to the needs of the application and the user (e.g., an access control system). Such a hook appears to be fundamental in the scope of electronic publishing system in order to prevent flooding of unwanted information, advertisement agents and other solicitations. Similarly, after considering incoming agents, attention should be given to outgoing agents. Since the agent environment will be hosting foreign agents, it is capital to constrain any migration or communication request to be granted clearance for leaving or authorization to communicate with the outer world. Migration

and communication should not be possible without the information consumer being explicitly aware of it or at least it should comply with general terms the user has agreed upon.

From a content point of view in a commercial environment where security, copyright, intellectual property, authoritativeness, untampering and other open and flexible business policies are of prime importance, enforcing these issues is mandatory. By encapsulating within an agent both crypted content and the necessary behavior (i.e., executable code of the agent implementing access policies) to release this content seems to be a promising direction. Thus, such agents become self contained, autonomous and secure. As a result, such agents can be freely copied and distributed at will (in the superdistribution sense) without infringing any copyright or intellectual property law. This is due to the fact that the agent is responsible for its own security. Content access will only be granted upon successful payment or presentation of valid proof of purchase. In other words only if the access policy bound to the content was able to be met at runtime by the agent.

4.1.3 Persistency

Persistency is an important issue in the field of mobile agents [79]. Mostly for two reasons: first because data usually outlives the programs that created them and second, because agents are intended to carry not only behavior but also state that must outlive program execution. To illustrate this point and as an example, one could think of a “shopping agent”. In the scope of an electronic publishing system, we consider this issue from two viewpoints: the agent platform and the agents themselves. From the platform point of view, it should be possible to freeze the agent environment as a whole any time (this also involves the freezing of the currently executing threads) and to restore it either on the same machine or after migration on a new host, resuming execution at the exact previously frozen state. This could be found useful in case of *desktop to laptop or palmtop* transfer for mobile users.

Concerning the agent instances, it should be possible to save agents to secondary memory and reload them subsequently. Since the agents are executing, two cases arise. In the first case, the currently executing threads of the agents must be frozen immediately and the agent saved. In the second case, the agent is instructed to complete current execution of critical sections prior to the freezing operation. This type of operation requires a way to mark the critical sections of an agent. The freezing operation on an agent should also be possible without saving it to secondary storage. This is, putting it in idle state until some condition is verified. This could be very useful for scheduling, synchronizing purposes and could also be useful to save system resources.

4.1.4 Migration

An electronic publishing system requires that the participating actors exchange payment against the provided service (i.e., the content, document, etc.). It is also most likely that such an environment will evolve over time, requiring to update software. Mobility and flexibility of persons have become common constraints of every day life. Therefore moving around at such a fast pace require means for freeing computer environments from being tied to specific hosts or geographical locations by allowing them to be easily transferred from host to host without losing any state while migrating. The issues raised here cover software upgrade, mobility of both the running

software and environment while preserving code and state[1] during migration. Agent technology should accounts for these issues together with the inherent security and consistency issues. However, it has appeared that for most applications for which agents have been considered weak mobility would suffice (i.e., no state preservation of currently executing threads). As a result, an agent is guarantied to always be started the same way thus able to determine its context in a consistent way and its condition of use.

The agent system should offer means for sending agents (i.e., migration) on either a one to one, or a one to many basis. In both cases it should be possible to specify whether a response is expected or not. When sending agents on a one to many basis, it should also be possible to specify that the agent is to be issued (i.e., copied) as many times as the number of destinations or that it should have a unique copy travelling from destination to destination until they have all been reached. When an agent migrates, many things can happen. The issue here is to provide the mechanism to instruct an agent what to do in situations such as unreachable destination, denied access, normal or abnormal termination, security violation, etc. For example, to instruct the agent what to do upon completion when it has migrated, a final action could be self destruct, return to original location, stay and sleep, execute, save, etc.

4.1.5 Practical Issues for an Agent System

Additionally, some other more practical issues should be considered and addressed in the scope of an agent environment. These relate to ease of installation and basic automatic configuration, recursive definition of an agent and other “housekeeping” aspects which are discussed below.

Ideally, the agent system should be totally transparent to the user especially in the scope of an electronic newspaper system which is aimed at non computer experts. The platform should be able to install itself almost automatically from scratch and remove itself likewise. Thus, a new user could download the environment through a plain web browser and simply launch it to use the electronic publishing system without even noticing the existence of an agent environment.

Concerning the platform configuration at start up time, it should be able, upon request, to automatically configure, in a kind of *plug’n play* way, most of the system resources like printers, databases, etc. Different levels of automatic configuration could be introduced depending on the level of the user (e.g., fully automatic, semi-automatic, personalized for experts). The reason is that no matter where a system is, such system resources are most likely to exist and to be needed by the user and the system. This way, when a platform is launched at a given location, those common resources can be automatically found and configured to meet the users needs.

There is an important issue concerning agent embedding and duplication. For example, one receives an electronic document agent and decides to make comments about its content. The issue is how to achieve this from an agent point of view. The ideal situation would be to create a new container agent holding the comments and embedding the initial untouched document agent. Conceptually, this is a very clean approach in the sense that it reflects the different levels of added value. Thus, for example, an information provider offering a news digest service can easily compose an agent on a given topic, holding information agents from other providers re-

1. This includes preserving the state of the currently executing threads also called strong mobility

lated to the given topic. The added value of the digest service is reflected by the containing agent which can have a price while preserving the embedded information agents of other providers. This approach raises however the issue on how an agent is to be duplicated. In the first example of an electronic document and its comment, there are three ways one might want to forward the information: with the comment, without the comment or only the comment with a reference to the electronic document.

Finally, there are a number of “housekeeping” issues. Having an environment hosting and executing software agents requires means to monitor them. For example, retrieving the state of a foreign agent which is faulty requires to remove it. An agent system must offer to the runtime environment the means to create dynamically new agents and to tailor them according to the needs of a running application. For example an electronic publishing system must be able to create a document agent on the fly with specific properties. Finally from the point of view of the programmer of an agent based system a clear interface (API) to the agent system is essential.

4.1.6 The Notion of Service within an Agent Environment

Having set forth the need for an infrastructure enabling commercial electronic publishing is necessary but not sufficient. We now need to introduce the notion of service within an agent environment. This notion of service is motivated by the fact that agents are pieces of software coming from an outside untrusted world. Software usually interacts with different devices such as the screen, the file system, maybe a database, the printer, other agents, etc. But since these agents are untrusted it would neither be reasonable nor desirable to grant them full access to what ever they want.

For this reason we need to provide an abstraction of such services within the agent environment in a way that will enforce proper usage of these resources in a controlled way. For example, an agent that needs ten kilobytes of disk space would issue a request to a “storage agent”. An newly arrived agent could need a window to ask a question to the user. Furthermore, this window could be associated with different rights depending on whether the agents needs to do input, output or both. As a result, one can reasonably anticipate the need for services such as access control, storage management, user interface, electronic commerce, foreign agent management, agent “janitoring”, etc. To illustrate this, one could think of it in an operating system - program way where the OS would be an “agent based electronic document / publishing OS” over which electronic document / publishing applications would run.

Such services will be discussed and described as part of the targeted framework later in this chapter when describing the core environment for commercial electronic publishing over open networks.

4.2 Electronic Document Distribution Model

In the design of our framework we have made the choice of packaging into an agent each document together with the program that controls access to its content. Each document is thus a separate agent that is responsible for its own security covering tasks such as:

- Release free information regarding the document (e.g., title, author, price, abstract etc.).

- Communicate with a local billing agent for charging and billing. The purpose of the billing agent is to provide a single point of contact for all the other agents in the system that need to charge the user account.
- Control access to the content. The content is normally encrypted to prevent unauthorized access and tampering. The agent will handle the decryption in cooperation with the billing agent.
- Possibly establish contact and communicate with a local representative of its creator.
- To be network aware and therefore be able to determine at all time its current, previous and creation location.

The main problem that the design addresses is to prevent the user from bypassing the agent security mechanisms and accessing the data component (i.e., the content) of the agent directly. If such attempt is successful then both the charging mechanism and the copyright protection will have been rendered inoperative. It should be noted that what we strive for is a system that will make it expensive to break the protection mechanism of an agent. The biggest risk, however, is generic failures. These would allow the user to devote significant effort to find a generic weakness in the system that would allow access to all the agents with small incremental effort.

Another consideration is the cost to the information provider if the system is compromised and the security procedures have to be changed. By encapsulating the security mechanism inside the agent, the information provider can alter the security policy easily without affecting the agents already deployed. Thus, compatibility constraints with the older versions will not be an issue. Since the user environment deals with the agents and does not take into account the way these agents work internally, multiple versions of the agents can be resident on the same machine without mutual interference.

4.2.1 Model Overview

Given those considerations we have opted for a modular architecture which is displayed in Figure 4.1. A key actor in this transaction is the credit institution which may be a bank, a credit card company, a telecom operator etc. The purpose of this institution is to act as a trusted party between the information provider (seller) and the information consumer (client). Both seller and client trust the credit institution to authorize the unlocking of a document in exchange of payment that is transferred from the information consumer’s account to the account of the information provider. This scheme is based on public key encryption [43][44].

4.2.2 Packaging

Under this scheme, document M is encrypted using conventional symmetric encryption and the key k is placed in the agent. To avoid the risk that the user may discover the key by searching through the code of the agent, we encrypt k using the public key of the credit institution (C), producing $E_C(k)$. A document identification string (DIS) containing additional information such as the cost of the document, the beneficiary of the transaction, document information for auditing and other statistical data, is appended to and signed by the information provider (using the provider's private key P). Either the signature of the DIS or the billing information (DIS) is also en-

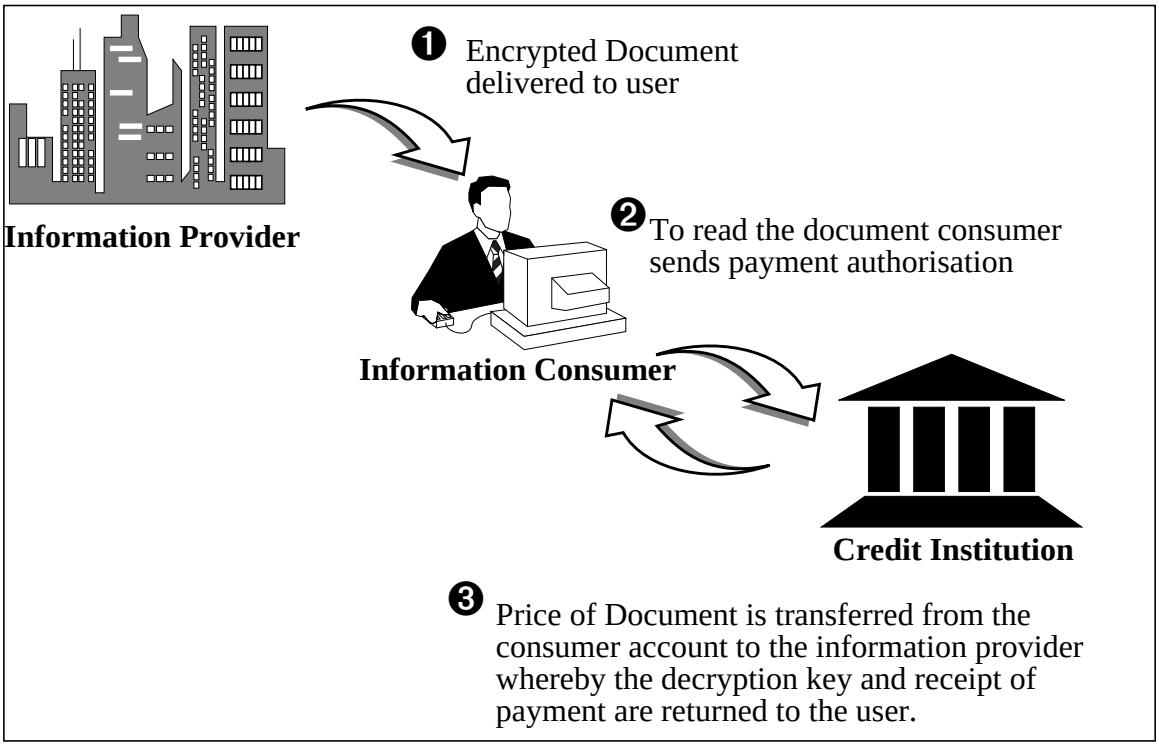


Figure 4.1 Commercial Document Distribution Model Overview

crypted with the public key of the credit institution in order to prevent information provider masquerade attacks, producing $E_C(I)$. The result, $S_P(E_C(k, I), \text{DIS})$, is then stored in the agent and the agent ready for shipping and superdistribution. In addition, we also include a signature of the agent's code (AC) implementing the operations and policies for accessing the content along with a signature of the encrypted document (BH). The result, $S_P(AC, BH)$ is also included in the agent. This last step allowing to bind the agent code, the document an the provider securely.

4.2.3 Accessing the Content

When the user wishes to unlock the document the agent will contact the credit institution and receive a special session key T encrypted with the user's public key ($E_U(T)$). The user will be able to decrypt $E_U(T)$, acquiring access to the key that will be used for the rest of the communication with the credit institution. $S_P(E_C(k, I), \text{DIS})$ will be signed with the private key of the user (U) and the result is encrypted with the session key and sent directly to the credit institution.

The credit institution uses the session key to decrypt the message:

$$D_T(E_T(S_U(S_P(E_C(k, I), \text{DIS})))) = S_U(S_P(E_C(k, I), \text{DIS}))$$

This will be validated using the user public key, giving:

$$V_U(S_U(S_P(E_C(k, I), \text{DIS})))) = S_P(E_C(k, I), \text{DIS})$$

From there we verify the signature of the information provider:

$$V_P(S_P(E_C(k, I), \text{DIS}))) = E_C(k, I), \text{DIS}$$

Then, the credit institution uses its private key to decrypt k and I:

$$D_C(E_C(k, I)) = k, I$$

Further processing can only be done if either of the following conditions is true:

I is the signature of the DIS	I is the DIS itself
if $I = S_P(\text{DIS})$	if $I = \text{DIS}$

Since these information have been verified against the public key of the information provider and either of the above two conditions are verified we can be sure that they are valid. We now have the decrypted key (k) and the document identification string (DIS).

At this stage an amount equal to the price of the document will be transferred from the user account to the account of the beneficiary. Then the document key k will be immediately encrypted using the session key that was generated for the client, producing $E_T(k)$. In addition the credit institution will construct a receipt for that document so that the user will be able to have that document unlocked in the future.

The agent will receive $E_T(k)$ and decrypt it with the session key to get k. At this stage the agent will be able to decrypt the document and display it on the screen. The basic mechanism can be used to implement different charging policies so that the user be offered a price range depending on whether the article will be viewed, printed, or whether audio or video components should also be unlocked and so on.

We mentioned earlier that the credit institution will hand a receipt as well as $E_T(k)$. This is so that the agent does not need to arrange for long-term safe keeping of the decryption key. If the agent is terminated (e.g. to make room for new material) the user can still retrieve the document and give the old receipt to the new agent and this should be sufficient for the document to be again unlocked provided the pricing policy allows for multiple content access.

To prevent these receipts from being copied by users and thus creating a back door to the agent payment system, we still require the receipt to be validated by a credit institution. Although the user, in principle, is responsible for the safe keeping of those receipts, by putting the credit institution in the loop we allow those receipts to be tightly tied to the user who purchased them. Under the scheme described below, only persons with access to the private key of the user will be able to use the receipt. In this case, the repeated use of the same receipt will alert the credit institution that a user account may have been compromised.

The receipt includes the identification of the document, the components that have been paid for, identification of the user who purchased the article, time stamps etc. This information is signed by the issuing credit institution and need not contain any hidden, or encrypted information.

4.2.4 Subsequent Content Access

The user wishing to unlock a paid document will hand the agent the receipt. The agent will append the receipt (r) to the $S_P(E_C(k, I), \text{DIS})$ packet described above. This will be signed and encrypted as before and the resulting $E_T(S_U(S_P(E_C(k, I), \text{DIS}), r))$ message sent to the credit institution. Note that if the various credit institutions have exchanged their public keys, the credit institution that processes the receipt need not be the one that issued it. Thus a user switching banks

for example will still be able to use the receipts charged to the old account, provided that we have a unique identification scheme for the users. If this is not possible the user may arrange for the old credit institution to inform the new one of the old account number.

The validity of the receipt is checked against the following elements:

- The signature of the issuing credit institution validates the contents of the receipt.
- The request $E_T(S_U(S_P(E_C(k, I), \text{DIS}), r))$ is signed by the user, so that we can compare the user identification code attached to the public key held by the credit institution with the user identification code embedded in the receipt.

If the two conditions are met we can be reasonably sure that the new request is valid and the credit institution will return just $E_T(k)$, since this time a receipt should not be issued.

4.2.5 Off-line Operation

In the case where a user would like to be able to view the documents off-line, the role of the credit institution must be delegated to a proxy. This proxy can be a smart card or a PCMCIA card that can perform public key encryption internally [39][40]. For example, the CAFE project [36] and the Mondex system [42] use the smart card technology for off-line transactions. The basic assumption is that the card should contain, but under no circumstances reveal, two private keys. The first will be the private key of the user and the other will be the private key of the credit institution.

Needless to say that if the credit institution key is compromised then the entire system of distribution will be compromised. Thus it is very important to ensure that the smart card technology chosen for this scheme ensures the non-disclosure of its contents.

The card should offer the following services:

- **Credit:** It will be “charged” with money from the user account at the credit institution during an on-line connection with this service. As documents are accessed while the user is off-line, charge is deducted from the card.
- **Document unlocking:** Since the card will have the private key of the credit institution it will be able to unlock documents using a scheme similar to the one used for the on-line transaction.
- **Log of charged items:** Since the charging will be performed off-line a record must be kept of the prices of the documents accessed and the beneficiaries of these transactions. During the next “recharge” of the card this information will be uploaded to the credit institution and will be credited to the correct accounts. The problem of a user that does not recharge his card can arise but could be solved for example by having a deposit on the card i.e., if the user stops using the card he will have an incentive to return the card to get back the deposit. (By return, we may mean that the user simply sticks the card into an ATM for example).

Current technical limitations of the smart card such as memory capacity may become a serious problem as the log of charged items may not fit in the card. As a result, the user would

suffer the pain of being forced to connect very often in order to settle the transactions (i.e., to free memory, clear the log). One possible solution to this problem could be to have the card hold multiple credit lines, one for each information provider. Thus, credit would simply be transferred from the user account to the credit lines of the selected information providers at the time of the card "charging". This is a debit card / prepaid approach that has the major drawback of linking credit to information providers in advance. The user will then not be able to use credit of one information provider to pay for documents of another while being off-line.

Another solution to this problem could be to have the card hold one slot per information provider and have the card hold money from the credit institution in a "user wallet". In this case the card would hold credit as a purse and the charging would be done by transferring the amount of the transaction from the purse to the balance of the specific information provider's slot on the card. Here again different options could be adopted depending on the number of credit institutions that are to be represented on the card. But in any case, the user will still be required at some point in time to connect back to the network in order to dispatch the off-line collected money to their correct beneficiaries (i.e., settlement).

For the time being most of these issues are bound to the limited storage and processing capabilities offered by smart-cards. However, we are confident that state of the art technology in this field will evolve driven by market demand and the huge commercial potential of applications requiring such technology. The main concern being to provide tamper resistant devices for off-line electronic commerce.

4.2.6 Discussion

The document distribution model as described above requires two network interactions with the credit institution for content access. Namely, a first one using asymmetric cryptography to acquire a session key from the credit institution in order to establish a secure and authenticated communication channel, and a second using symmetric cryptography to do the actual transaction(s) using this session key.

One could argue that this is very inefficient as the information consumer could generate the session key himself and thus eliminate one network interaction with the credit institution:

To do so the user generates the session key T and a random nonce n. Signs both and encrypts the result with the credit institution's public key, producing: $E_C\ (\ S_U\ (\ T, n\ )\ )$. The user can then directly use the session key T as described before to produce : $E_T(S_U(S_P(E_C(k, I), \text{DIS})))$. Finally, the user sends both : $E_C\ (\ S_U\ (\ T, n\ )\ )$ and $E_T(S_U(S_P(E_C(k, I), \text{DIS})))$ to the credit institution which continues as if it had generated the session key after having verified the freshness of the nonce.

The problem with this approach relates to security. Namely, the trust that can be put in leaving the user environment generate session keys (cryptographic algorithm and key length used). This is the reason for which there is this overhead of session key acquisition. The credit institution being a trusted party of both the information consumers and providers is more likely to offer and require an acceptable level of security. However, this could very well be determined at run

time depending on the ability of the consumer to generate secure session keys thus allowing to use either solution.

4.3 The Overall Architecture of the Hep Framework

Based on the choice of using an agent infrastructure and given the document distribution model described in the previous section, the following layered architecture is proposed, assuming an underlying Java virtual machine.

The whole agent based Hep architecture is composed of three layers shown graphically in Figure 4.2. The lowest gray shaded layer is the *agent layer* representing the underlying agent

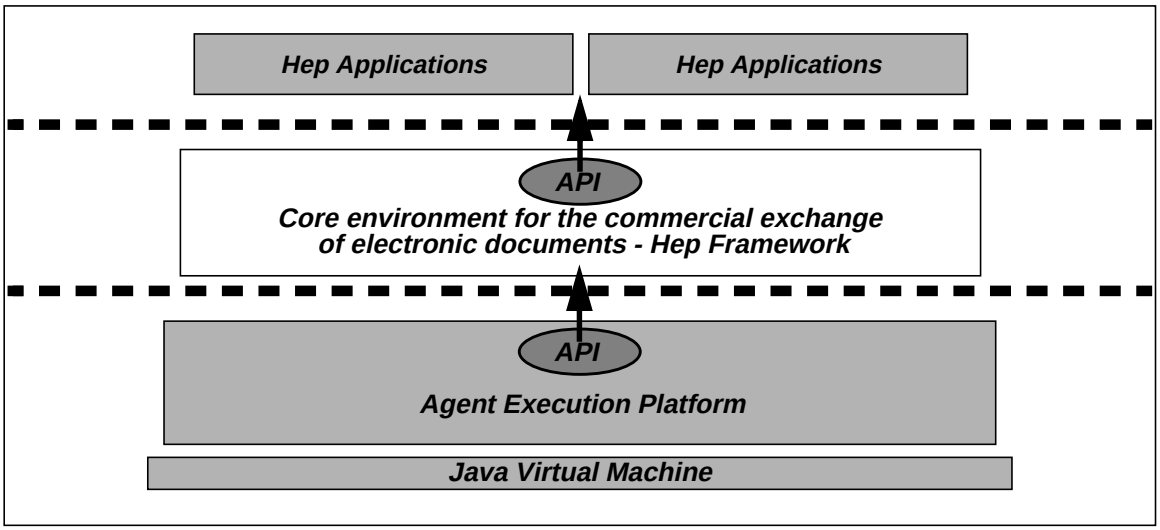


Figure 4.2 Layered representation of the Hep architecture

infrastructure (Java Virtual Machine and agent execution platform) over which the Hep framework relies. The second layer, the *core environment,* is the basic building block of the Hep framework. It enables the safe and secure distribution and exchange of electronic documents (i.e., Hep documents). The top gray shaded layer, the *Hep application layer,* implements the various tools and applications bound to the application specific layer. These tools are the interface to the users and all the application specific logic. This layer holds as many components as the number of Hypermedia Electronic Publishing applications that are to run within the Hep framework. To be noted that we have not distinguished between consumer and provider applications as we anticipate these roles to be often merged: an information consumer can also be an information provider of his own added value. Conversely, an information provider might also be an information consumer. Thus Hep applications could be symmetrical.

The layered approach offers the advantage of a clear separation between the Hep application layer and the Hep core environment enabling the commercial exchange of electronic documents. Such an approach provides a framework for the electronic publishing field offering a common underlying infrastructure to different classes of electronic publishing applications including electronic newspaper systems, digital libraries and other value added electronic publishing services. These can be considered as classes of electronic publishing applications sharing

many common characteristics such as copyright protection, revenue collection and security among the major.

4.4 The Agent Execution Platform

A central notion within an agent system is the notion of *Location.* Locations must be uniquely identifiable and represent the places where agents are created and executed. Agents migrate between locations. All the locations together represent the agent system as a whole, thus enabling the necessary abstraction to represent such a distributed agent system as a whole, spanning across a global open network such as the Internet. The agent environment targets to offer a first level of security in terms of authentication of communicating parties and guaranteeing that the software agents have not been tampered with while migrating. The agent system must offer two types of agents: *user agents* and *system agents*.

User Agents are, from a privilege management point of view, *low privileged* agents. They are granted very limited rights to undertake any action in the system and have no rights to act outside the agent system unless they acquire prior authorization from higher privileged agents. User agents can communicate with other agents, migrate and create new user agents.

System Agents have more privileges than user agents. They are assumed to be permanent well behaved citizens on the agent execution platform. They are intended to be used as an interface to system resources outside the agent system (i.e., database, file system, etc.). These agents must be loaded at platform start-up time, they exist throughout the platform lifetime and they cannot migrate to other platforms.

Mobility in an agent system is achieved by *migration*. It involves the instantiation or the creation of the agent that is to migrate. Then it is instructed to migrate whereby it completes cleanup and final actions prior to the migration process. It is then sent to the destination location where it will be instantiated and started. Issues surrounding the issue on weak and strong mobility fall out of our scope but for many reasons such as security and consistency among the major, week mobility is assumed and considered sufficient in our case.

Inter agent communication in an agent system can be achieved in two different ways: remote procedure call (RPC) and message passing. Remote procedure call is a synchronous communication mechanism which implements a method call to an agent either on a local or a remote location. Message passing can be both synchronous or asynchronous, and allows transfer of messages between agents.

4.5 The Core Environment for the Commercial Exchange of Electronic Documents

The agent execution platform can be decomposed into three main *areas* for holding agents depending on their privileges and roles: the *Hep entrypoint area*, the *Hep restricted area* and the *Hep system area*. These areas are defined in order to handle efficiently the security requirements of the system.

The general idea is that any incoming or outgoing agent must first pass through an *entrypoint area* before it is granted either access rights or clearance for leaving. Once a foreign agent

has been granted access rights, it resides in the *Hep restricted area*. In the *restricted area* an agent is not allowed to undertake any action on his own behalf without having acquired prior authorization from an agent residing on the *Hep system area*. Foreign agents can never reside in the *Hep system area*. Finally, all user interaction and local system resources are managed by agents that reside in the *Hep system area*. Figure 4.3 illustrates the complete Hep architecture.

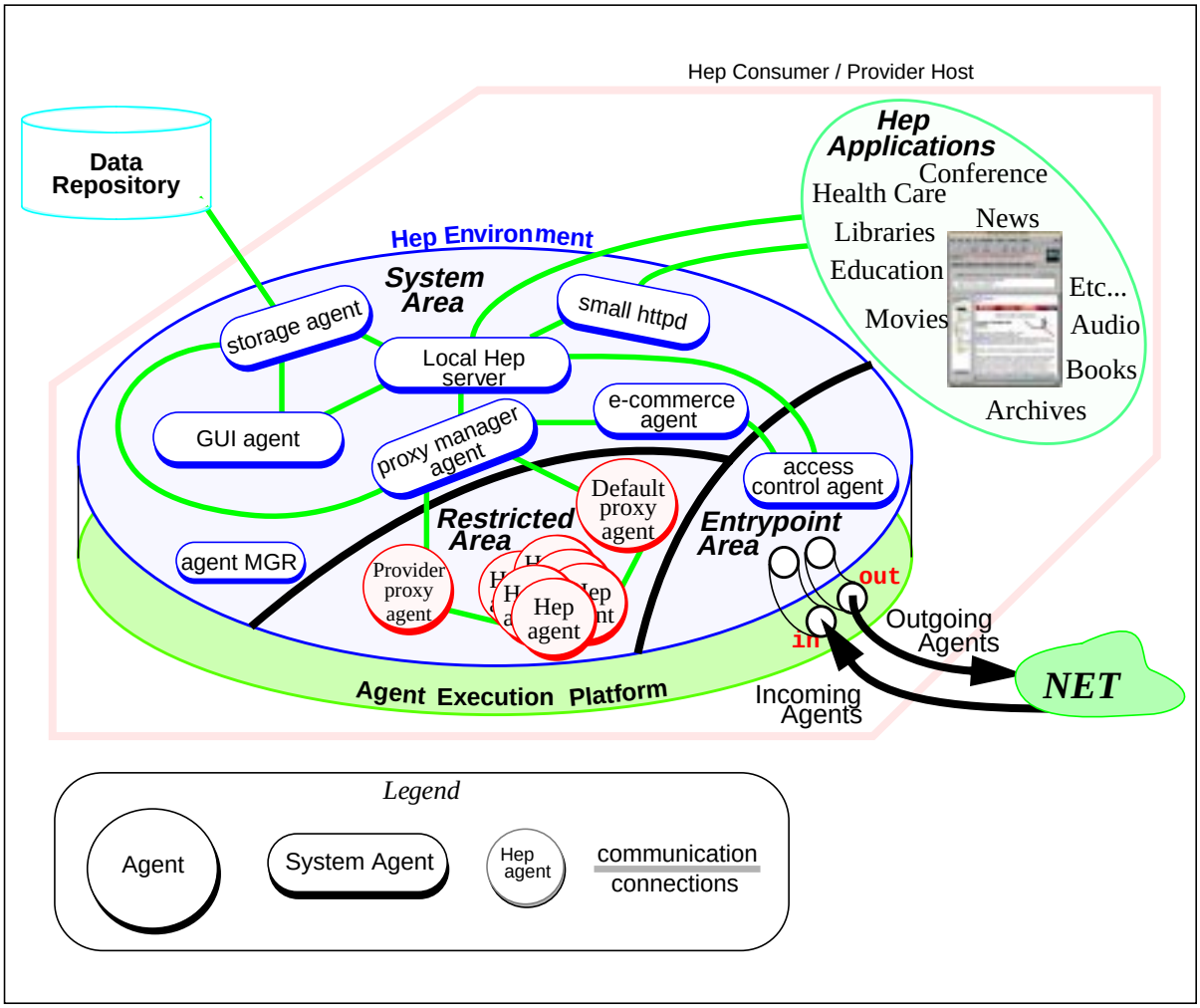


Figure 4.3 The architecture of the system based on agents

Existence of these three areas together with the system agents they hold (including the default proxy) can be assumed and relied upon within the Hep framework. They constitute the Hep framework providing the necessary common infrastructure allowing to build and run Hep applications.

4.5.1 The Entrypoint Area

Although security is taken into account at the agent platform level, another level of security needs to be introduced from the Hep framework point of view to enforce application level security policies. Hep provides mechanisms to master and control agent traffic between the platform and the outer world. This is done by introducing the *Hep entrypoint area*. The role of this area is to be a placeholder of incoming and outgoing agents before they acquire authorization for ei-

ther entering or leaving the platform. In addition, this area holds a unique system agent: the *access-control agent*.

The access-control agent

The *access-control agent* has high privileges on the platform as it is a system agent. It provides at the Hep level the equivalent of a *fire-wall* in order to have total control over incoming and outgoing agents. This is the second level of security ensuring that agents are identified and registered as “allowed to migrate”. Any incoming agent, upon arrival on the platform, has to establish contact with the access-control agent. The access-control agent checks whether the newly arrived agent is authorized to enter according to the local access control scheme. In the event of rejection of authorization the incoming agent can be either destroyed or returned to its sender. The former case would probably be the best solution since sending back a denied agent to its source is an important information which we probably don’t want to give. However if the foreign incoming agent acquires the right to stay it is granted only the privileges corresponding to its role on the system and it is relocated in the Hep restricted area. The Hep system agent (the local Hep server agent) in charge of handling that *arrival* event is notified. Likewise, any outgoing agent must also contact the access-control agent in order to be granted clearance to leave the platform. All the agent traffic can be monitored, traced and logged by the access-control agent.

The access-control agent also takes care of immediately processing key management agents such as requests for session key and public keys. The reason for this is that there is no need to host a foreign agent for such requests. They are thus immediately answered and the agent can leave without further notice.

An agent at any given time belongs to an area of the Hep core environment (as introduced at the beginning of this section) depending on the rights it has been granted. The possible types of area restriction are as follows:

DEADZONE_AREA

This is the default area type assigned to any agent before it is granted any kind of right on the agent platform. This area type is set when the agent is migrating or moved to persistent storage.

ENTRYPOINT_AREA

This area type is assigned to any incoming or outgoing agent before requesting authorization to stay or clearance to leave the agent platform. Given such an area clearance, an agent will only ne allowed to contact the access-control agent.

RESTRICTED_AREA

This area restriction type is granted by the access-control agent to any foreign agent after it has been identified and registered on the agent platform. From this point on, a foreign agent with this area restriction will be able to interact with the system in order to complete the task it was assigned.

SYSTEM_AREA

This area restriction is never granted dynamically by the Hep core environment for security reasons. It is the default area restriction of any system agent except for the access-control agent that resides in the entrypoint area.

4.5.2 The Restricted Area

The *Hep restricted area*, is the placeholder for all the foreign user agents when they have been granted access rights. The user agents have restricted rights and most of the actions they undertake have to be accepted or processed by Hep system agents. A common characteristic of the user agents is that they all reside in the Hep restricted area since they are considered potentially harmful due to the fact that they are foreign agents. There is however one exception to this rule for the default-proxy agent which is not mobile as presented below. We have four user agents: the *information provider proxy agents*, the *proxy-update agent*, the *default-proxy agent* and the *Hep agents* which can be either *document agents* or *service agents*.

The information provider proxy agent

The *information provider proxy agent* acts as a local representative of the information provider on the user's host. It handles all the interactions between the local consumer and the remote provider to which it belongs. It also handles the interaction with the local Hep documents of the provider it represents. It is optionally installed when a user requests information for the first time from an information provider. However its existence is not mandatory and when the proxy agent of a given information provider is not found, its role is undertaken by the default-proxy agent (presented below). For example when receiving a forwarded article, the corresponding information provider proxy agent might not exist locally. The user can request an information provider proxy agent at a later time. This agent stays on the user's platform and it can be updated by the information provider who owns it, through a *proxy-update agent* described below. It communicates with the Hep proxy-manager system agent and the article agents it owns (i.e., belonging to the same information provider) to retrieve article information (i.e., price, title, author, etc.), article crypted key or content. Moreover, this agent is expected to implement the information provider's specific behaviors to achieve fine grained policy dependent issues such as special pricing policies, frequent buyer advantages, value added services, etc.

The proxy-update agent

The *proxy-update agent* is used by information providers to update their proxy agents residing on the hosts of information consumers. For example, when a new release of the information provider proxy agent is issued, the proxy-update agent is used to undertake the upgrade process.

The default-proxy agent

The *default-proxy agent* is the *universal* information provider proxy agent. It replaces any unavailable information provider proxy agent on the information consumer's system for interacting with "orphan" document agents (i.e., agents that do not have local representatives of their sources). For example, when receiving a forwarded document, the issuing information provider of that document can be unknown. In such a case, the default proxy agent will be able to under-

take the role of the missing proxy agent to handle and interact with the *orphan* document agent. Although this agent is a user agent, it is neither foreign nor mobile and is thus not allowed to migrate. There can only be one instance of this agent in the Hep environment. It is part of the distribution and is started at platform launch time.

The Hep agent

The *Hep agent* can be of two different types. Namely: it can be either a *Hep service agent* or a *Hep document agent*.

The *Hep service agent* is what can be considered as a stationery agent and is used both at the Hep level and at the application level for inter-platform communication and interaction on a service basis. Such services include for example key management, billing, usage and metering, service request, hauling, etc. Hep service agents can be created by any agent but their migration and communication rights are bound to be checked at the Hep level. Such agents can have different traveling semantics like for example being one way agents that will never come back or for which no acknowledgment of any kind is needed, agents that are expected to come back (return agents) and multiple destination agents expected to travel according to a destination list. This agent is further decomposed in two subtypes:

- the *Billing Agent* used for all commerce transactions such as *withdraw*, *payment* and *receipts*.
- the *Hep Request Agent* used as a general purpose request-reply transaction agent such as *public key request*, *session key request*, *document request*, *information request* and *fragmented information* replies for large objects, *general request*, etc.

In the case of a *Hep document agent* it is the actual document wrapper. It is composed of two elements: the document content (i.e., the *raw material)* and the code defining the behavior to access this content. The document is encapsulated in an agent both for security reasons and for distribution convenience. When the document agent is unable to find its local information provider proxy agent, the default proxy agent takes care of interacting with that *orphan* Hep document agent. Document agents, can be freely copied, forwarded and held since content access is bound to successful completion of the access policy (i.e., payment). Furthermore, this agent can be of three different types:

- a plain Hep document agent,
- a Hep container agent for a set of documents,
- both a Hep document and a container agent thus allowing for multiple levels of added value.

4.5.3 The System Area

The *Hep system area* holds the remaining Hep system agents. Namely, the *agent-manager agent*, the *proxy-manager agent*, the *electronic-commerce agent*, the *storage agent*, the *local Hep server agent*, the *Hep GUI application agent* and the *small-httpd agent*. The characteristic of agents belonging to this area is that they are system agents and as such they have access to system resources outside the platform and all the resources on the platform. Moreover, due to

their nature (i.e., system agents) they can not migrate. The system agents that reside in this area represent the core elements of the Hep framework.

The agent-manager agent

The *agent-manager agent* is responsible for the management of all the agents of the agent execution platform. It is through this agent that other agents are either loaded from or saved to secondary storage, created, started, stopped, destroyed, etc. It helps monitor and control the agents that reside on the platform as well as the platform itself. This agent can be contacted by any agent. Depending on the type of the contacting agent, different actions are available. For example, a Hep system agent could request that a particular user agent be terminated or a user agent might want to retrieve information concerning its own execution or which agent provides a given service.

It is the first system agent to be loaded on the platform and its initial task is to make sure that the Hep system is correctly loaded and started together with the Hep applications. It does so by managing a catalog referencing all the agents of the platform. Each agent that is loaded must register itself for services it can provide. Similarly, agents must unregistered services they provide before being removed from the platform. To do so, each system agent implements the following two methods goUp() and shutdown() which are called respectively upon platform start-up and removal.

Throughout the existence of the agent execution platform, the agent-manager is also in charge of what we have called a "*janitor*" service. The janitor is responsible for checking, at regular time intervals, that all the agents evolving on the platform are not idle (i.e., doing something useful). This is inherent to the agent paradigm where an agent platform holds agents in a similar way a community has members. A member or agent in our case can be apparently idle but it is actually waiting for an event. A user agent must at all time maintain an "activity state" information about whether it is active (i.e., doing something or waiting for some event to occur) or not. In case the user agent is idle and has no further activity, the janitor will ask the agent to remove itself from the platform. If this is not successful, the janitor will then simply kill the reluctant user agent without further notice. This scheme has the advantage of keeping the number of simultaneous active agents on the platform (i.e., load) as low as possible. Theoretically, such situations should not occur because each user agent is associated with an idle activity timer. Thus, if a time-out occurs for a user agent and it is idle, it will remove itself from the platform after completing some necessary cleanup actions.

The proxy-manager agent

The *proxy-manager agent* is responsible for managing the different information provider proxy agents. It also determines the right proxy to contact for relaying requests it receives. If a proxy of a given information provider cannot be found locally the request is passed to the Default Proxy which is the general purpose proxy able to handle the requests.

Moreover, it is responsible for the control of all the actions that a foreign agent wants to undertake. For example, the arrival of a new or an update *information provider proxy agent* and its registration on the Hep platform, a request for payment to open and decrypt the content of an

document agent, the local storage and retrieval of Hep document agents, the request for creating a user agent, any migration or communication request among other possible actions.

The electronic-commerce agent

The *electronic-commerce agent* is responsible for handling payments and other monetary transactions. It provides a unified interface to electronic payments independent of the underlying payment scheme used. It is intended to be used as a general purpose agent responsible for implementing the methods required in order to accommodate the commercial document distribution scheme discussed earlier in this chapter. It can be contacted either by the *proxy-manager agent* on behalf of the *proxy-agent*, or by the *local-Hep-server agent* for other commercial transactions (e.g., transaction logs, user account management, etc.). This agent is the only one authorized to create an outgoing billing agent. To do so it communicates with the *access-control agent* before the billing agent is granted clearance to migrate.

It is anticipated to use the electronic-commerce agent as a place holder for wide spread electronic-commerce systems and protocols. It was not the purpose of this work to devise yet another (micro)payment protocol or system but rather to be able to accommodate currently available payment systems be they cash, credit or debit type. Ultimately, the user is expected to use different commercial electronic payment systems with the financial institutions he collaborates with. It would be unreasonable to force the user into proprietary business models. Rather the user should be left to choose and express his own preferences according to personal needs and situations through an open architecture (electronic commerce framework).

In the scope of the prototype, a virtual cash-like implementation is provided where users are expected to withdraw amounts from different information providers or credit institutions. This cash can be spent anywhere among the Hep information providers. When a user requests to access a document, the required amount of the transaction is “reserved” from the user’s electronic wallet until the transaction is successfully completed. At this point, the user holds a transaction receipt and the document content is displayed. The transaction is committed and the user can see he has been debited with the corresponding amount in his wallet.

The storage agent

The *storage agent* is responsible for all the actions requiring access to a storage system. It can be a database when available, a plain file system or even a caching system for an electronic publishing application. This agent provides a unified general purpose interface to data management. It can be contacted by any Hep system agent. If a user agent needs to access the storage system, it must do it through a system agent in order to acquire the right to do so.

This agent implements methods to read or write a file from / to the file system, and to manage a database account to execute database queries.

The local-Hep-server agent

The *local-Hep-server agent* is the core element of the Hep framework. It almost stands in the middle of everything that happens over the Hep platform. This agent is always contacted by the access-control agent upon arrival of any Hep related agents on the platform. It is the third level

of security in the acceptance of foreign agents on the platform. This level ensures that the incoming agent corresponds to either a pending request or an identified and allowed action (e.g., information provider proxy-update agent). This agent also processes when possible and dispatches if necessary, all the Hep requests issued through the Hep reader (i.e., the Web browser) or any other Hep GUI application agent residing on the platform.

This agent serves as an intermediary between the Hep core environment and the different Hep application layers (one for each Hep based application). The idea being is that it is “peered” at the application layer to be further refined according to the application specific needs.

The GUI agent

The *GUI agent* implements all the GUI aspects of Hep platform except for the Hep reader which is a standard Java enabled Web browser such as Netscape for example. Regarding the reader, Hep only takes care of managing the reader process to launch and stop it. The Hep environment distribution includes a version of Netscape 3.0, but the user has the choice of using his own browser. It is also anticipated to use the HotJava been Web browser in order to enforce consistency and stability between the browser and the Hep environment. Mainly, this agent communicates with the local Hep server agent, the storage agent and with the user directly. Examples of Hep GUI application agents include control panels, profile managers, user information update notifiers, application level data warehouse managers, electronic wallet user interface among others which are described in the next chapter presenting an implementation.

Moreover, this agent is also intended to be used as the general purpose GUI resource allocator for foreign agents. Information provider proxy agents and documents are most likely to need a way to interact with the user for a number of policy dependent issues and reasons. Thus, this agent provides user interface resources on a case by case basis for security reasons. It would not be reasonable to let an agent flood the screen with flashing attention getters and windows without the explicit consent of the user.

The small-httpd agent

The *small-httpd agent* implements a local reduced web server in order to process all the Hep requests issued by the local Hep user using a plain Java enabled Web browser as a Hep reader. It is only contacted by the web browser when Hep links are selected or with specific Hep related requests. Other Web links go directly to the Internet.

This server implements a multi-threaded http server that will only accept http requests from the local host for security reasons. It listens for requests on port 8088 of the local host. The URLs have the form: http://localhost:8088/hypernews-request, where hypernews-request identifies the request to be processed by the http daemon in order to undertake the corresponding action. There are five different request types:

- *Applet commands* used to communicate between Hep applets (e.g., navigation applets for context selector and sections selector) and the Hep environment. These requests are used for example in the implementation described in the next chapter to set the UDP ports, acquire various information such as the contexts and their sections.

- *Hep reader environment requests* used to acquire the applets themselves and related resources such as images, widgets and other Java classes, etc.
- Requests with the ".html" suffix used for *container page requests* holding both the context and the section identifiers.
- Requests with specific application dependent suffixes are used as *Hep document requests* to access the document contents. Such requests hold the document agent identifier as parameter. For example, in the implementation of the hypermedia electronic newspaper application, the ".hna" suffix was used for *HyperNews article requests*.
- Other requests are considered as document embedded elements such as an image for example. A cookie is used to uniquely identify the corresponding document agent to which the request is to be passed.

4.6 The Hep Protocol: Inter-Agent and Inter-Platform Communication

In the Hep framework, we distinguish between two types of communications. Namely, inter-platform and intra-platform communication. Inter-platform communication is to be understood as communication between agents residing on different agent execution platforms (AEP) through an open network such as the Internet. This basically boils down to the notion of communication between interconnected hosts. Intra-platform communication involves communication between distinct "running" agents within the same agent execution platform (AEP). In both cases the platform is to be understood as the Hep environment or layer.

In the case of inter-platform communication, the general rule is that such communication is to be done by agent migration only. As a result direct inter-platform communication is not anticipated for the time being mainly for security reasons. However this could change in the future if necessary and provided the inherent security issues be addressed and enforced. Consequently, inter-platform communication is achieved by creating a new Hep service agent and migrate it to the destination platform where it will communicate on an intra-platform basis as discussed below.

Intra-platform communication is nothing more than inter-agent communication bound to a platform. The general rules for inter-agent communication within a given platform are as follows:

- System agents can always initiate communication with any other agent no matter what type the target agent is.
- User agents cannot initiate communicate with any other agents unless explicitly authorized.

The first rule is justified by the fact that system agents are assumed secure since they are not mobile and their creation is bound to the platform for which they were created and over which they reside permanently. The second rule is grounded by the inherent mobile nature of user agents which are considered untrusted and potentially malicious unless otherwise authorized. Reason for which the default behavior is to prevent communication.

This level of authorization can be expressed from the Hep point of view as a matrix of authorized communications depending on the type and the area restriction of both the agent that wants to initiate communication (i.e., the caller, From) and the target agent (i.e., the callee, To). In this matrix, provided in Appendix C, the corresponding communication right can be read at the intersection of each caller and callee.

To be noted that such communication rights, as described in the table, are to be enforced at Hep level. These can only be overridden at higher levels to restrict them but by no means to widen them. In other words, as you move up the authorization path you are stopped in a non negotiable way by the first denial encountered.

4.7 Discussion

This approach is novel in the sense that it relies on agent technology. By encapsulating documents into agents together with the necessary behavior to protect and release their content we not only gain superdistribution but also mobility of documents and the necessary flexibility required to implement whatever business model (i.e., policy) information providers can think of. By providing behavior at the level of the document itself, it becomes active. The proposed document distribution model offers an open ended approach allowing it to be adapted to any commercial policy a content or added value provider wishes to implement.

The Hep framework as a core infrastructure represents the minimal set of tools and services required and shared by any commercial electronic publishing application over open networks such as the Internet. It provides the means to build upon and develop classes of electronic publishing applications thus providing a clear separation between the application level and the infrastructure that enables it.

Chapter Five

The HyperNews Prototype

The HyperNews[1] (hypermedia electronic newspaper) project [80] [81], part of the MEDIA project [82], aims at developing an electronic newspaper environment based on agent technology. This environment will offer the information providers of newspapers, magazines and alike the means to commercialize electronically the information they hold under similar conditions as the printed versions. It will also offer the information consumers the means to reduce the time spent in retrieving the information for which they have expressed interest from different information providers [83].

The HyperNews users, information consumers and providers, run an agent execution platform on which the different components of the hypernews system execute as software agents. Mobile agents are used to encapsulate the data in hypernews article agents and to update specific software agents bound to the information providers. The resulting article agents can be distributed and duplicated at will since the access to the content is bound to the payment (i.e., superdistribution [6]). This is achieved through a scheme for the commercial distribution of electronic documents that enforces copyright control and payment at the time the information is accessed for reading[84]. Such an environment offers the newspaper and magazine publishing industry the means to commercialize news articles in a way that enforces copyright control and revenue collection. Moreover, it offers the information consumer an interface which is both *vendor independent* and tailorable according to the users needs.

5.1 Overview of the MEDIA Project

The MEDIA project is composed of three sub-projects: ASAP [82] [85], the Agent System Architecture and Platform, KryPict [86], a software environment for copyrighting, authenticating, archiving and retrieving pictorial documents in multimedia databases, and HyperNews, the pilot Hypermedia Electronic Newspaper application.

The ASAP project addresses key issues in the construction of distributed applications over vast and dynamic networks of computers using *mobile software agents*. Mobile software agents embody a new paradigm for the design and implementation of distributed applications. Each agent is a self-contained, autonomous entity, able to move around a network and interact with

1. This project is supported by the Swiss Federal Government with the FNRS SPP-ICS projects MEDIA (5003-045332) and HyperNews (5003-045333)

other agents as well as local services. The goal of ASAP is to deliver an execution platform that supports development of applications based on mobile software agents. This technology will be an infrastructure for developing *commercial information systems* in large scale, dynamic and heterogeneous networks of computers.

KryPict project's main goal is to develop copyright enforcement and document authentication methods that will allow to digitally watermark images. As a result, information providers will be able to make accessible their images over the network without having to fear that these images be stolen and that their copyright be infringed. In a manner similar to what paper mills have been doing for centuries, the idea is to encode a digital signature (digital watermark) within one's own digitized pictorial document; the primary function of this watermark is to unequivocally identify the owner of the picture. The difficulty is here to find a watermarking procedure which will be resistant to a wide variety of treatments. An added requirement could be that the watermark keeps track of some of the processing performed on the document. The two most typical applications of such digital watermarking could be the establishment of the original source of a picture, and the certification that a given picture has not been tampered with or doctored.

The HyperNews project aims in the design and development of an application for the commercial dissemination of information, demonstrating the capabilities of the MEDIA project. The target application is an electronic newspaper in which each news article is "sold" independently, allowing the user to choose and pay only for the articles in which he is interested in. In order to allow the commercialization of news distribution, news articles will not be simple data but rather they will be encapsulated within agents. This way every time the reader wishes to read an article he will have to engage the corresponding agent which will take the necessary actions for honoring the commercial transaction. For example, after checking the reader's authorization or triggering the transfer of some electronic currency from the reader's account to the news publisher's account, the agent will decode the encrypted data (news article) and present it to the user.

The introduction of paid news articles on the network will enhance the service quality offered to the newspaper readers. Today a large number of electronic newspapers exist on the network which however offer a simple electronic version of their printed edition.With a paid electronic newspaper however the reader will be able to receive higher quality, faster and even continuous information updates, personalized services and less annoying advertisement. We anticipate that paid electronic newspapers over the WWW will become the major competition to existing news-feed and news-casting services.

In order to gather experience and gain knowledge of the field, we have implemented in Java a first prototype of the HyperNews system very early in the project. It served as a proof of concept for the initial discussions with our partner the publisher "L'Hebdo"[87]. It also helped identify the issues and the requirements for an electronic newspaper system based on agents [88]. The prototype included a full environment for the information consumer and a very restricted publishing tool for the information provider. A fake electronic wallet was implemented in order to show the commercial aspect of the project. A number of tools were implemented to manage the information interests of the user and the layout of the electronic newspaper. This prototype suffered however from major limitations due to its rapid development and early stage in the project. Namely, no agent infrastructure was used, all the communications were done through

stream sockets and no security scheme was considered. Based on this work, we have designed a framework for the commercial exchange of electronic documents over open networks and implemented a second HyperNews prototype which is described in this chapter. It is based on agent technology and was implemented in Java.

5.2 A Hypermedia Electronic Newspaper System

The pilot application of the MEDIA project is HyperNews a hypermedia electronic newspaper environment. The HyperNews (Hypermedia Electronic Newspaper) is using the MEDIA infrastructure for developing an integrated commercial electronic newspaper system. HyperNews offers the publishers of newspapers, magazines and alike the means to commercialize electronically the information they hold under similar conditions as the printed versions, supporting copyright control and handling revenue collection. In addition, HyperNews, offers the newspaper reader all the advantages of a printed copy issue, like anonymity and free choice of newspaper to read, as well as advanced information access capabilities, like breaking news updates, per article read payment, and event evolution tracing.

The Hypermedia Newspaper has three major components (Figure 5.1):

1. the information consumer side (Hypermedia Electronic Newspaper - HEN)
2. the information provider side (Electronic News Server - ENS)
3. the network linking the two sides

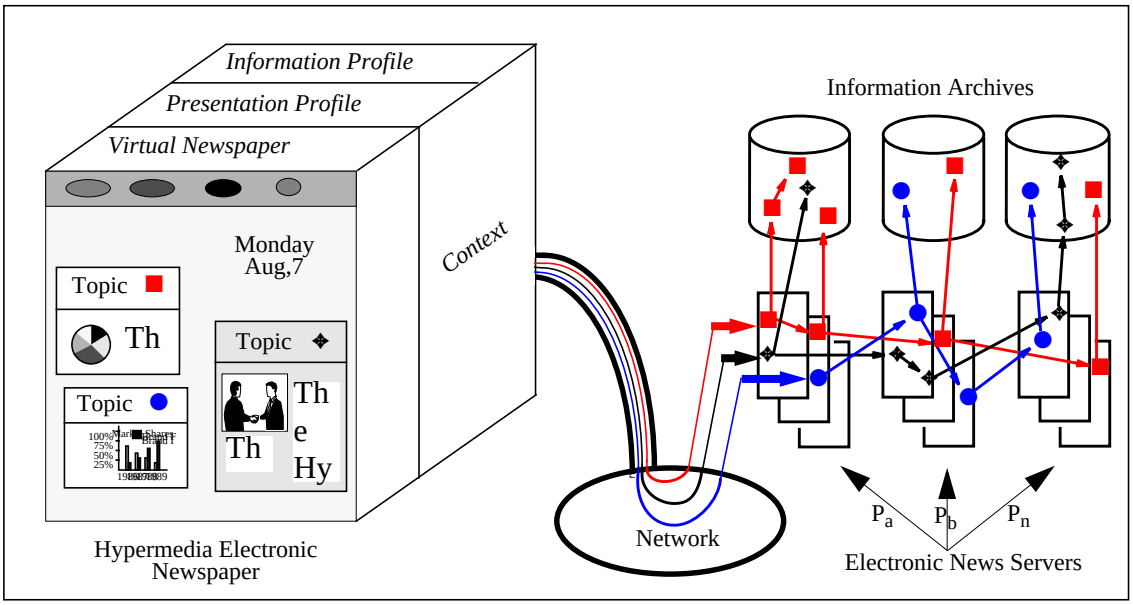


Figure 5.1 Logical view of the Hypermedia Newspaper architecture

On the client side, the HEN presents a personalized virtual newspaper with the relevant information topics retrieved according to the reader's information profile in a layout specified by the reader's presentation profile. Both of these profiles represent a given context for the reader. Each reader can have more than one context, according to his interests. For example while he is reading the newspaper at the office the context will be specified according to professional interests. However, at home the context will be defined to reflect private interests, which can be quite

different from professional interests (e.g. sports, cinema etc.). On the server side, the different information providers are independent from each other. The hyperlinks represent the accurate and up-to-date information generated by the client request. The hyperlinks to the information archives represent the historical evolution of each piece of information if any is available.

The HEN allows the reader to retrieve, read and pay only those news articles that interest him from the different ENS that are available on the network. Related articles from different ENS's can be connected with hyperlinks allowing the reader to easily retrieve and read all information about a specific subject or event. In addition hyperlinks can be available to older articles of the same subject so that the evolution of the event can be easily traced. The ENS classifies the news articles according to the classification scheme of the information provider, allowing the HEN to retrieve the news articles, set up the hyperlinks according to the reader's interests and present the personalized version of the newspaper to the reader.

News articles can be generally characterized by being of low value, having a short life time and of relatively small size.

5.3 Summary of the Requirements for a Commercial Electronic Newspaper Environment

The requirements for a commercial electronic newspaper / magazine can be seen from two points of view: the information consumer's (the reader), and the information provider's (the publisher). For an in-depth description and understanding of these requirements, we refer the reader to chapter 3. In summary, the reader of an electronic newspaper or magazine should be able to:

- choose his information providers and specify his information interests independently for every provider
- define the presentation structure of his electronic-newspaper or magazine
- be notified when information updates are available on issues of interest
- access the historical evolution of an article
- retain full anonymity when reading an article
- pay only for the articles he chooses to read
- hold and distribute articles to other readers without breaking copyright laws
- access, without having to pay again, articles he has already paid for
- publish new articles or commentaries embedding articles of other providers together with his own added value

The information provider (publisher of an electronic magazine, content, added value), on the other hand should be able to:

- produce on demand copies of electronic articles in a cost effective way
- easily transform existing electronic material into self contained electronic articles

- collect revenue from his electronic publishing activity and protect his intellectual property and copyrights against illegal access and use
- accommodate and customize marketing policies and article prices
- offer high service availability to information consumers

5.4 Existing Approaches and Systems

In recent years, electronic news systems have gained interest from the research community. To this respect a special issue on electronic news was published in the International Journal of Information Processing & Management of September 1997 [89]. This issues features a very interesting introduction [90] raising the issues underlying the electronic metaphor of news. Moreover it stresses the unusual breadth of research in this field touching a great number of disciplines. The papers in this issue are categorized in three sections addressing issues related to social aspects, content and presentation.

Many projects have approached the newspaper publishing paradigm from different viewpoints, some of which are briefly presented below. However none address the copyright and intellectual property issue as being a key factor among other differences.

5.4.1 The fishWrap Project

The MIT Media Laboratory has developed an electronic newspaper system called fishWrap [91] [92] [93] used by the MIT community since 1993. It is currently in its release four. This work is very relevant with respect to news customization and profiling issues. It aims at combining individuals personal information needs with the need to be informed on general issues. The fishWrap system manages the users profiles that can be updated by feedback. The system however has a major limitation in that it is a centralized system to which information must be sent before dissemination.

A public version of the fishWrap news service is available and can be tested free of charge currently at The Gate [94] (the Bay Area’s Home Page) which is part of The Chronicle Publishing Company. It includes the San Francisco Chronicle, the San Francisco Examiner and other television and media related companies. They also provide within the fishWrap environment news reports from the Associated Press and other news information sources. It is a password protected Web based system requiring user identification. Usage of the personalized news service requires registration with The Gate. It is currently free of charge. In the registration process, the user is asked explicitly if the company can use or sell the user related information collected. Once registered, it is possible to setup a personal newspaper structure by the means of the Paper Builder allowing the user to setup an arbitrary number of sections in which can be inserted any predefined topics. From there on, the user can then access the personal newspaper according to the defined sections. Based on the user’s “home location” the fishWrap system will also present the user with information related to the home town and country. The system also provides a “Community” page called: Page One featuring articles selected by fishWrap readers. A new edition is generated upon user request by selecting the “New Edition”.

The fishWrap system differs in a number of way from our approach and suffers from major limitations in terms of copyright, payment for content access, customization and profile management, content classification scheme, privacy and anonymity among the major.

- *copyright*: nothing is provided in order to enforce copyright and intellectual property rights protection. The system relies only on traditional copyright law. Copyright notice can be found on every page identifying the copyright owner and the prohibited actions.
- *payment*: nothing is provided for payment of content access. The whole service is free of charge.
- *content distribution model*: the fishWrap system is centralized, it serves as an information "agregator" thus requiring that all participating information providers be registered with the fishWrap system rather than exist on their own providing a service in a distributed way.
- *content classification scheme*: is centralized within a fishWrap service provider. As a result, this scheme is a "one size fits all" classification scheme that might not suite the information providers.
- *profile management*: is centralized within the fishWrap system. As a result, it is not the user that holds the profile that could be used for different fishWrap providers. Thus requiring to define multiple profiles.
- *privacy and anonymity*: are almost not accounted for. Namely, regarding privacy, the user choose whether the fishWrap service provider is allowed or not to disclose or sell information on the user to commercial partners. However, it is not clear to what level a content provider is informed about content readership. Regarding anonymity, it is simply not provided since the user is required to complete a signon process thus revealing his identity.

5.4.2 The HyNoDe Project

The HyNoDe (Hypermedia News On Demand) [95] product is based on an ESPRIT project HyNoDe (number 22160), conducted by Intracom [96] within the Fourth Framework Programme, aiming towards the design and development of a commercial news on demand application. The objective of the HyNoDe project is to design and develop an efficient and market-oriented environment supporting news-on-demand applications. Adapting and integrating the existing IT components needed for news authoring, storage and delivery, the project aims to develop appropriate mechanisms to support news filtering according to the needs of each end-user (the news consumer).

HyNoDe is composed of a set of tools for information providers allowing to compose, link and publish news articles. It also provides for the information consumers profile management facilities to specify their interests according to their needs. It is a password protected Web based system requiring user identification. The HyNoDe approach relies on a hierarchical structure of servers where content providers publish news elements at service provider sites which in turn interact with local service providers before interacting with end-users. As a result user profiles are managed centrally at each level of the hierarchy.

Currently, the trial period is under way with two information providers and is free of charge. However, it is anticipated to make such a service profitable by means of off-line payment systems based on traditional billing services. Such a system suffers from all the limitations and drawbacks presented and discussed for the fishWrap system describes above.

5.4.3 The Electronic News Delivery Project

The Electronic News Delivery Project [97] aims at designing and implementing future news delivery systems. Several prototypes have been implemented within this project. Here again many issues surrounding copyright protection and revenue collections are not addressed. This project has a strong emphasis on layout management, presentation of the electronic news metaphor in the *broadsheet* sense.

5.4.4 Other Approaches to News Publishing

The vast majority of newspapers, magazines and the like publications are already on the Web. Some as on-line supplements to a hard copy edition while others are fully electronic editions. Although they offer different levels of personalization, content selection and filtering, business models and payment strategies, none is able to enforce copyright protection. Furthermore, if the information consumer is interested in a number of different information sources, it will be his responsibility to manage as many user/password accounts, profile definitions and billing systems as the desired number of information providers.

Recently, another approach emerged, offering a new type of interaction to their users with the push model [98] where the user is “fed” with information following a broadcast metaphor. PointCast [99] is a popular example of such services. Free information (supported by advertisement) are broadcast to the user’s screen saver “while having lunch, on the phone or any time the computer is idle”. Although this approach generated unusual hype, it very quickly disappeared due to a number of drawbacks such as the significant network congestion it induced. Moreover such services offer the user limited personalization (i.e., through filters) and predefined channels. The focus is different from a commercially viable electronic newspaper system. It does not offer the information providers means either to protect intellectual property or to generate revenue from this electronic publishing activity. The idea is more on the side of a free (i.e., supported by advertisers) information news feed in a “one size fits all” way, hoping not to fall into a “shove” model.

Finally privacy and anonymity issues are simply not addressed. It is a common practice for information providers to measure, monitor and gathered usage data in order to build profiles on their consumers without them being aware of it in most cases. These profiles represent a tremendous source of information which can then be used for targeted advertisement or even sold to third parties. Such profiles being intellectual property, information consumers are entitled to be protected against their disclosure or at least accept it explicitly eventually against some advantage.

5.5 Enabling Technology

The HyperNews design and implementation started in early 1996 and it was based on the technology state of the art of the time. The technology choices upon which HyperNews is based are strongly interdependent, rely heavily on the Java language, and are based on the state of the art technology available at the beginning of the project (second quarter of 1996). In this section we present the technology choices we made and describe how the current sate of the art will affect them in the forthcoming migration of the system to today's (mid 1998) technology.

5.5.1 The Programing Language

Very early in the project definition phase the Java language [71] [72] [73] [74] was beginning to establish itself as a choice language for rapid prototyping and platform independence. The whole implementation was carried out using the Java Development Kit Version 1.0.2 (JDK 1.0.2) together with early releases of the RMI (Remote Method Invocation) and Object Serialization packages.

With the currently available versions of the JDK (i.e., JDK 1.2 final Release), most of the technology described below comes for "free", in the sense they are either part of the Java distribution or have been taken into account for easy integration within the Java environment. For example Remote Method Invocation and Object Serialization, JDBC [100], Security [101] [102] [103], Java Beans [104], Java Foundation Classes (JFC) [105] [106], Swing GUI Components [107], etc.

Currently we are working on the migration of HyperNews to JDK1.2 in order to take full advantage of these integrated features and drop on the way most of the external packages we discuss below.

5.5.2 The Agent Execution Platform

At the time the project started very few agent environments were available. *Mole* [62] [63], developed at the University of Stuttgart IPVR, was one of the few existing environments that was fully implemented in Java. This was a strong advantage with respect to the time critical aspect of the project, portability and the limited man power. Thus, Mole was chosen by the ASAP project to be extended in order to meet the security requirements identified in the MEDIA project. Namely to enforce authoritativeness and untampering of content during migration between locations as well as address the issues surrounding the detection of malicious agents. It has thus been used as the agent infrastructure over which Hep and HyperNews were implemented.

During the implementation, another agent platform was made available by Object Space: Voyager [61]. It is also fully implemented in Java and was considered at some point for further implementations of HyperNews. However due to the commercial state of the product and its related licensing agreements this option was abandoned. Current release of Voyager is Version 2. Most of the available Java based agent platforms suffer from major security problems related to the security model of Java itself. Reason for which the future implementations of HyperNews will rely on the JavaSeal [66] approach addressing these security issues.

In Mole, migration involves stopping an already running agent and sending it to the remote location to resume its execution. However Mole does not account for strong mobility but only for week mobility [117] in the sense that executing threads cannot be resumed remotely at the exact point they were stopped. From the point of view the HyperNews implementation this is not an important restriction but rather an advantage for stability and consistency of the migrating agents that need to reestablish consistent links to local resources upon migration, since the environments are likely to be very different.

Inter-agent communication in Mole can be achieved in two different ways: remote procedure call (RPC) and message passing. Remote procedure call is a synchronous communication mechanism which implements a method call to an agent either on a local or a remote location (Figure 5.2). Message passing can be both synchronous or asynchronous, and allows transfer of messages between agents (Figure 5.3).

```
Object[ ]       paramArray      = new Object[ #ofParameters ];
RPCMessage      rpcm            = null;
Object          result          = null;
...
//Initialization of the paramArray Objects
...
rpcm = new RPCMessage (     callerAgentName, callerLocation,
                            calleeAgentName, calleeLocation,
                            errorSemantics, RPCmethodID, parramArray);
result = (Object) getCurrentLocation().call(rpcm);
```

Figure 5.2 RPC call in Mole

```
String[ ]               paramArray      = new String[ #ofStringParameters ];
UnformattedMessage      msg             = null;
...
//Initialization of the paramArray Strings
...
msg = new UnformattedMessage(     callerAgentName, callerLocation,
                                  calleeAgentName, calleeLocation,
                                  errorSemantics, parramArray);
getCurrentLocation().message(m);
```

Figure 5.3 Message passing in Mole

The general behavior of a mobile agent can be described the following way. Mobile agents are dynamically loaded on the local platform. Then, depending on their “behavior”, they can die or migrate either on their own behalf or be forced to do so. Agents are characterized by a unique agent name (8 integer values within the Mole agent system) and a description string. Upon arrival of an agent on a location an init() and a start() methods are called. Before leaving a location, a stop() method is called. For further details on the Mole agent system, we refer the reader to the following articles which provide in-depth description of the system [62] [63].

5.5.3 The Cryptographic Package

Since the first release of the JDK did not include cryptographic or security packages, we adopted a free external cryptographic library implemented in Java. Namely Cryptix from Systemics Ltd.

Version 1.1 [108] was used throughout the implementation due to its availability at the start of the project. For efficiency reasons, this version used native libraries. Binary distributions of these libraries are available for most architectures such as Windows 95, Windows NT, Solaris, Linux and IRIX among the major ones. However this limitation disappears with release 2.2 and above which are fully implemented in Java. Current available release of Cryptix is version 3.0.3 both in Java and Perl.

5.5.4 User Interface Issues

The limited set of high level widgets provided by the Abstract Window Toolkit (AWT) in the first release of the JDK was to some extent a limitation to the rapid development of the prototype in research conditions. This is the reason for which we choose to use an external widget package. Namely, Objective Blend [109] and Objective Grid for Java [110], from Stingray Software Ltd., were chosen for the project. These are now part of Rogue Wave Software Inc. [111] who merged with Stingray Software in February 1998. These packages provide a useful set of pre-built classes for high level widgets such as trees, grids, tabs, etc. which are now part of the Swing package.

From the point of view of information formatting and rendering, we decided to take advantage of existing technology. It would have been counter productive to build yet another formatting language and rendering tool. Thus choosing HTML as the formatting language and a plain Java-enabled Web browser as the interface for consuming the information appeared to be a sound choice.

5.5.5 Database Issues

The Informix Illustra Object Oriented database was used in the prototype due to its availability within the KryPict companion project of the MEDIA project. A Java package (ITG) was available for interfacing the database from within Java. This was used for integrating the database aspect in the prototype from the information providers' view point. Needless to mention that interfacing with any other database vendor is now a simple implementation issue thanks to the Java Database Connectivity (JDBC)[100]. As a matter of fact, our industrial partner, L'Hebdo in this project chose the Oracle database during the project. It will thus be used in further developments.

5.6 The Hep Based HyperNews Environment

The HyperNews environment is an example of the application layer using the Hep framework as described in the previous chapter. This framework was designed to provide a common abstraction to classes of commercial electronic publishing applications.

Figure 5.4 illustrates the complete architecture of the Hep based HyperNews environment with the agents of the HyperNews specific application layer. Namely, the local HyperNews server agent which is a peer of the local Hep server agent in the HyperNews application layer and the HyperNews article agent which are HyperNews specific document agents. The sample windows illustrate the application user interfaces and their communication channels. The remaining elements which are grayed-out in the HyperNews layer are part of the underlying Hep frame-

work. The remainder of this chapter describes the implementation of the HyperNews environment based on the Hep framework.

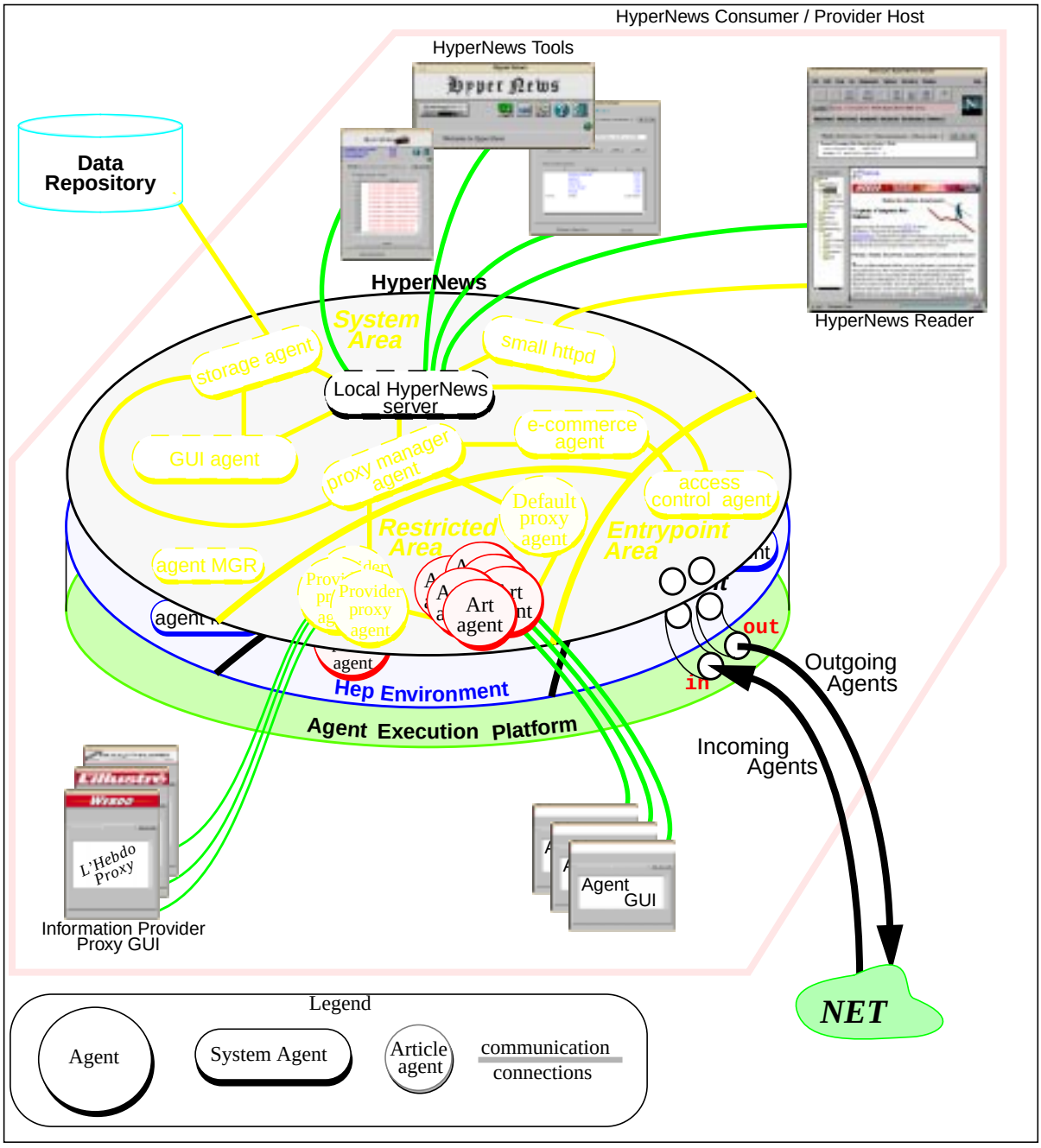


Figure 5.4 The architecture of the HyperNews system based on Hep

5.6.1 Implementation of the Commercial Electronic Document Distribution Model

In the scope of the HyperNews project, we have devised a scheme for the commercial distribution of electronic documents [84] that satisfies the defined security and distribution requirements. It is based on public key encryption and requires a trusted third party between the information consumers and providers which may be for example a credit institution or a bank. Both parties trust the credit institution to authorize the unlocking of the article against payment from the information consumer which is credited to the information providers account. Upon success-

ful payment to the credit institution, the article key is released and a receipt is given to the information consumer for subsequent access. This receipt is issued only for the information consumer that purchased the article. Thus the receipt is *nominative*. However this can also be bound to what ever commercial policy the providers wish to use. A general overview of the model is given in Figure 5.5.

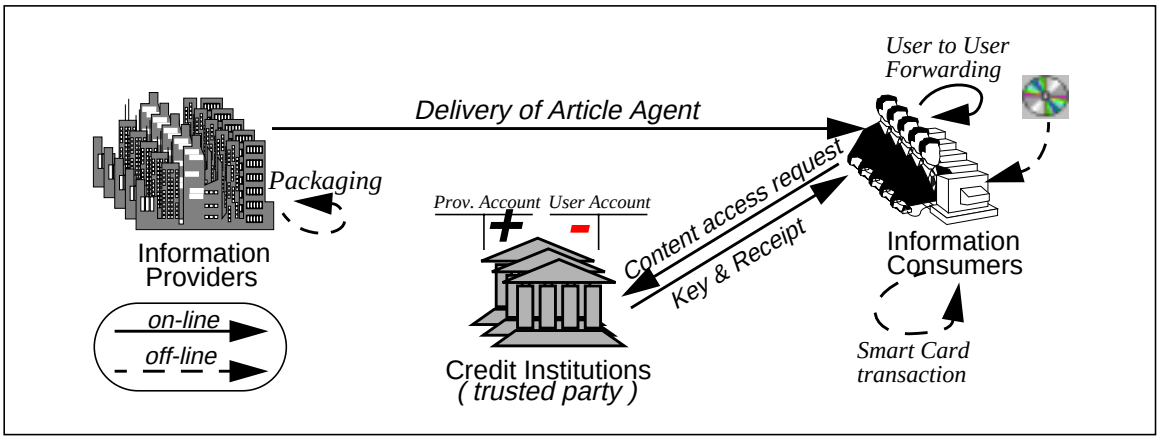


Figure 5.5 HyperNews Model overview

The operational overview of this scheme and its implementation is described below. There are basically two aspects to it: packaging and usage.

The *packaging* consists of preparing the electronic document agent with its encrypted content and attributes for full public distribution. The content, considered as a binary large object (BLOB), is encrypted with a symmetric key (K). This key is itself encrypted with the public key of the accredited credit institution together with the billing information (I). This is done as many times as the number of credit institutions the information provider is willing to support. An article information string (AIS) is added to the agent providing the necessary public (i.e., free) information about the content such as title, authors, price, abstract, etc. In addition we include the code (or its signature) (AC) implementing the operations and policies for accessing the content along with a hash of the encrypted blob (BH). Finally, the encrypted key, the agent code, the

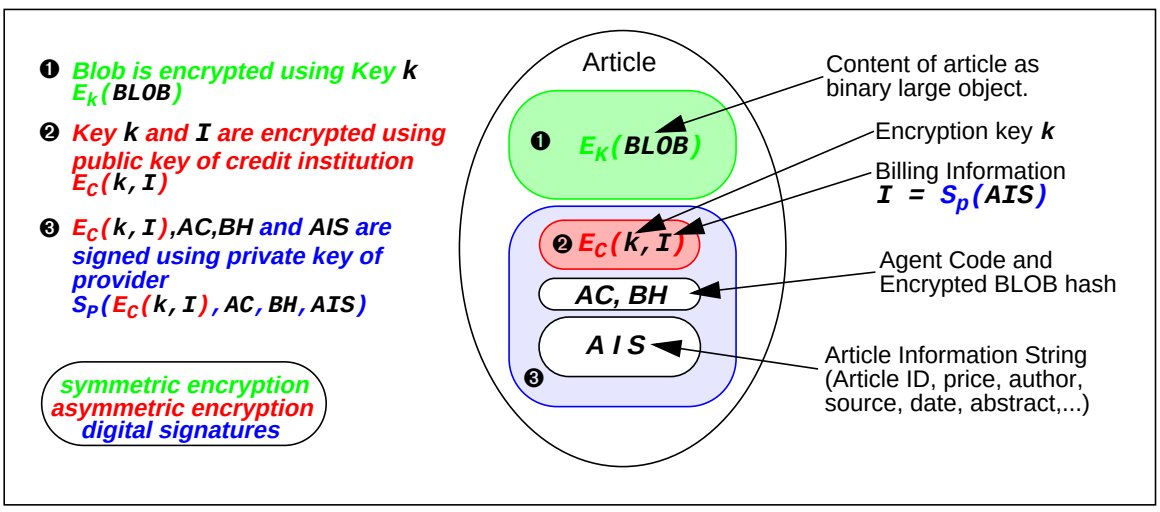


Figure 5.6 The electronic document agent packaging

BLOB hash and the AIS are signed by the information provider with his private key. This process, shown graphically in Figure 5.6., binds all parts of the agent together in a secure way.

The *usage* defines the process of both the access and the subsequent accesses to the content (i.e., unlocking). In both cases there are two steps to cover. The first step is to acquire a session key (T) for the further secure communication with the accredited entity (i.e., credit institution or alike) which will process the access request and release the document key. The second step is the actual content access request and acquisition of the article key and receipt as shown in Figure 5.7 and Figure 5.8. The request is formed by extracting from the article agent the encrypted key

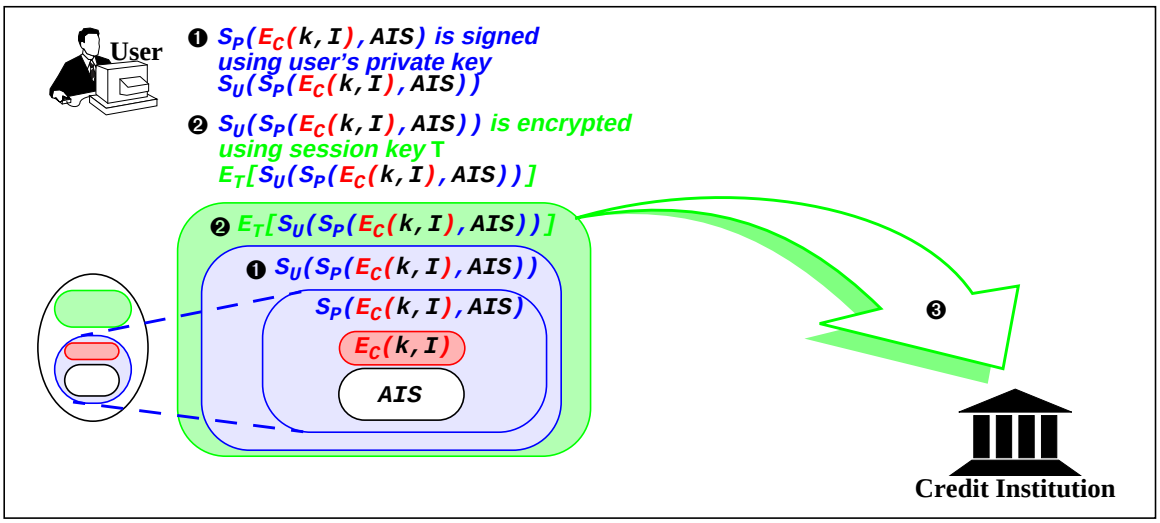


Figure 5.7 Second step: access request

corresponding to the credit institution and the article information string. This is then signed by the user and the result is encrypted using the session key previously acquired in the first step. Upon receiving such a message, the credit institution will be able to decrypt it knowing the previously issued session key, verify the signatures of both the user and the provider and thus reveal the AIS and the encrypted key. At this point, the billing occurs and if it is successful, the article

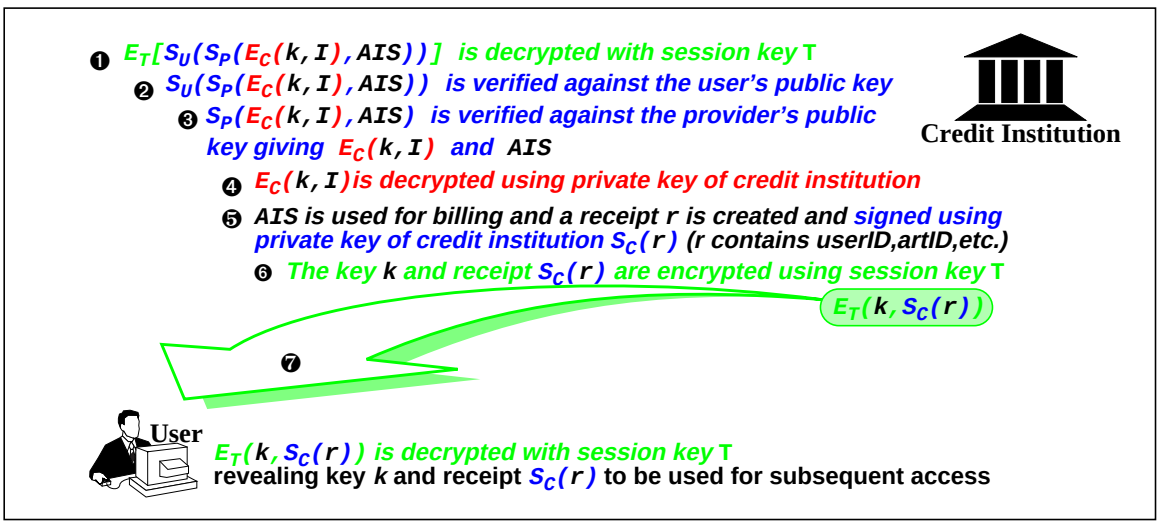


Figure 5.8 Second step: article key and receipt

key K will be decrypted with the private key of the credit institution and a signed receipt generated for this transaction. Finally, the article key and the signed receipt are encrypted with the session key and the result sent back to the user (i.e., to the article agent responsible for releasing its content).

Subsequent access to the article content is done in exactly the same way except that a receipt is appended to the request of the second step. Upon successful verification of the receipt by the credit institution (i.e., verification that receipt issuer signature and user identification match between receipt holder and requestor), the article key will be returned to the user but no receipt needs to be returned unless a particular commercial policy requires it.

5.6.2 Key Management and Acquisition

The only keys that need to be exchanged between the participants (i.e., information providers, consumers and credit institutions) are their public keys and the session keys between the information consumers and the credit institutions. For the time being, no use is made of certification authorities for public key acquisition. However this can be integrated easily in future implementations. The session key acquisition is secured by using asymmetric cryptography for encryption and signatures.

Every participating entity knows its own private key. The credit institutions need to know the public keys of both the information providers and consumers, which is a reasonable assumption for a trusted third party. The information consumers need to know the public keys of the information providers and the credit institutions. However, the information providers only need to know the public key of the credit institutions. Finally, the information providers know the document/article symmetric key that served for content encryption, which is also a reasonable assumption since they own their content/added value. This is summarized in Figure 5.9. From a

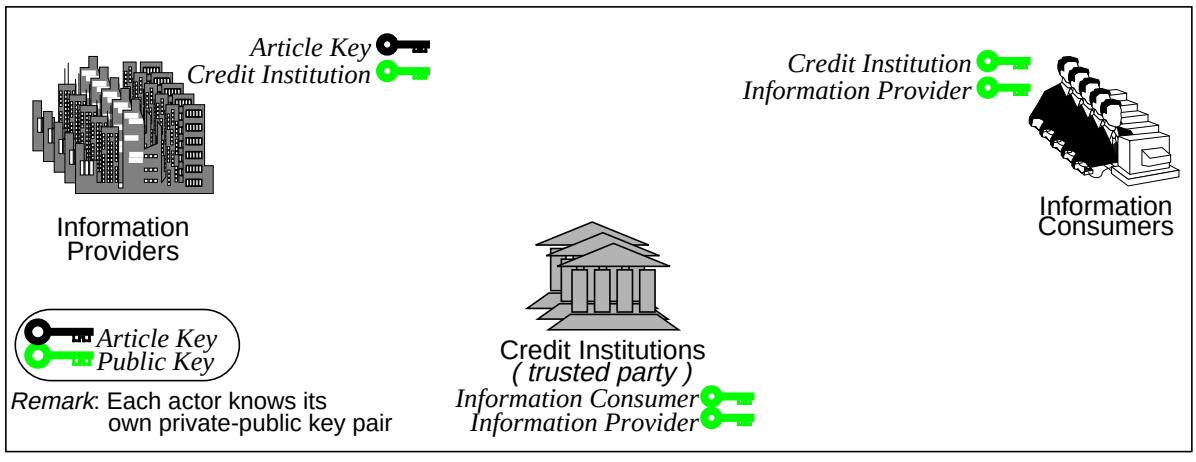


Figure 5.9 Key management: who knows what

key management point of view, the major advantage of this document distribution scheme resides in the fact that there is no overhead for document key exchange or replication since the document key is encrypted with the public key of the accredited institutions and held by the article agent itself. Thus, even in case of information provider bankruptcy the content can still be

accessed through one of the credit institutions. Moreover information providers can even forget about the article key.

5.6.3 The HyperNews Article Agent

The HyperNews article agent is a subclass of UserAgent implementing the MobileAgent interface allowing it to migrate between agent platforms and the TimerCallback interface for inactivity time-outs. Since the main idea behind the design of the article agent is to allow its free distribution without copyright infringement (i.e., superdistribution [6]), the article content is stored encrypted together with the article encrypted key within the agent and the article agent is responsible for its own security.

The HyperNews article Agent is the Agent wrapper of the article content. It provides a set of instance variables:

- the article content stored in an instance of the Article class which is serialized, encrypted and stored in a byte array.
- the Article Information String (AIS class)
- the signed MD5 hash of the Article Information String
- the article key and the signed MD5 hash of the AIS encrypted with the public key of the credit institutions (CIKey class). One per credit institution stored in an array.
- the signed MD5 hash of the all the CIKey objects stored in an array.
- a optional array of embedded HyperNews Article Agents thus providing a recursive encapsulation of different levels of added value. This way an article agent can become a container together with its own added value.
- state reflecting the agent type, the area restriction code, whether it is idle or not and information about the creation location, the previous location, the current location, source and destination locations when migrating.
- the action to undertake upon expiration of the article agent timer. Namely, save before being removed, simply remove, restart the timer or do-nothing.
- information whether a Provider Proxy exists locally or not and if the case arises, its agent name.

together with the following interface:

- various constructors to be used according to the desired type of article agent (i.e., article, container or both).
- an initialization method init(), which is called automatically after migration or when reloaded on a local agent execution platform.
- a start method start(), which is also automatically called right after the initialization method.
- a stop method stop(), automatically called to complete clean up actions before migration or local storage.

- the timer expiration callback method hasExpiered(), which is called automatically upon expiration of the article agent timer.
- the migration method moveTo(LocationName ln), which is called to send an article agent to a destination location.
- the standard Mole RPCMessage dispatcher dispatch(RPCMessage rpcm), returning a result of type Object. It calls two private methods of the article agent: openHNArt() and getEmbededFile(). In both cases the returned Object is an instance of a byte array of the content which is returned as the result of an http request to the browser by the http daemon system agent of the local environment. The first one is called when the user requests access to the article content. The second is called to retrieve the article embedded files of an article currently being displayed to the user.
- other conveniency methods to operate on the article agent attributes.

5.6.4 The HyperNews Protocols

From a protocol point of view, communication is achieved with *two-state return agents*. Two-state in the sense that at any given time such agents are only in either of two states, namely outgoing for remote service request or returning after service completion. In other words, agents are created at a given location with a specific mission to accomplish. The agents are then sent to their destination location where they will request the service(s) and hopefully return to their initial location holding the result(s). Upon completion, the agents are disposed. The term *hopefully* for the return of the agents was used intentionally because anything can happen to the agents during their journey to, from or even at the destination location. There is thus no guarantee whatsoever regarding their return. For this reason, every such two-state return agents, prior to leaving, is associated with time-out mechanisms at the source location. That is, a semaphore-like object is created with a unique identifier to be used for notification upon return of the agents. It is combined with the time-out mechanism so that the waiting threads are notified accordingly and the resources relinquished.

There are three types of communicating parties: the information / service providers, the information / service consumers and the credit institutions which is trusted by both the providers and the consumers. It is to be noted that the roles of the consumer and provider can be combined for example in case of article forwarding among consumers, the sender then holds the role of a provider and reciprocally. The different types of agents that migrate among the communicating parties within HyperNews are summarized in Figure 5.0.

5.7 The HyperNews layer

The HyperNews layer implements all the HyperNews application specific components and logic, which are mostly reflected through user interfaces.

Both the information consumers and the information providers run the same environment with slight differences depending on their role. This is expressed by a HyperNews property defining whether a user is an information consumer, an information provider or eventually both. This is specially useful when an information consumer wants to be able to republish material he

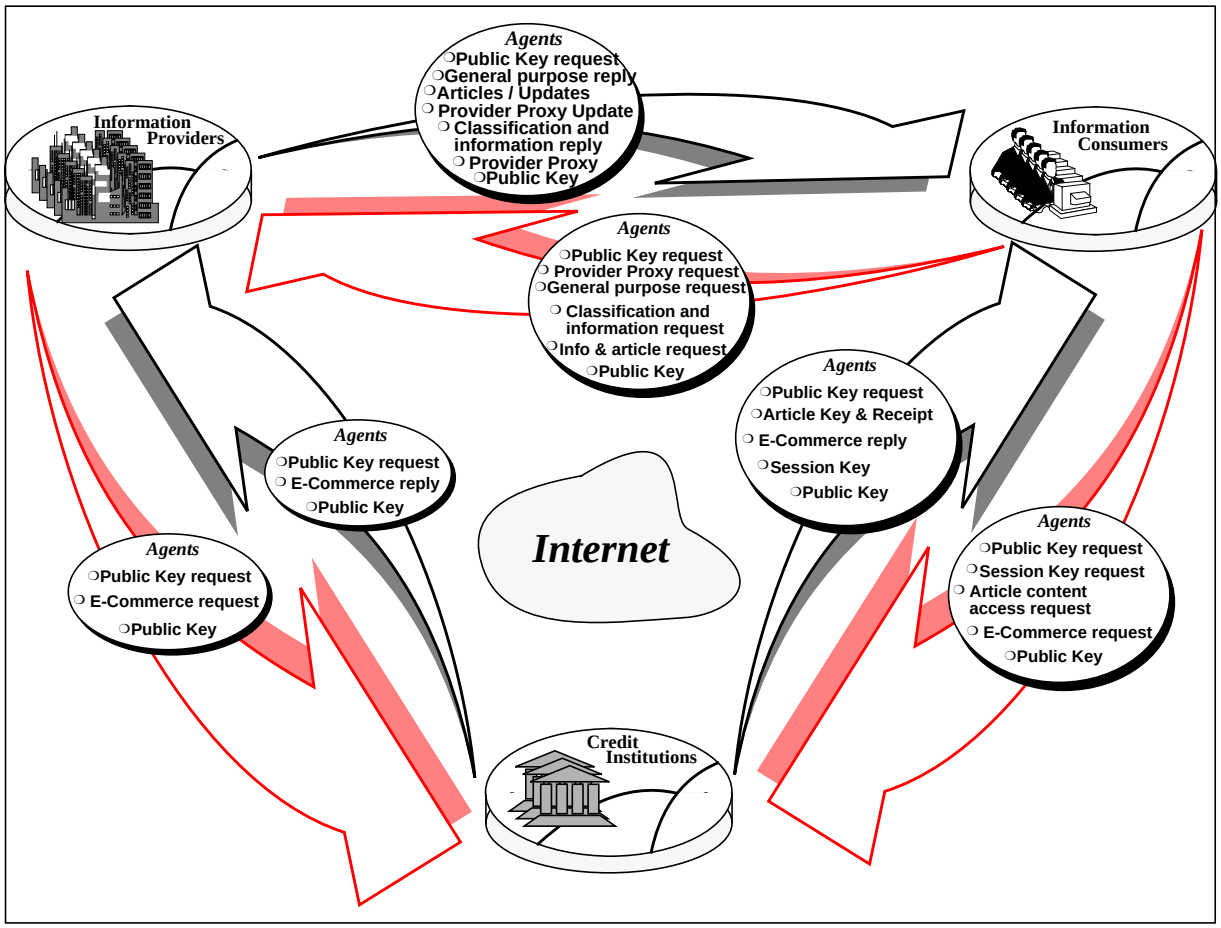


Figure 5.0 The HyperNews agent migration overview

does not own thus becoming an information provider of his own added value together with other information provider's content. All this being totally legal since access to content is bound to successful payment or presentation of a valid proof of purchase (i.e., receipt). Based on these properties, the HyperNews environment at launch time will be tailored accordingly. The main HyperNews user interface in collapsed mode is shown in Figure 5.1.

The HyperNews environment can be tailored in a number of ways through various properties allowing the user to define in a persistent way his preferences such as the viewing mode, browser to use, filtering attributes, etc.

5.7.1 Information Consumer Tools

In order for the user to be able to define, customize and access his personal electronic newspapers (i.e., contexts) the following tools are provided:

- the *Context Manager*, to manage personal electronic newspapers
- the *Information Profiler*, to specify for each context both where the information is to come from and what is of interest to the reader
- the *Presentation Profiler*, to specify for each context the layout or in other words how the retrieved information is to be presented to the reader
- the *HyperNews Wallet*, to manage all the electronic commerce related tools.

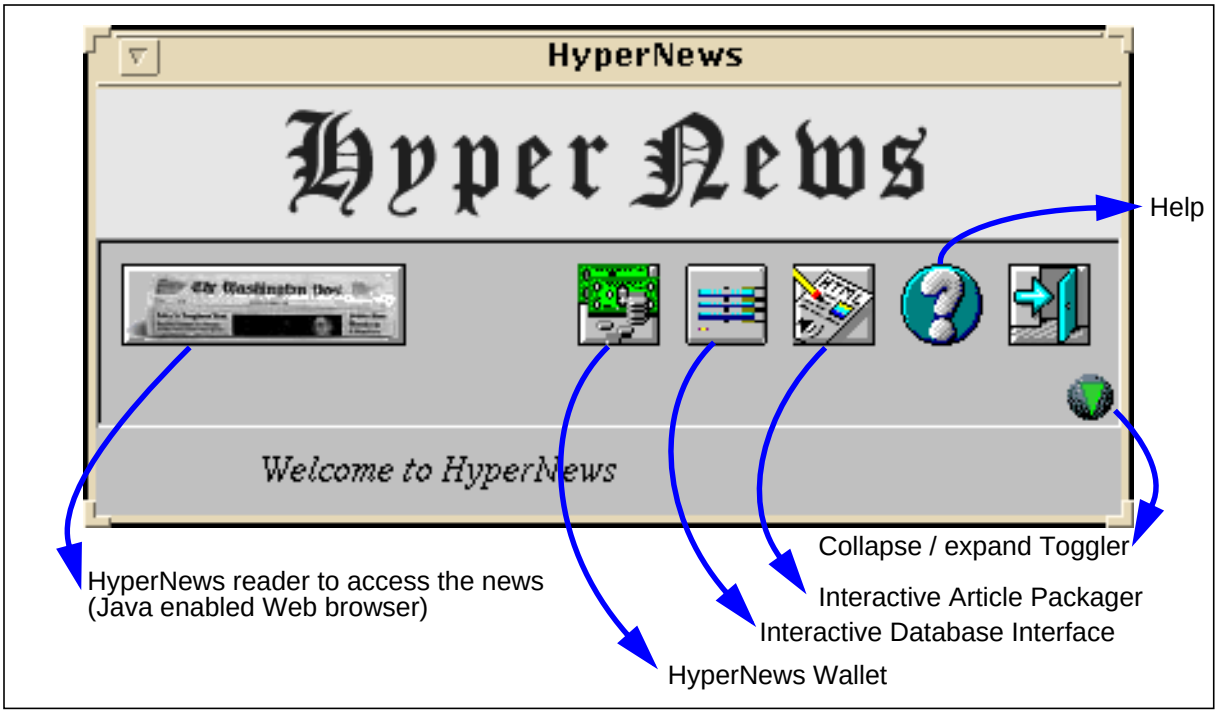


Figure 5.1 The main HyperNews user interface (collapsed)

- the *HyperNews reader*, to access the hypermedia electronic newspaper system
- the *Preference and utility* tool, to define user preferences

The Context Manager

The context manager allows the user to manage his personal electronic newspapers or information contexts in a similar way to a “news stand”. With this tool the user can create, remove and edit his information contexts. These are stored on the users host in a specific directory of the HyperNews system. Throughout the implementation, contexts are represented graphically as “tabs” or “tabed panels”. An example of the context manager with three information contexts (projects, private and work) is shown in Figure 5.2.

The Information Profiler

Once the user has defined one or more contexts, for each one of them he will have to specify his interests using the information profiler. This is done in two steps. First by selecting among all the available information providers the ones from which information is requested. Second, for each requested information provider by selecting from the provider’s classification the elements he is interested in.

An example of the information profiler for the sample “work” context is shown in Figure 5.3. In this example all three available information providers are selected (left hand side). The selection is identified by a “tic” over the folder image. On the right hand side, the whole classification of the currently selected information provider (L’Hebdo) is displayed and two classification items are currently selected (Laboratoire and Editorial). The selection of an information provider on the left hand side triggers the display of its corresponding classification on the right hand side from which the user may select the desired items he is interested in.

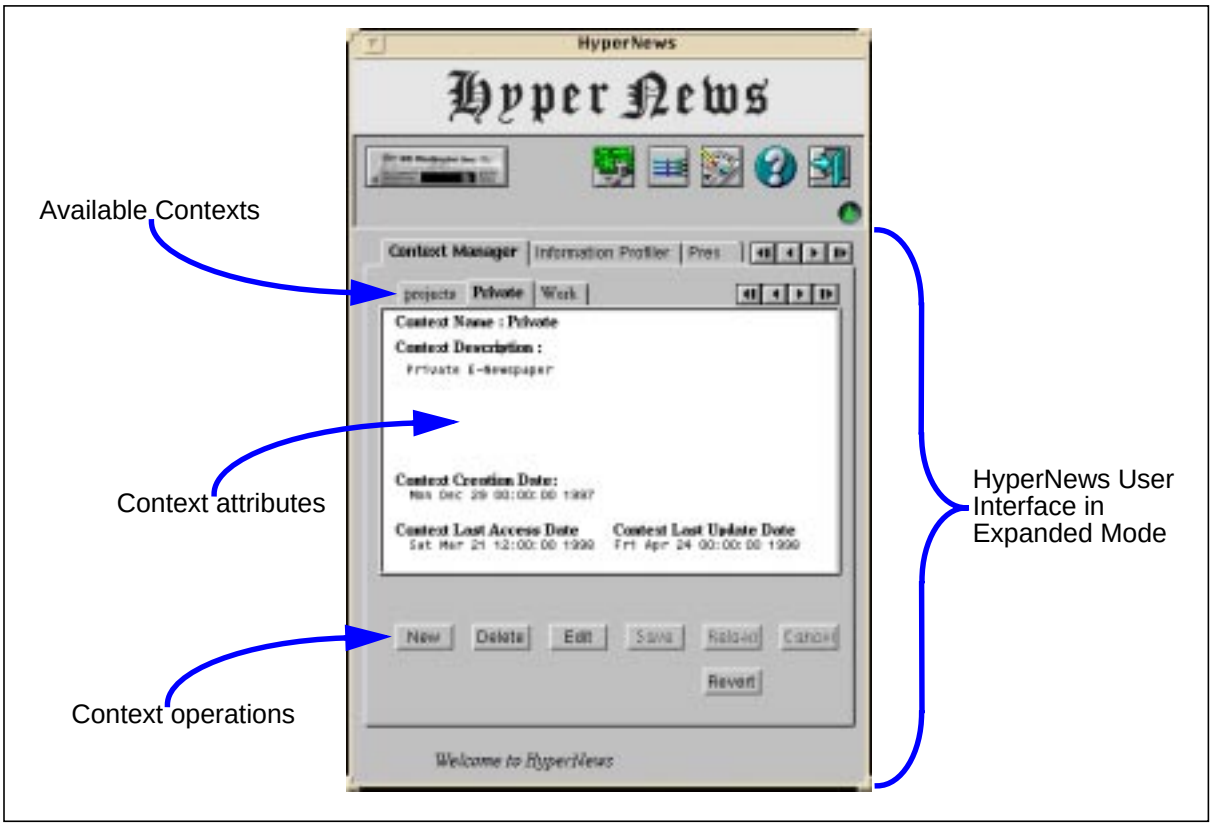


Figure 5.2 The HyperNews Context Manager

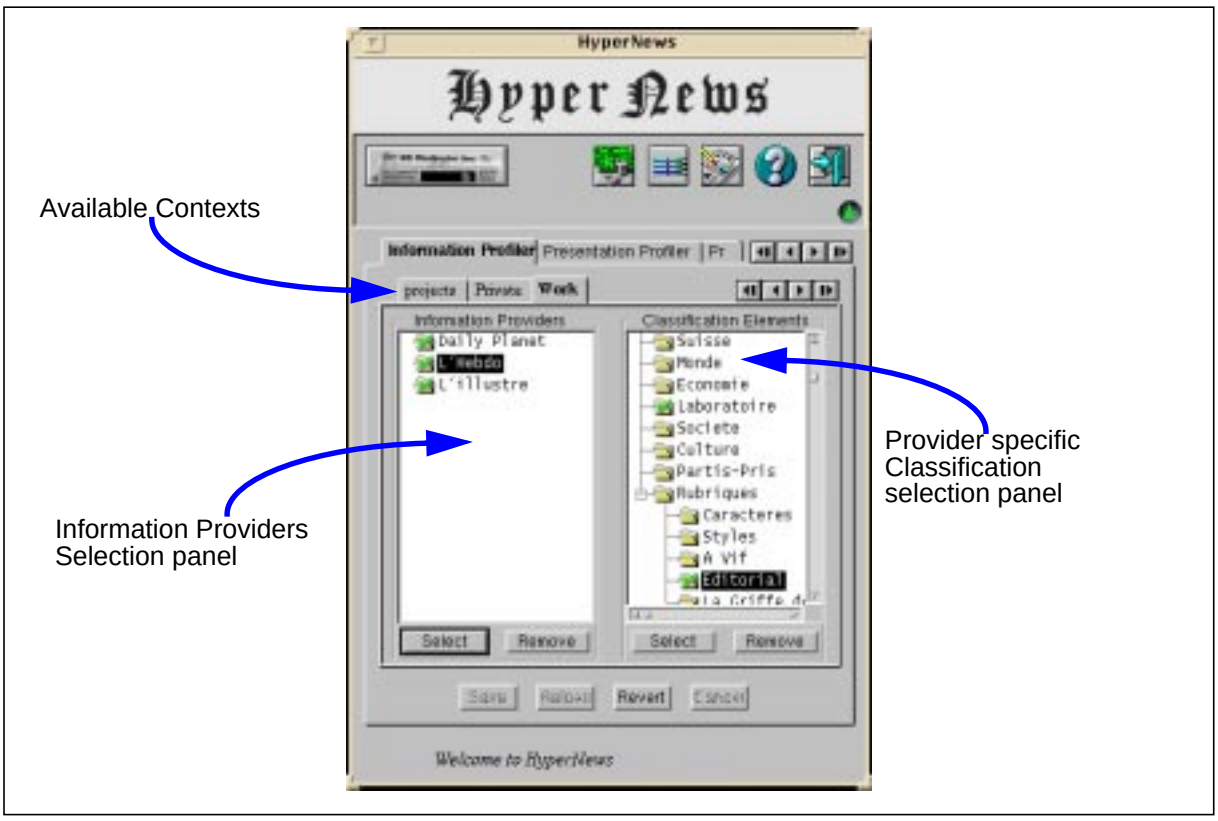


Figure 5.3 The HyperNews Information Profiler

The Presentation Profiler

The last step to fully define a users context is achieved with the presentation profiler. For each information context, the user defines the layout (i.e., structure) of his personal electronic newspapers and interests the previously selected classification items from the requested information providers into this structure.

An example of the presentation profiler for the “work” context is shown in Figure 5.4. In this example, the top panel (Selected Classification) shows for each chosen information provider the selected classification items available. These items are available for insertion in the user defined structure of the bottom panel (Sections and Content). The bottom panel shows the “root” page of the work context in which the user chose to put all the “Editorials” of the selected information providers. As a result, when the user will access this page he will be presented only articles that match these classification items. Namely all the editorials of the selected information providers.

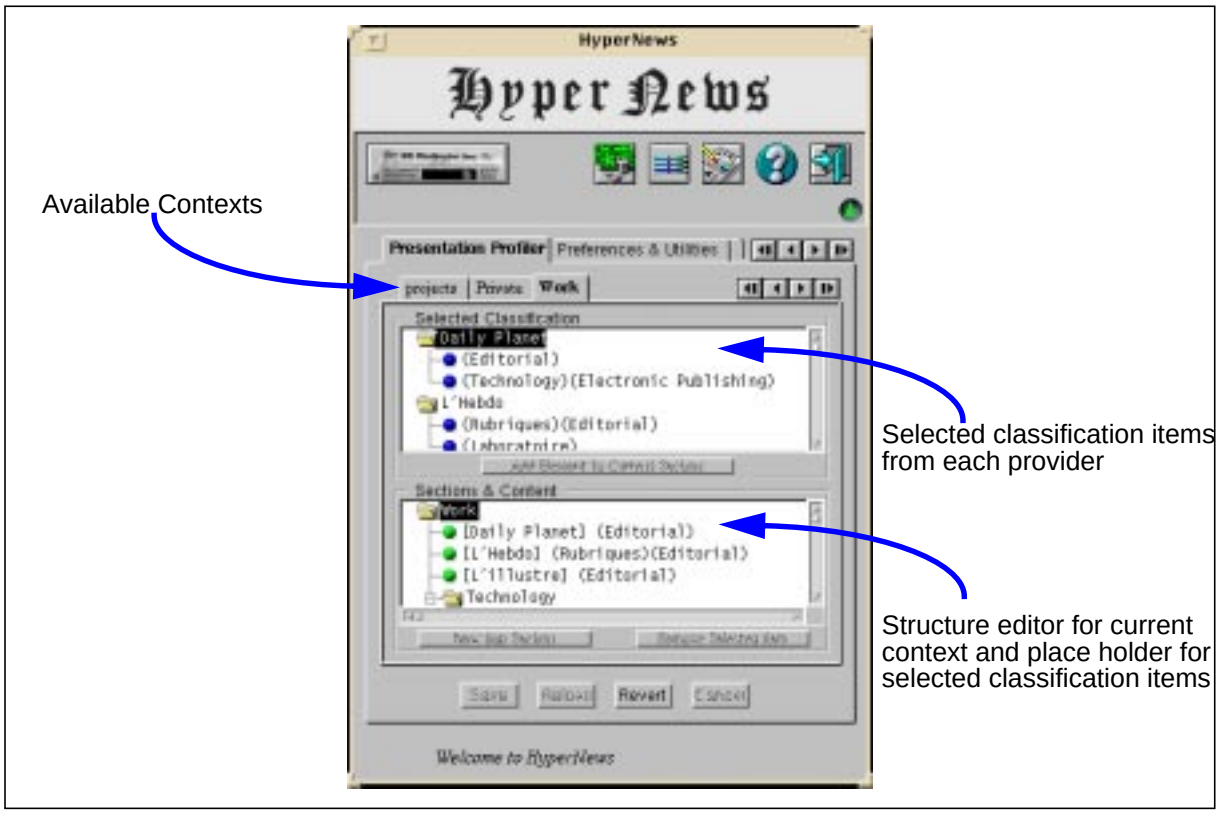


Figure 5.4 The HyperNews Presentation Profiler

The HyperNews Wallet

This tool is the user’s interface to the electronic commerce components. In the current implementation of the HyperNews prototype, it only accounts for “fake” cash like transactions. However other type of payment systems and protocols have been anticipated be they cash, debit or credit. Ultimately, the user should be able to accommodate for any third party electronic commerce system in a transparent way and express his preferences in order for the system to take these into account in the commercial transactions. This tool is also the user’s interface to receipt

management. As shown in Figure 5.5, the user can have access to all the receipts currently owned.

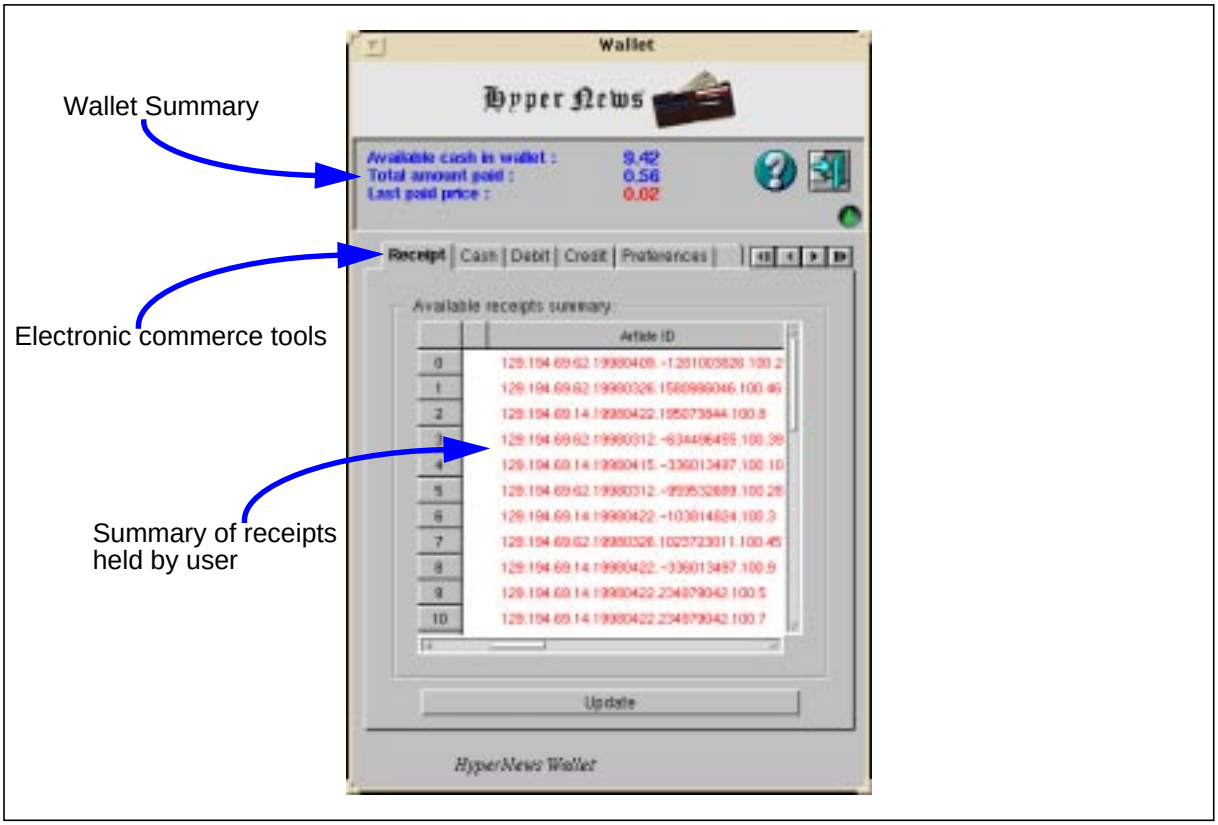


Figure 5.5 The HyperNews Wallet (receipt summary)

The HyperNews Reader

Finally, the user accesses his personal electronic newspapers through the HyperNews reader which can be any Java enabled Web browser. A default version of netscape 3.0 is included in the HyperNews distribution The basic structure of the interface is composed of three elements each of which is in a separate frame. The first two hold applets whereas the last one is a simple frame in which resulting data is displayed in HTML. Communication between the applets is implemented using a the UDP protocol. Each applet implements a UDP server that interprets the received messaged and take actions accordingly. This is also used by the HyperNews environment to communicate with these applets when needed. The http protocol is first used to exchange information such as the port number between the applets and the environment. Following is a description of these three elements:

- The *context selector*: is a Java applet in the top horizontal frame. It allows the user to skip through his contexts (i.e., electronic newspapers) using the Tabs labeled with the context names. Each context panel displays summary information available such as the last time the context was accessed, the number of available updates, etc. It also provides a button labeled “Update Now” allowing the user to trigger an update process for the given context. Selecting a context via a Tab triggers a change in the sections selector to display the corresponding sections of the newly selected context. This is achieved by sending a mes-

sage (UDP datagram) to the section selector applet in order for it to retrieve the sections of this new context.

- The *sections selector*: is also a Java applet in the bottom left vertical frame. It shows at all times the sections of the currently selected context in a tree structure. The selection of a section triggers an http request to the HyperNews system. its result is redirected in the viewing area. Such a request corresponds to requesting the summaries (as desired by the user) of all articles available locally for the current section of the current context filtered according to the users preferences.
- The *viewing area*: the remaining frame (bottom right frame), shows any of the following depending on the user's action:
  - *article summary container page*: showing all the HyperNews articles that match the selected section of the current context. Matching criteria can be customized according to user preferences (e.g., all the articles, only new or old articles since or until last update, already read or unread articles, articles published before and / or after given dates). For each article summary, the user can choose which attributes should be displayed (e.g., source information, image of logo if available, authors, abstract, publish date, expiry date, price, receipt existence and options such as article annotations, send or forward article, delete article). The article title is always shown as it is the hyper-link to access its content. The retrieval of article summaries is initiated by the user when selecting a section of the current active context in the sections selector applet. An example is shown in Figure 5.6.
  - *a HyperNews article*: is shown when the user has requested to access its content. Upon successful payment or receipt verification, the article content is displayed in the viewing area. HyperNews hyperlinks are identified by an "HN" chain logo. When selecting such links the HyperNews system will propose the corresponding summary information to the user for acceptance. If the article cannot be found locally, it will be fetched from its source. The other hyperlinks are normal Web links that can be browsed in a transparent way by the user. An example of a HyperNews article is shown in Figure 5.7.
  - *a plain Web page*: can be accessed directly from any HyperNews article containing plain Web URLs to access external Web based referenced material. This is an advantage of using standard viewing and rendering tools.

The Preference and Utility Manager

This tool allows the user to tailor the HyperNews environment in a number of ways such as general information, reader preferences and article summary preferences:

- the *General* preference panel allows the user to specify user name, e-mail address, a description string and the role(s) played (i.e., consumer, provider or both). See Figure D.1 (a) in Appendix D.
- the *Reader* preference panel allows the user to specify which HyperNews reader is to be used, either the built-in netscape or one's own browser. It also provides the means to se-

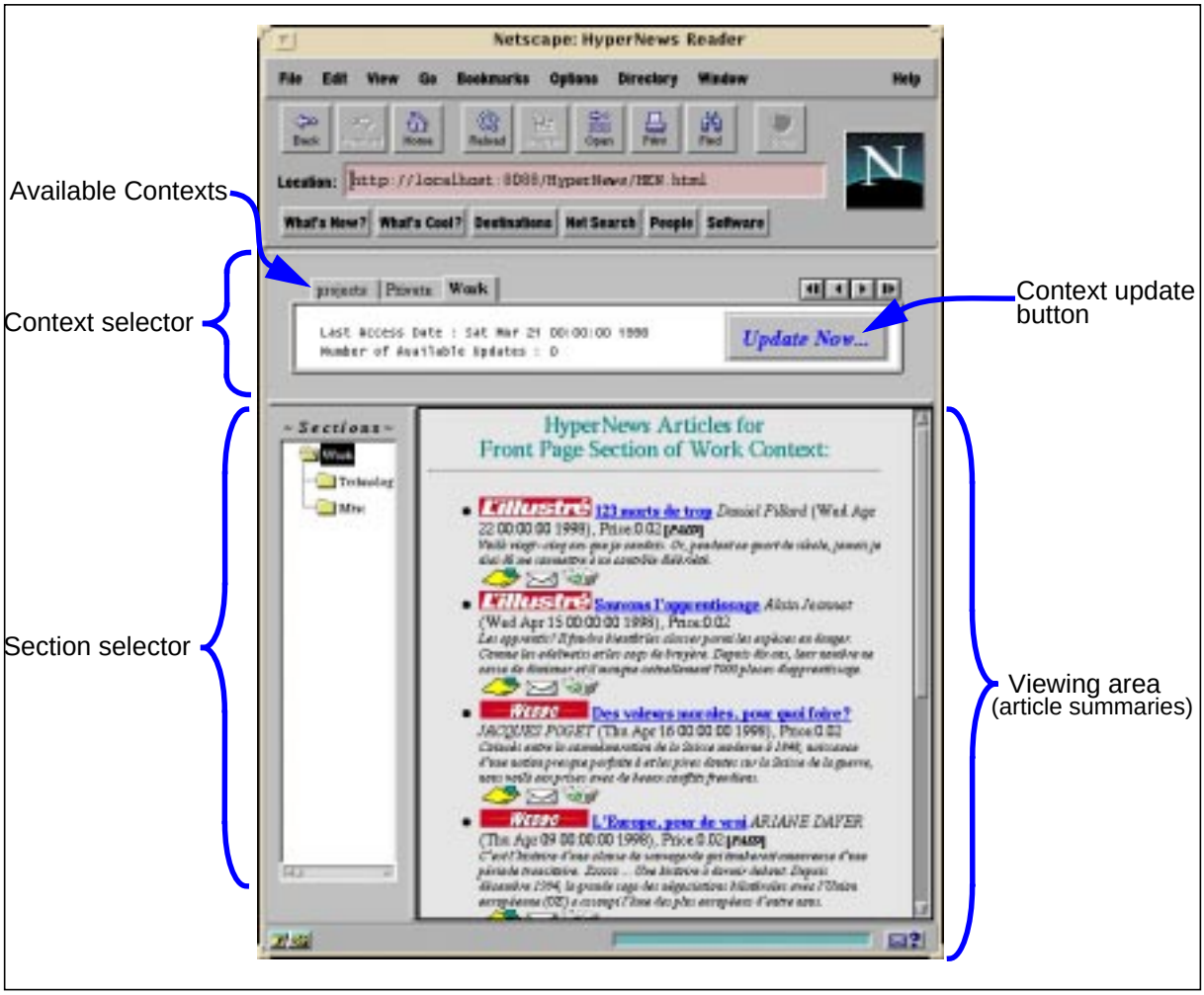


Figure 5.6 A HyperNews article container page

lect the article filtering attributes when accessing a container page. In other words article selection in personal newspaper pages will be done according to criterions selected in the viewing mode selector. Here the user has the choice of filtering the articles among a set of attributes. Namely either all the articles or selected according to combinations based on the last update (old or new), the payment (read or unread) and date intervals (before and / or after given dates).See Figure D.1 (b) in Appendix D.

- the Article Summary panel allows the user to select among all the attributes attached to articles the ones that are to be shown in the container page when displaying the summary of articles. Namely, the source name and logo, the author, the abstract, the publishing and expiry date, the price and the existence of a receipt to show whether the article has already been paid for or not. To be noted that the article title will always be presented as it is the link to the article itself. Finally, the user can choose among three basic tools to annotate, send or delete the article. See Figure D.1 (c) in Appendix D.

These tools are very basic and are only provided to show where and how user preferences are to be taken into account. Further implementation of HyperNews would require higher levels of such tools and much greater control over personalization and utility tools.

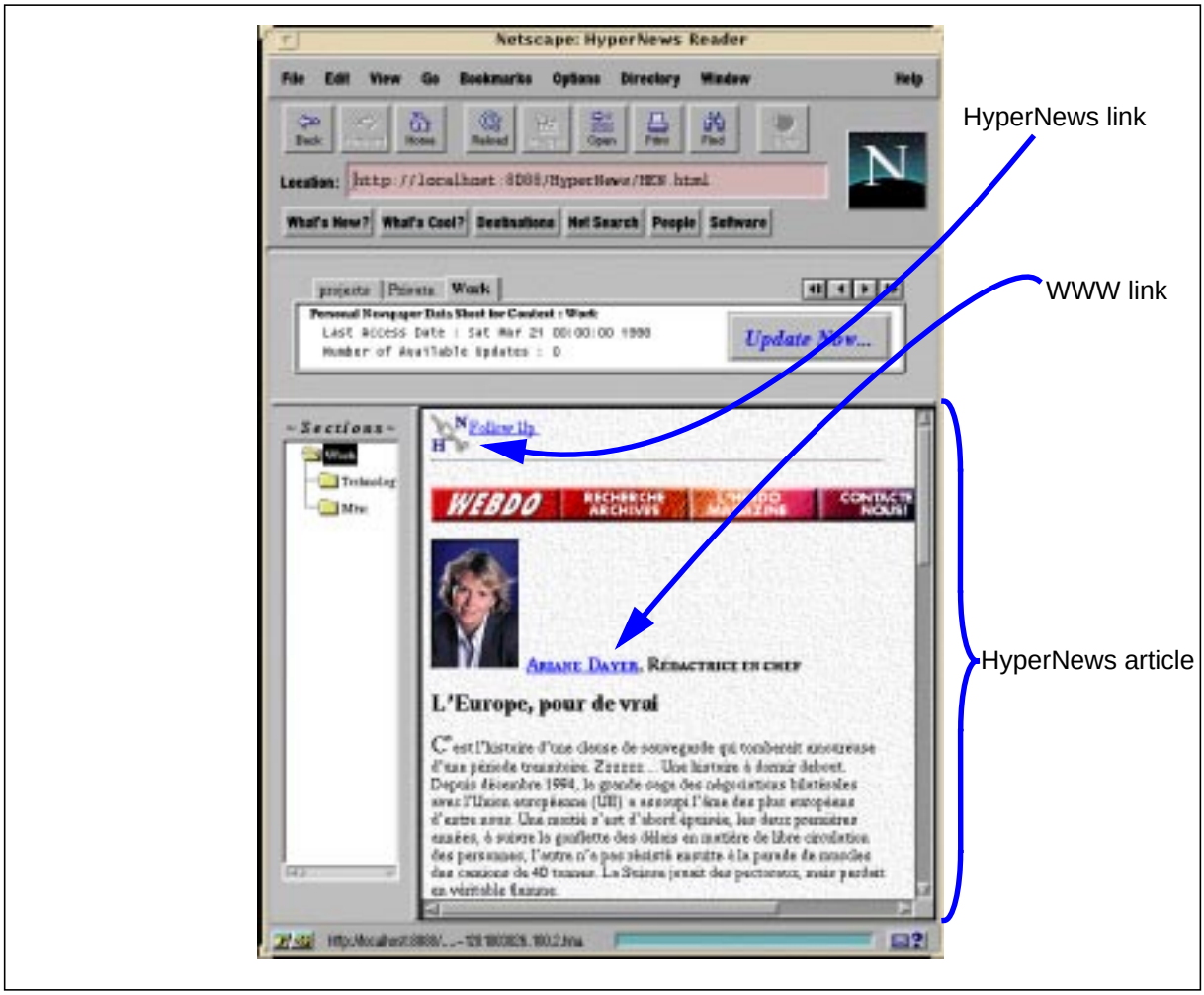


Figure 5.7 A HyperNews article example

5.7.2 Information Provider Tools

In order to publish content, information providers need tools for creating the secure agent based HyperNews articles together with their attributes. From this point of view, an important requirements was the ability for the provider to reuse existing material in electronic form such as ready made HTML formatted Web pages. Two ways of packaging these articles either interactively or in batch mode are provided. Both ways rely on the http protocol for retrieving the content as well as its embedded elements such as images, applets, etc. This offers the advantage of a unique way of accessing the “raw” data through a URL (i.e., http, ftp, file, etc.). The structure of the HyperNews article is discussed in further details later in this section.

The Interactive HyperNews Article Packager

This tool allows any user to package interactively a HyperNews article. There are four steps to achieve this:

1. First, the article content must be acquired through a URL of where the data can be retrieved from. An example of this step is shown in Figure 5.8. The top group is used to set the URL of the source, retrieve it and parse its content in order to acquire the embed-

ded elements as described in the resulting group at the bottom showing a summary of the article's content. In this example, four embedded images were identified and retrieved. The content of the article itself is stored in the file named "index.html" with a total of 5 files and 32 Kilobytes.

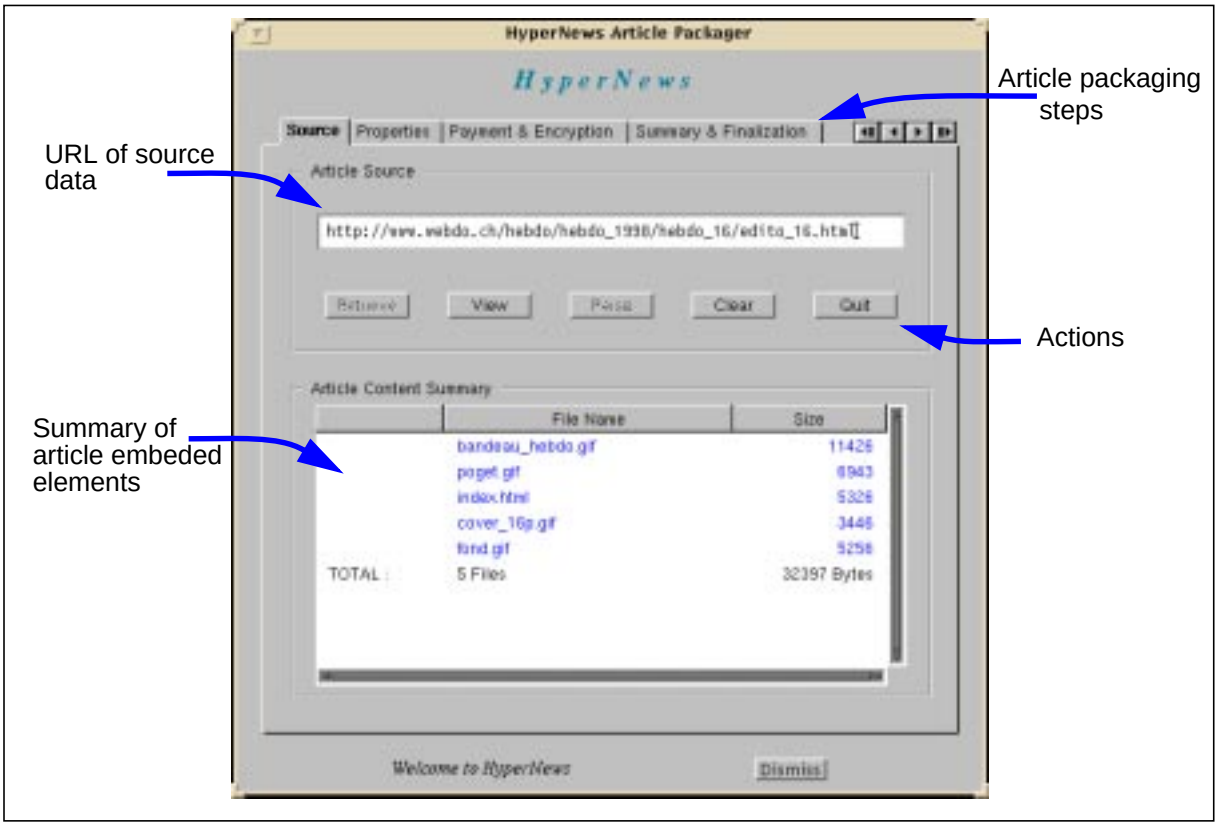


Figure 5.8 The HyperNews Interactive Article Packager (source data acquisition)

2. The second step consists of defining the article's properties such as title, author, abstract, price, primary and alternate classification, publish date, expiry date, volume and issue number, keywords, etc. At this point, a unique article identification number will be assigned to the article based on a subset of these attributes and a sequence number thus guaranteeing uniqueness of the article ID.

3. The third step provides a way to define whether the article is to be encrypted or not. In case encryption is requested, the article publisher will be able to define which credit institutions are to be supported for processing a payment for this article and whether the information provider himself is to be included as potential payment processor. This last option can be very useful for allowing internal processing of article access inside the providers domain for example or if a user has a special agreement with the provider and does not require anonymity.

4. Finally, the last step towards publishing consists of verifying the summary and finalizing the publication of the article. At this point the article will be checked for consistency and published thus becoming available for retrieval and dissemination.

The Batch HyperNews Article Packager

The previous interactive tool is specially useful for information consumers and sporadic publishing of individual articles or also when articles are published on irregular time intervals directly by the authors themselves. However it is most likely that such an editorial process will be discussed prior to committing the publishing process of a set of articles. In this case it would be unrealistic to require such an interactive process for each and every article but rather have them all published in batch mode. For this reason an other tool was included in the set of information provider tools: the Batch HyperNews Article Packager. This tool, shown in Figure 5.9, allows an information provider to do exactly this by providing a file in a specific format holding for each article the required information allowing it to be packaged into a HyperNews article agent.

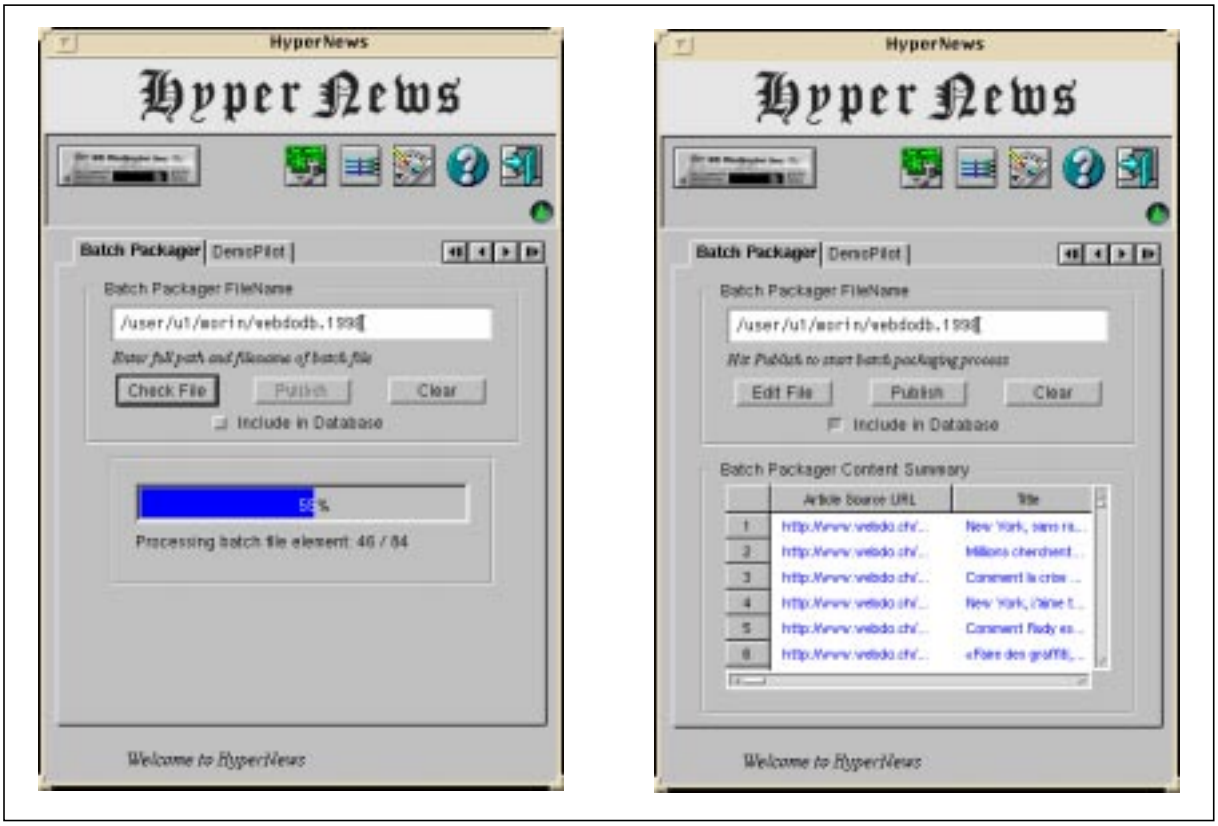


Figure 5.9 The HyperNews Batch Article Packager

This is done in two steps: first, the file is read and checked for consistency and integrity (left image of Figure 5.9). Article agent names are allocated in advance thus allowing cross-referencing of HyperNews articles within the file. Second, the user after checking the results of the first step in a table can commit the publishing of all the articles and also include them in the database if desired (right image of Figure 5.9).

Other Information Provider Tools

In a similar way to the information consumer, the information provider is able to tailor the HyperNews environment according to specific situations and needs.

the *Provider Preference* allows the information provider to set such user dependent elements. For the time being only the database settings have been taken into account allowing the

information provider to require the use of a database or not. In the former case database attributes such as database vendor, name, user name and driver can be keyed in. However it is anticipated to offer far more fine grained control over a broad range of settings relevant to the information provider. See Figure D.2 (a) in Appendix D.

the *Classification Manager* allows the information provider to create and manage the classification scheme. This classification scheme will be used to allocate published articles main and secondary classifications upon which information consumers will be able to express their interest. See Figure D.2 (b) in Appendix D.

5.8 Security Issues

Security is of paramount importance in an infrastructure such as HyperNews specially when it comes to issues related to copyright, intellectual property rights and mobile code migrating over open networks between a priori untrusted locations. Reason for which such an infrastructure must enforce authoritativeness, trust among communicating parties, untampering of content during migration and prevent copyright infringements through encryption.

The RSA [43] [44] public key algorithm (512-bit) is used for asymmetric cryptography. It is used for both encryption and digital signatures. The MD5 [112] one-way hash function was used to compute message digests of transferred objects.

Like for the article content, we choose to use the IDEA [17] symmetric block cipher for the session keys. The keys are 128 bits long and their validity in time can range from unique usage (i.e., use once and throw away) to a given time interval based on the providers policy with respect to this matter.

In the current implementation the deciphering of the article content is achieved by the article agent itself upon receiving the article key from the credit institution when successfully processed. The key is kept in memory only for the time needed to complete the decryption and immediately garbage collected afterwards. This way, the key is neither stored in the users environment nor revealed to other agents but the one it concerns. The arguments such as a memory dump and other similar attacks stay valid however scrambled memory techniques and hardware could be used to address such issues. Moreover, the article key being unique to a single article even in case of a compromised key this would only give access to one specific article. And such article having properties of low value and short life time makes it even less attractive to break.

An other issue of critical importance is the acquisition of the public keys over open networks. Reason for which in future implementations the trust chain could be enforced through certification authorities such as VeriSign’s digital IDs or the ISO authentication framework X509 protocols [113]. From this point on, security is enforced at all levels of the architecture: agent environment, core environment and HyperNews application through techniques mentioned above.

5.9 Assessment and Validation

In order to assess and evaluate the achievements of the HyperNews project, this section briefly presents some quantitative results and relates a public demonstration held at Computer'98 in Lausanne in cooperation with our industrial partner L'Hebdo.

5.9.1 Quantitative Results

To evaluate and test the HyperNews prototype, we have used a set of articles taken from all the issues of L'Hebdo between January and April 1998 (i.e., 16 issues with a total of 228 articles). The overhead of the agent encapsulation of an article is roughly averaged to a constant value of 1.5 Kilobytes. As shown in table 1, examples are provided for the smallest and the biggest article. The average values of the whole sample are also provided.

	Article Object Size (bytes)	*Article Agent Size (bytes)*	*Overhead (bytes)*	*Overhead (%)*
Smallest Article	677	2'106	1429	211.08
Biggest Article	180'399	181'801	1402	0.78
Average of the sample (228 art.)	42'262	43'723	1460	3.46

Table 1: HyperNews Article Agent Overhead

The second performance measurement concerns the cost of accessing the content. For the time being it requires one or two network accesses depending on the life time of the session key. The rest is bound to processing power for the decryption of the content. In the scope of the evaluation and testing, these issues seamed reasonable. Testing platforms included Sparc 4, Sparc Ultra 1 and 2. However, the bottom line is the network as the transaction the transaction is online. Thus performance is inherently bound to the network load.

The complete Hep framework and the pilot application HyperNews took 6 man-months for the design and 6 man-months for the development. The total sizes of the code, both in lines of source code (Java) and byte-code are summarized in the following table

	Lines of code	*Byte-code*
Agent platform	9.200	126 KB
Hep	13.000	200 KB
HyperNews Application	17.000	350 KB

Table 2: Overall size of system

5.9.2 Public Demonstration

The Computer'98 exhibition was held for its 20th edition at Palais de Beaulieu, Lausanne, from April 28 to May 1, 1998. It is the major computer and information technology event and fair of the french part of Switzerland.

L'Hebdo magazine, member of the Ringier publishing group, has been very active in the field of Web based content publishing since 1995 both as an on-line supplement to the paper based magazine as well as selected excerpts of the magazine published on their Web server. They have undertaken not less than five different versions of their Web site (Webdo), reaching now over one million hits during the first quarter of 1998. Webdo has now acquired its autonomy as a virtual full edition starting this fall. Early 1996, contacts were established between the Object Systems Group of the University of Geneva - CUI, and L'Hebdo to study the opportunities of content commercialization over open networks such as the Internet while enforcing copyright control. These initial discussion led to the HyperNews project supported by the Swiss Priority Program for Information and Communication Structures (SPP-ICS 5003-45333), launched in June 1996, where L'Hebdo is our industrial partner.

In the scope of this collaboration, L'Hebdo has hosted the HyperNews team on their stand at Computer'98 to demonstrate the HyperNews prototype. Announcement of this event was published in L'Hebdo magazine of April 16, 1998 and an article presenting the project [114] appeared in the Webdo Mag, issue number 4, April 1998 distributed freely throughout the exhibition (also distributed as a supplement of L'Hebdo magazine of April 23, 1998). Moreover for the occasion a project description flyer was kindly printed by L'Hebdo for distribution on the stand during the exhibition.

Demonstration Scenario

For the purpose of the demonstration and to illustrate the "cross information-source" aspect of HyperNews, three different information sources were used. Namely: L'Hebdo for which all the articles of 1998 were available as HyperNews articles, *L'Illustré*, another magazine published by the Ringier publishing group for which several articles were also available, and a third "fake" information provider, named D.P. was used for the sake of the demonstration. All three information providers were running remotely on different hosts at the University of Geneva, respectively on a Sparc Ultra 2, a Sparc Ultra 1 and a Sparc station 4.

From the point of view of the information consumers, three different users were available. The first one (User 1), a Sparc Ultra 1, running on the site of the demonstration. A second one (User 2) was running remotely on a host in GM D (Bonn) and a third one (User 3) was hosted by the University of Geneva. This configuration is summarized in Figure 5.0.

The demonstrator allowed for full customization of the environment. Namely, to define new contexts together with their corresponding information and presentation profiles.

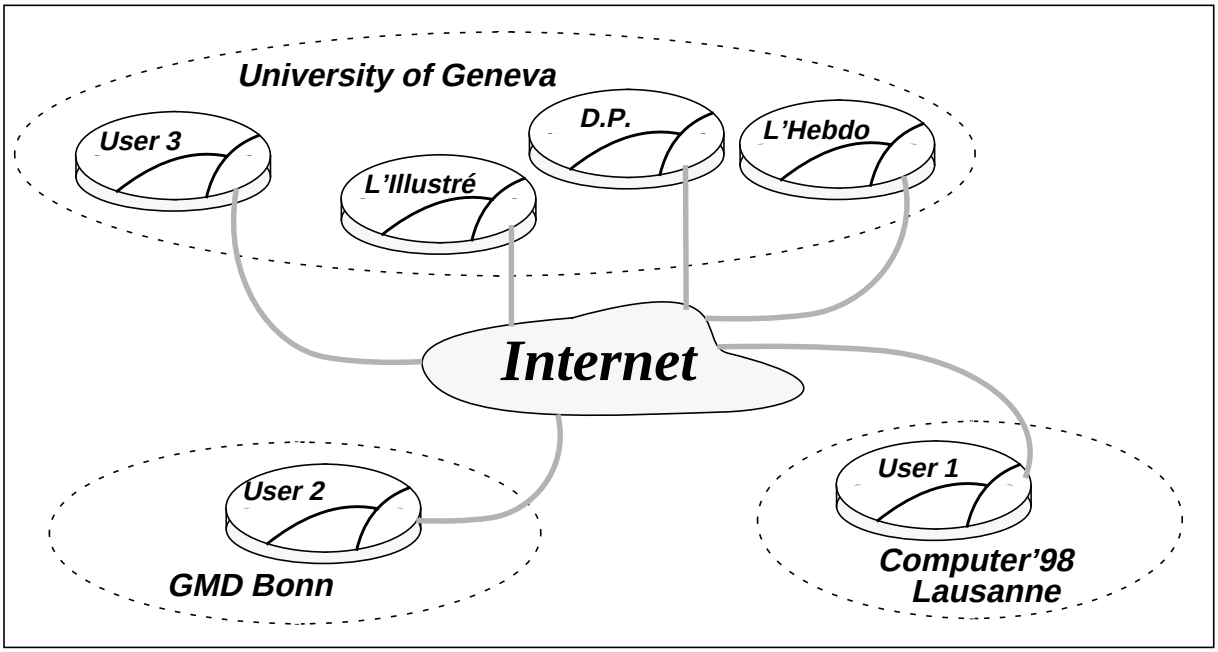


Figure 5.0 HyperNews demonstration configuration map

Feedback and Reactions

These four days of demonstration have been very successful. Many demonstrations of the HyperNews prototype were given throughout the Computer'98 event. Generally speaking, the most recurring comments and feedback received were the following:

- Very strong interest regarding the issues addressed by the HyperNews project
- Very high relevance of the project towards commercially viable value added services
- Strong potential of the approach (standard) bound however to wide spread, general acceptance (i.e., it is unrealistic to use such a service only for one or two information providers).
- Very good understanding and acceptance of the fact that our approach is not in competition with "traditional Web/Internet content and services" but rather in the field of trusted and copyrighted value added services for which a fee could be required depending on the commercial policy of their providers.
- From potential users point of view, this is exactly the type of service they have been waiting for ever since they started "surfing" on the Internet in order to cope with all this overwhelming information as well as the burden of navigating and managing as many passwords and accounts as the number of information sources they are interested in. However, here again it is very unlikely that a user would download and install such an environment only to access a couple of information sources.
- Finally, from people who were familiar with the project, either through technical reports or discussions, seeing the resulting prototype made everything clear about the project and its objectives.

In summary, this opportunity we have been given to demonstrate the results of our work through a demonstrator within a public event was very useful both to collect feedback, reactions and to gain visibility for the project and the addressed issues.

5.9.3 Other Dissemination Means

Following the Computer'98 exhibition, a radio interview was made about the HyperNews project in the "Village Global" [115] program of the Radio Suisse Romande La Premiere on May 16, 1998. The roughly 15 minutes interview was a very general description of the Hyper-News project presenting the issues and goals of the project targeted to a non technical audience.

Similarly, a newspaper article about the project was published in the June 10, 1998 issue of the Tribune de Genève [116], in a column dedicated to research issues and achievements of Ph.D. candidates at the University of Geneva.

Chapter Six | *Conclusions*

This thesis addresses the issue of commercial electronic publishing over open networks. Such open networks today materialize in the form of the Internet. However they are likely to evolve and change over time but the conceptual notion of open networks with their inherent characteristics of being global, untrusted and insecure will remain.

In this context of open networks, the field of publishing, although mature, is undergoing profound changes initiated and emphasized by the advent of the World-Wide Web and its almost immediate acceptance by the industry. However, the publishing industry is still reluctant to publish their content electronically despite the fact that most of this content is already in electronic form. The reason for this holds in the low or almost inexistent means to protect their copyright and intellectual property thus hampering commercial electronic publishing from harvesting the tremendous returns that are expected. During the last few years, different approaches and solutions have been proposed targeting commercial electronic publishing of documents. However most of them have major limitations with respect to issues such as copyright protection, privacy, anonymity, flexibility of distribution models, freedom of choice with respect to commercial partners and electronic payment schemes, dynamics of author rights properties linked to documents, content type and formats or applications to which these documents are intended for, etc.

The approach proposed in this thesis aims at providing a framework for commercial electronic publishing over open networks based on the mobile object (agent) paradigm, thus providing a common abstraction shared by classes of electronic publishing applications. Such an approach should make it possible to take into account the needs that are common to the field of electronic publishing in a commercial global networked market which is competitive and untrusted by essence. The use of mobile objects or agents in this context represents a new and original direction whose major advantage is to give a total autonomy to the electronic document while guaranteeing and enforcing a “self-controlled” safe and secure usage of its content at all times. Such an infrastructure should consequently support and encourage the commercial exchange and dissemination (i.e., superdistribution) of electronic documents in the broad sense on global information networks such as the Internet.

6.1 Contributions

Towards this goal we first identified a set of requirements that should be met by such an abstraction. These requirements have been considered from the point of view of its different actors

(i.e., the information consumers and providers), the electronic goods and the infrastructure enabling their trade in a distributed electronic market.

Based on these requirements, we next designed a framework called Hep based on mobile objects or agents. The Hep framework provides an abstraction to the application layer for implementing commercial electronic publishing applications. Within this framework we designed a scheme for the secure commercial distribution of electronic documents that supports the superdistribution model. Documents are securely encapsulated in mobile agents together with the programs that control access to their content. The resulting mobile document agent is thus responsible for its own security and can be held, copied and distributed freely without infringing any copyright or intellectual property right since access to its content is bound to successful execution of the attached policy (i.e., payment, usage, etc.).

The major contribution of this work is that we have shown that secure commercial dissemination of electronic documents is possible today. This claim is further supported by the implementation of the Hep based HyperNews electronic newspaper environment prototype. Furthermore, we have shown that our approach is open and flexible enough to accommodate different and diverse business models and user requirements.

6.2 Open Issues

There are three major open issues within this work. First, regarding long lasting documents that are likely to be kept for an indefinite time, second concerning the off-line operations when accessing documents and third security with respect to code.

Long lasting documents are a real issue as security is bound to current technology. The problem is that unlike newspaper or magazine articles having the property of being of low value and short lifetime, documents such as books will exist for decades. Releasing them today encrypted with current encryption technology provides security requiring decades of computation to break. However a couple of technological r-evolutions in the future could well bring sufficient processing power to home computers and thus offer anybody the necessary power to break today’s encryption algorithms within a few minutes.

A possible solution could be to have the document key be a combination of a partial key that would be included in the document whereas the other part would be derived from an index to a shared secret between the credit institution and the information provider. Thus in order to recover the document key, the credit institution would have to lookup the shared index according to the index and combine the result with the first part. As a result, breaking the scheme would involve many simultaneous disclosures of keys and shared secrets which is less likely to occur than a single key. Moreover, breaking the security of a document agent would reveal an unusable document key as it would only be a partial key. This approach cannot prevent cryptanalytic attacks on the ciphertext. It would also involve significant key management for the distribution of the shared secrets.

An other way, which could be combined with the previous solution, could be to have only part of the content in the document agent and the rest acquired dynamically upon content access. However, this approach has major drawbacks in that the document agent is not any more self

contained in the superdistribution sense and content access becomes bound to being on-line. All these issues require careful attention as they represent trade-offs between security and ease of use or flexibility. Long lasting documents are also bound to face other problems concerning data type and software evolution, thus requiring downwards compatibility with respect to document formats as well as document agent.

Still in the field of security, code security is becoming an important issue. The reason being that as mobile code is emerging and becoming a popular paradigm we will be soon highly concerned about code security. In a similar way to data security, code security is also concerned by integrity, authoritativeness and confidentiality: Integrity to guarantee that code has not been tampered with, modified in any way and conform to its original. Authoritativeness, to guarantee that the code is genuine and that it was really created by the claimed origin. Confidentiality to hide code in a way that makes it hard to reverse engineer. Some interesting work was done recently in this field with code obfuscation (control flow [118] and data structures [119]) and software watermarking [120]. The code obfuscation approach consists of transforming (using an obfuscator) a program into another program which is functionally identical. The goal is to make the obfuscated code very difficult to understand and reverse engineer. With software watermarking the attempt is to embed a large number in a program. This embedding process should allow easy retrieval after program transformation, be imperceptible to an adversary and should not degrade the performance of the program. Such techniques definitely need to be included in mobile agent environments.

Concerning off-line operation, we have discussed some possible solutions based on smart-card technology within this work. However the document distribution model requires that the smart-card holds highly sensitive information about the credit institutions as it is intended to undertake the role of the credit institution. Namely, it would require to store a private key of the credit institution on the smart-card. Although possible, this solution is not really desirable as it defeats the purpose of such a key to remain private. Not that the smart-card reader will reveal the key but the sole fact of having this key on the card will trigger the will to hack it. These issues are currently being investigated within the continuation of the HyperNews project together with another project called SmartPay.

6.3 Research Directions

There are a number of directions that could be further investigated from a research point of view at both applied and theoretical levels. But before going into further details, it is worth mentioning that many research fields are more and more interdependent. Taking an example which is directly linked to the subject of this thesis, the field of computer science now needs to interact not only within the many aspects of its own discipline but also with law, economics, public policy, ethics, etc. Achievements in any of these fields having impact on the others. Therefore, we should keep in mind that we are bound to communicate with each other towards common goals.

Commercial real time stream services are likely to become common in a near future as bandwidth availability increases. Such services will be needed both for the corporate users in business and home users in the entertainment industry. Example of such services include, but are not limited to, training and education, multimedia communication, cooperation, music and

video on demand, customer care, health care, etc. As such the market itself will be the major driving force. The need is there and the market is only waiting for the right technology. Although the approach described in this work does not prevent from packaging streams into agents, they are however to be considered as off-line stream services, in the sense that all the content is included (i.e., pre-packaged) within the agent. The issue raised here focuses on the real-time connection oriented stream service requiring to establish secure connections with service providers. In this respect, the agent would only hold the means to initiate and mediate such commercial stream services and not the content it-self. Such real time stream services represent a different class of electronic publishing application for which the Hep framework could offer an interesting testbed infrastructure.

As with many technologies which are neither wide spread nor standardized, different competing approaches are bound to exist simultaneously. The field of mobile agents suffers from this and as a result many different agent environments are currently available, each of which address specific issues. Agent environment interoperability could be identified as a research direction to study the issues and design a common abstraction for mobile agents, thus allowing agent environments to interoperate in the same way as “Objects” interoperate on platforms like the OMG CORBA [124]. Similarly, we can state that the problem of electronic publishing environment or infrastructure interoperation follows the same direction. The Hep framework described in this thesis is an approach to the content commercialization issue over open networks. However, it is unrealistic to assume that information consumers will be willing to install and use many different environments each satisfying similar requirements but which are not interoperable due to proprietary approaches. This would simply defeat the purpose and to illustrate this issue, it would be like having different Web browsers for different Web servers or even worse, like having different CD players for different music production companies. Thus, work in the direction of standardization or interoperation of electronic publishing models is needed keeping in mind that the key factors of success from a user’s standpoint are simplicity and transparence.

From a more theoretical point of view, the field of electronic commerce is lacking a way to express and thus evaluate in a formal way electronic commerce agreements. This issue is rather delicate to address since no common and agreed upon legal framework exists yet. However research in the direction of formal languages to express basic commercial terms and conditions for electronic commerce is likely to be needed in a not so distant future. Being able to express basic commercial and business patterns in a formal way could offer significant advantages, for example, in detecting fraudulent or illegal agreements prior to their execution. In a similar direction, research is currently being done to find ways to formally specify digital property rights through languages (e.g., DPRL work currently under way at Xerox PARC, STAR Lab of InterTrust is also working in this directions). Given such a formal framework, it would then be possible to build a whole set of tools for the creation, management and use of such formally specified electronic terms and conditions for electronic commerce. Moreover, given the necessary cryptographic means to enforce authoritativeness and untampering, these could then serve as evidence in a court of law for dispute handling. This could even go one step further towards on-line dispute handling provided the third party that is to handle the dispute is trusted and has authority to settle the disputes.

In fact trying to put some knowledge and decision capabilities in mobile agents towards an *avatar* metaphor could create new heuristics; as electronic commerce is seeking new patterns for doing business over open networks it is likely that the need for some sort of "*intelligent shopping agent*" will emerge. The common example for such a pattern would be to have an agent sent out to inquire and possibly book a set of individual transactions as when planing a trip. In this example the user would need a plane ticket, a hotel, a car rental and a ticket for the opera. Either the transaction is committed as a whole or not at all. This is typically a multi party transaction involving decision making, bargaining and negotiation. However this raises significant concerns with respect to security and more specifically privacy. In fact one would not want such an agent to hold sensitive information such as available budget, competitor's bids, etc. as snooping them would be very tempting. A promising direction towards this goal could be to investigate the possibilities to split agents into different pieces. Some of these agents would not be mobile but hold all the private information and sensitive data, and thus be able to decide and take actions while the others would be task oriented mobile agents tied to their creator. Taking the example above would lead to the following situation: the deciding agent would launch a number of task agents on the network each of which would only hold information to the extent of what the user is willing to reveal (e.g., object of the transaction, maximum price). Each of these task agents would then collect bids and communicate them securely or bring them back to its creator before moving on to the next bid. In this situation the mobile agent acts as a remote peer screening the privacy of the user.

Finally, within this work we have become aware that tamper proofing in general is difficult to reach and not desired in many business related areas above given levels as this would involve disproportionate efforts compared to the marginal security gains it would provide. These levels and costs can be calculated in order to draw the limit of what is often called an "acceptable level of risk" and thus determine the desired security level. The credit card companies provide one of the best examples of this approach. Nothing can be claimed to guarantee 100% security. However tamper resistance is the goal in the sense it aims at giving the hardest possible time to attackers. Achieving security through software is weaker than through hardware. Thus tamper resistant hardware devices are promised a very successful future as security requirements grow. This is a promising research direction to provide the devices that will strive to enforce security such as a Copyright Chip for example as argued by Mark Stefik in [27].

6.4 Concluding Remarks and Discussion

In summary, Electronic Publishing represents far more than what is currently expected from this field. It is a profound revolution of the information system in its entirety where there is no such thing as static information but rather a notion of real time evolving interconnected information. Information is itself augmented by attributes reflecting properties and relationships. Information is not the raw material any more but rather data generated dynamically by interconnected processes.

We are in the midst of creating the world's first truly global electronic information economy. As such, we are building tomorrow's electronic information society. In order to succeed, it must be truly competitive and open to all its potential market actors in a fair and efficient way.

6.4.1 Electronic-”You-Name-It” or a New Dimensions of “You-Name-It”?

Global open networks have enabled and brought to life the electronic dimension of many existing and relatively ancient activities such as publishing, commerce, trading, auctioning, communication, education, etc.

The term “Electronic” as a qualifier of publishing and commerce for example is likely to fall into disuse. Electronic publishing and commerce are not new industries but new dimensions; natural evolutions of existing industries enabled by the advent of global open networks such as the Internet. However, if it stays, it will only signify the means rather than a field per se, thus allowing to distinguish tangible (i.e., made of atoms) from digital (i.e., made of bits) publishing and trade. To illustrate this issue, consider for example an individual buying a book in two different situations: (i) through the network and (ii) in a shop. In the first case, upon finding the desired book and requesting to purchase it, the user could be prompted to answer the following two questions: First regarding delivery, electronic or hard copy? Secondly regarding payment, invoice or direct electronic payment? In the second situation, in a shop, the seller would also ask whether the buyer wants the hard copy version or the electronic version either sent by e-mail or downloaded to a PDA[1], a Xerox Gyricon Display [121] based device, or other similar electronic book metaphor technology such as the Acorn NewsPAD [122], the Softbook or the Rocketbook [123]. Then the buyer would be asked for the payment method such as cash, credit, usage based in case of electronic version of the document or electronic payment. As we can see, the electronic dimension only enriches existing paradigms and enable new heuristics.

We are not saying here that electronic publishing and commerce do not raise significant and important issues that have been or will have to be addressed by the research community in very diverse domains and fields. But only that Electronic-”you name it” does not constitute new fields but merely enriches our environment with a new dimension which is not the least.

6.4.2 The Road Ahead

It is not our purpose to draw speculations on the future nor to foresee it, as such a task is bound to the use of a “crystal ball”. History has shown that predictions in our field tend to never materialize as anticipated. However, we would like to state some general assumptions on which we rely in the frame of this work.

- *A global digital market* is likely to emerge. This assumption is commonly agreed upon and the Web has proven the demand for such a market. Moreover, this assumption is also supported by the massive investments done in the field for which huge return is almost unanimously expected.
- *Copyright and Intellectual Property Rights* are most unlikely to simply disappear and not be needed any more, thus leading to this “free information” vision some people share. The stakes are too important, not only on a financial basis but most of all not to destroy the incentive under which creation and creativity can be expressed for the benefit of mankind.

1. Portable Digital Assistant or Hand-Held computer

- *Free information* will continue to exist. In this respect the Web has fulfilled the need for a global forum of free speech where anybody can express almost anything. It does not compete nor is it in contradiction with Copyright and Intellectual Property Rights. In fact, it is not because information is free that its recipients should not be guarantied that it is genuine. Furthermore, it is not because information is free that its creators are not entitled to be acknowledged for their work. Nor does it mean that they can be deprived of their authorship or even see their original work transformed, modified without anything to say. Free Web based information and copyrighted information are fully complementary. They serve different purposes, different markets and have a promising digital future coexisting in a peaceful information society.
- *Technology as a service* is part of the equation in finding an equilibrium in this momentum towards the digital information market. The other parameters being public policy, law and privacy. It is too early to set the equation in clear, however these equally important facets of the problem will have to help each other rather than fight. What we can reasonably assume in this context is that technology has a role to play, and it is most likely to be in the field of security as a service to help enforce its other components. In this respect, cryptography and tamper resistant devices (e.g., Copyright chips, smart-card readers, etc.) will be very important. Moreover, other technical aspects of expressing digital equivalents to contractual terms and conditions are likely to play an equally important role (e.g., fair use, first sale rights, etc.). Digital watermarking techniques also have a role to play upon releasing copyrighted content in the clear in our world made of atoms, be it audio, video, images or plain text.
- *Bandwidth and networking* progress will be needed as the digital information market expands. The needs in performance for both corporate and domestic users and the potential of the market will be the driving forces for such progress.
- *Value added services* such as brokerage services and other intermediary services in the digital market will emerge and become more and more important as the market expands. Such services will require interoperability, architecture independence, network aware pieces of software having the property of being able to move around the network.
- *The frontier* between an intranet - behind which a host is considered “safe” - and the Internet will fade in favor of a network of interconnected untrusted hosts. Each of which having the security requirements found in today’s corporate intranets (e.g., firewalls). Although the intranet is likely to survive as a service to the corporate user, the security aspects will be moved upstream towards the host.

6.4.3 Closing Remark

There is a major open issue in our domain: Security. More specifically, security in a temporal perspective. Let us take the simple example of a book, which we buy and preciously store on a shelf in our living room or office. This book, made of atoms, can stay there for decades or even centuries and is bound to existing laws and copyrights. However, considering the digital equivalent of the same book raises significant questions if for example it is encrypted with current technology and released throughout the world. What technology will be available ten years from

now? History has shown that ten years can be even more than a generation in technology. Who would dare risk today to claim at least ten years cryptographic security? Most probably no one. The best that can be claimed today is bound to current technology and knowledge. Or to use the familiar limitation found in the field of economics: *Ceteris Paribus*[1]. But technology has proven that such assumptions do not resist to time in our field of computer science.

1. *All other things being equal.*

References

[1] Raj Reddy, “The Universal Library: Intelligent Agents and Information on Demand”, *Forum on Research & Technology Advances in Digital Libraries*, May 15-17, 1995, McLean, Virginia, Springer-Verlag, LNCS 1082, 1996.

[2] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Frystyk Nielsen and A. Secret, “The World Wide Web”, *Communications of ACM*, Vol. 37, No 8, August 1994, pp. 76-82.

[3] The National Center for Supercomputing Applications, “A Beginner's Guide to HTML”, http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

[4] T. Berners-Lee, R. Fielding and H. Frystyk, “Hypertext Transfer Protocol-HTTP/1.0”, *RFC 1945*, May 1996.

[5] Vannevar Bush, “As we may think”, Atlantic Monthly, July 1945

[6] R. Mori and M. Kawahara, “Superdistribution: The Concept and the Architecture”, *Transaction of the IEICE*, Vol. E 73, no. 7, July 1990, pp. 1133-1146.

[7] R. Mori and S. Tashiro, “The Concept of Software Service System (SSS)”, *Transaction of the IEICE*, J70-D.1, Jan 1987, pp. 70-81.

[8] Brad Cox, “Superdistribution”, *Wired Magazine*, September 1994, pp 89.92.

[9] Brad Cox, “Superdistribution Objects as Property on the Electronic Frontier”, Addison-Wesley, 1996.

[10] Kaplan M.A., “IBM Cryptolopes^{TM}, SuperDistribution and Digital Rights Management”, IBM Corporation, December 1996, http://www.research.ibm.com/people/k/kaplan

[11] Kohl U., Lotspiech J. and Kaplan A., 1997, “Safeguarding Digital Library Contents and Users Protecting Documents Rather Than Channels”, IBM Research Division San Jose, California, and Hawthorne, New York, *D-Lib Magazine*, September 1997, http://www.dlib.org/dlib/september97/ibm/09lotspiech.html

[12] O. Sibert, D. Bernstein and D. Van Wie, “The DigiBox: A Self-Protecting Container for Information Commerce”, proceedings of *First USENIX Workshop on Electronic Commerce*, New-York, July 11-12, 1995, pp 171-189.

[13] D. Van Wie, O. Sibert and J. Horning, “Panel on the InterTrust Commerce Architecture”, *20th National Information Systems Security Conference*, Oct. 7-10, 1997, Baltimore.

[14] Softlock Services Inc., *SoftLock*, http://www.softlock.com/

[15] Open Market Inc., *Folio 4*, http://www.folio.com/

[16] STAR Lab, InterTrust Technologies Corporation, http://www.star-lab.com/

[17] Schneier Bruce, “Applied Cryptography: protocols, algorithms, and source code in C”, Second Edition, John Wiley & Sons, Inc,1996.

[18] International DOI Foundation, “ The Digital Object Identifier System”, http://www.doi.org/

[19] Christopher Brown-Humes, “NATWEST: Bank makes alliance with US technology group”, *Financial Times*, Sept 7, 1998.

[20] Rights Exchange Inc., “Rights Exchange”, http://www.rightsxchange.com/

[21] SoftLock Services Inc., “SoftLock”, http://www.softlock.com/

[22] Breaker Technologies Ltd, “SoftSEAL”, http://www.softseal.com/

[23] Rainer Mauth, “Better Copyright Protection”, BYTE Magazine, May 1998, pp. 5-6.

[24] Open Market Inc., “Folio 4 Family of Products”, http://www.folio.com/products/folio/

[25] World Intellectual Property Organization, WIPO, “Berne Convention for the Protection of Literary And Artistic Works of September 9, 1886”, http://www.wipo.org/

[26] World Intellectual Property Organization, WIPO, “WIPO Copyright Treaty”, December 20, 1996

[27] Charles C. Mann, “Who Will Own Your Next Good Idea”, The Atlantlic Monthly, September 1998.

[28] Charles C. Mann, Mark Stefik, Lawrence Lessig and John Perry Barlow, “Life, Liberty, and the pursuite of Copyright?”, Roundtable, The Atlantlic Unbound, September 1998.

[29] Pamela Samuelson, “Intellectual Property and the Information Society”, *Plenary Lecture*, HICSS-31, Hawai International Conference on System Sciences, Kona, Hawaii, January 6-9, 1998.

[30] Mark Stefik, “DPRL: The Digital Property Rights Language”, Xerox Palo Alto Research Center, CA.

[31] S. Glassman, M. Manasse, M. Abadi, P. Gauthier, P. Sobalvarro, “*The Millicent Protocol for Inexpensive Electronic Commerce*”, *Fourth International World Wide Web Conference*, Boston, December 11-14, 1995.

[32] B. Cox, J.D. Tygar, M. Sibru, “*NetBill Security and Transaction Protocol*”, proceedings of *First Usenix Workshop on Electronic Commerce*, New-York, July 11-12, 1995

[33] D. Chaum, “*Achieving Electronic Privacy*”, Scientific American, August 1992, p. 96-101

[34] CyberCash, Inc., “*CyberCash White Papers*”, http://www.cybercash.com/cybercash/wp/whitepapers.html

[35] Net1, Inc., “*The NetChex system*”, http://www.netchex.com:80/

[36] J.-P. Boly, A. Bosselaers, R. Cramer, R. Michelsen, S. Mjolsnes, F. Muller, T. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallee, M. Waidner, “*The ESPRIT Project CAFE, High Security Digital Payment Systems*”, *Third European Symposium on Research in Computer Security*, LNCS 875, Springer-Verlag, Berlin 1994, p. 217-230

[37] IBM, “*Internet Keyed Payment Protocols (iKP)*”, http://www.zurich.ibm.ch/Technology/Security/extern/ecommerce/iKP.html

[38] VISA, MasterCard, “*Secure Electronic Transactions, New draft as of 8/7/96*”, http://www.visa.com/cgi-bin/vee/sf/standard.htm, http://www.mastercard.com/set/set.html

[39] D. Naccache, D. M’Raihi, “Cryptographic Smart Cards”, IEEE Micro, Volume 16, Number 3, June 1996.

[40] J.-F. Dhem, D. Veithen, J.-J. Quisquater, “SCALPS: Smart Card for Limited Payment Systems”, IEEE Micro, Volume 16, Number 3, June 1996.

[41] Sun Microsystems, Inc, “Java Card Technology”, *Home Page*, http://java.sun.com/products/javacard/

[42] Mondex, http://www.mondex.com/

[43] R.L. Rivest, A. Shamir, and L.M. Adelman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems”, in *Communications of the ACM*, Vol. 21, n. 2, Feb 1978, pp. 120-126.

[44] R.L. Rivest, A. Shamir, and L.M. Adelman, “On Digital Signatures and Public Key Cryptosystems”, MIT Laboratory for Computer Science, Technical Report 212, 1979.

[45] Sun Microsystems, “Java Commerce Home Page”, http://java.sun.com/products/commerce/

[46] Java Soft, “*White Paper: The Java Electronic Commerce Framework (JECF)*”, http://www.javasoft.com/products/commerce/doc.white_paper.html

[47] Sun Microsystems, “Java Wallet Documentation”, http://java.sun.com/products/commerce/docs/index.html

[48] S. Ketchpel et al., “U-PAI: The Stanford Universal Payment Application Interface”, in proceedings of Second *USENIX Workshop on Electronic Commerce*., Oakland, California, Nov. 18-21, 1996, pp. 105-121.

[49] Jose L. Abad-Peiro, N. Asokan, Michael Steiner, Michael Waidner, “Designing a Generic Payment Service”, *IBM Syst. Journal*, Vol 37, No. 1, January 1998.

[50] SEMPER Final Project Report, Lecture Notes in Computer Science, Springer-Verlag, to appear in 1999.

[51] Courrier International, http://www.courrierint.com/

[52] J. Vitek and C. Tschudin (Eds.), “Mobile Object Systems: Towards the Programmable Internet“, Springer-Verlag, LNCS 1222, 1997.

[53] K. Rothermel and R. Popescu-Zeletin (Eds.), “Mobile Agents”, Springer-Verlag, LNCS 1219, 1997.

[54] J. White, "*Mobile Agents*", General Magic White Paper, General Magic, Inc. 1996.

[55] A. Black, N. Hutchinson, E. Jul, and H. Levy, "Fine-grained mobility in the Emerald system", *ACMTransactions on Computer Systems*, 6(1), 1998 pp. 109-133.

[56] General Magic, "*Agent Technology: Odyssey*", General Magic, Inc., http://www.generalmagic.com/technology/odyssey.html

[57] K. A. Bharat and L. Cardelli, "Migratory applications", in *Mobile Object Systems: Towards the programmable Internet*, pp 131-149, Springer-Verlag, LNCS 1222, 1997.

[58] R.S. Gray, D. Kotz, G. Cybenko and D. Rus, "D'Agents: Security in a Multiple-Language, Mobile-Agent System", in *Mobile Agents and Security*, pp 154-187, Springer-Verlag, LNCS 1419, 1998.

[59] IBM Research, "*IBM Aglets Workbench - Home Page: Programming Mobile Agents in Java*", http://www.trl.ibm.co.jp/aglets/

[60] Rowan Dordick, "The Secrets of Agents", *IBM Research magazine*, 1997, Issue 1, http://www.research.ibm.com/resources/magazine/1997/issue_1/agents197.html

[61] Object Space Inc., "ObjectSpave Voyager Core Technology", Object Space Inc., http://www.objectspace.com/products/voyager/core/index.html

[62] Markus Strasser, Joachim Baumann and Fritz Hohl, "*Mole - A Java Based Mobile Agent System*", *Second ECOOP Workshop on Mobile Object Systems*, University of Linz, July 8-9, 1996.

[63] J. Baumann, F. Hohl, K. Rothermel, M. Schwehm, M. Straßer, "Mole 3.0: A Middleware for Java-Based Mobile Software Agents", to appear in Proceedings Middleware'98, Chapman & Hall, 1998.

[64] J. Vitek, M. Serrano and D. Thanos, "Security and Communication in Moblie Object Systems", in *Mobile Object Systems: Towards the programmable Internet*, pp 177-199, Springer-Verlag, LNCS 1222, 1997.

[65] Giovanni Vigna (Ed.), "Mobile Agents and Security", Springer-Verlag, LNCS 1419, 1998.

[66] J. Vitek, C. Bryce and W. Binder, "Designing JavaSeal or How to Make Java Safe for Agents", in *Electronic Commerce Objects*, Technical Report, 1998, D. Tsichritzis (ed.), pp 105-126.

[67] Ralf Hauser, Gene Tsudik, "On Shopping Incognito", in *Proceedings of The Second USENIX Workshop on Electronic Commerce*, pp 251-257, Oakland, California, November 18-21 1996.

[68] J. Ebersole, "Protecting intellectual property rights on the information superhighways", *International Publishers Association Bulletin*, Volume X, No. 3, 1994, pp. 3-4.

[69] Pamela Samuelson, "The Copyright Grab", Wired Magazine, Issue 4.01, January 1996.

[70] N. Asokan, "Fairness in Electronic Commerce", Ph.D thesis, University of Waterloo, Ontario, Canada, 1998.

[71] Ken Arnold and James Gosling, "The Java Programming Language, Second Edition", *The Java Series*, Addison-Wesley, 1998.

[72] James Gosling and Henry McGilton, "The Java Language Environment, A White Paper", Sun Microsystems, Inc., 1996

[73] James Gosling, Bill Joy and Guy Steel, "The Java Language Specification", *The Java Series*, Addison-Wesley, May 1996, http://www.javasoft.com/docs/white/langenv/

[74] Sun Microsystems, "Java Technology Home Page", Sun Microsystems, Inc, http://java.sun.com/

[75] Tim Lindholm and Frank Yellin, "The Java Virtual Machine Specification", *The Java Seies*, Addison-Wesley, 1997.

[76] P. Madany, "JavaOS: A Standalone Java Environment", *White Paper*, Sun Microsystems, Inc., 1997, http://java.sun.com/products/javaos/javaos.white.html

[77] Sun Microsystems, "JavaStation - An Overview", *White Paper*, Sun Microsystems, Inc., 1996, http://www.sun.com/nc/whitepapers/javastation/javast_ch1.html

[78] Sun Microsystems, "picoJava I Microprocessor Core Architecture", *White Paper*, Sun Microsystems, Inc., 1996, http://www.sun.com/sparc/whitepapers/wpr-0014-01/

[79] M. Mira da Silva and A. Rodrigues da Silva, "Insisting on Persistent Mobile Agent Systems", in *Mobile Agents, Proceedings of First International Workshop, MA'97*, K. Rothermel and R. Popescu-Zeletin (Eds.), Springer-Verlag, LNCS 1219, 1997, pp. 174-185.

[80] Jean-Henry Morin , ``HyperNews: a Hypermedia Electronic-Newspaper Environment Based on Agents", in Proceedings of HICSS-31, Hawaii International Conference On System Sciences, IEEE 1998, January 6-9, 1998, Kona, Hawaii, Volume II, pp 58-67.

[81] Jean-Henry Morin and Dimitri Konstantas, “HyperNews: A MEDIA Application for the Commercialization of an Electronic Newspaper”, in *Proceedings of SAC’98, 1998 ACM Symposium on Applied Computing*, Atlanta, Georgia, February 27 - March 1, 1998, pp. 696-705.

[82] Dimitri Konstantas, Jean-Henry Morin and Jan Vitek, “MEDIA : A Platform for the Commercialization of Electronic Documents”, in *Object Applications*, Ed. Denis Tsichritzis, CUI, University of Geneva, 1996.

[83] Jean-HenryMorin and Dimitri Konstantas, “Towards Hypermedia Electronic Publishing”, Proceedings of *second IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications*, Stanford, California, August 7-9 1995.

[84] Prevelakis V., Konstantas D. and Morin J-H., 1997, “Issues for the Commercial Distribution of Electronic Documents”, in *Communications and Multimedia Security, CMS‘97*, Volume 3, S. Katsikas (Ed.), Joint Working Conference IFIP TC-6 and IFIP TC-11, 22-23 September, 1997, Athens, Greece, pp. 265-276.

[85] Krall A. and Vitek J., 1997, “On Extending Java” in *Proceedings of the Joint Modular Languages Conference, JMLC'97*, Linz, Austria, Springer-Verlag, March 1997.

[86] Rauber C., Ruanaidh J. Ó and Pun T., “Secure distribution of watermarked images for a digital library of ancient papers”, in *Second ACM Conf. on Digital Libraries*, Philadelphia, PA, July 23-26, 1997.

[87] World-Wide Web site of L’Hebdo magazine, http://www.webdo.ch/

[88] Jean-Henry Morin, “Requirements for a Hypermedia Electronic-Newspaper Environment based on Agents”, in *Object at Large*, Ed. D. Tsichritzis, CUI, University of Geneva, University of Geneva 1997.

[89] C. R. Watters, F. J. Burkowski and M. Shepherd (Guest Editors) “Special Issue: Electronic News”, in *Information Processing & Management*, vol. 33, No. 5, September 1997, Elsevier Science Ltd.

[90] C. R. Watters, F. J. Burkowski and M. Shepherd, “Introduction to a special issue on electronic news”, in *Information Processing & Management*, vol. 33, No. 5, September 1997, Elsevier Science Ltd., pp. 579-581.

[91] Pascal R. Chesnais, Matthew J. Mucklo, and Jonathan A. Sheena, “*The Fishwrap Personalized News System*”, *Proceedings of the 1995 2nd International Workshop on Community Networking*, pp 275-282, Princeton, NJ, June, 1995.

[92] MIT Media Laboratory, “Fishwrap Documentation center”, http://fishwrap-docs.www.media.mit.edu/docs/

[93] MIT Media Laboratory, “fishWrap - Personalized News System Home Page”, http://fishwrap.mit.edu/

[94] Chronicle Publishing Company, “SF Gate”, http://www.sfgate.com/

[95] Intracom S.A., “HyNoDe - Hypermedia News On Demand Service Site”, http://hynode.intranet.gr/hyn/

[96] Intracom S.A., “Intracom Home Page”, Greece, http://www.intranet.gr/

[97] C. R. Watters, M. A. Shepherd and F. J. Burkowski, “Electronic News Delivery Project”, in *Journal of the American Society for Information Science*, John Wiley & Son, vol. 49, no. 2, February 1998, pp. 134-150.

[98] The Economist, “Pushmepullyou”, The Economist Newspaper Limited, Nov. 16, 1996.

[99] PointCast Inc., *home page*, http://www.pointcast.com/

[100] Sun Microsystems, “The JDBC Database Access API“, Sun Microsystems, Inc, http://java.sun.com/products/jdbc/index.html

[101] J. Steven Fritzinger and Marianne Mueller, “Java Security“, Sun Microsystems, Inc, 1996.

[102] Sun Microsystems, “Secure computing with Java: now and the future“, White Paper, Sun Microsystems, Inc, http://www.javasoft.com/marketing/collateral/security.html

[103] Li Gong, Marianne Mueller, Hemma Prafullchandra and Roland Schemers, “Going Beyond the Sandbox: An Overview of the New Security Architecture in the Java(TM)“, *USENIX Symposium on Internet Technologies and Systems (USITS '97)*, Sun Microsystems, Inc., December 8-11, 1997, Monterey, California.

[104] Sun Microsystems, “Java Beans“, Sun Microsystems, Inc, http://java.sun.com/beans/

[105] Sun Microsystems, “Java Foundation Classes (JFC)“, Sun Microsystems, Inc, http://java.sun.com/products/jfc/index.html

[106] Sun Microsystems, “Java Foundation Classes: now and the future“, White Paper, Sun Microsystems, Inc, http://java.sun.com/marketing/collateral/foundation_classes.html

[107] Sun Microsystems, “The Swing connection“, Sun Microsystems, Inc, http://java.sun.com/products/jfc/swingdoc-current/index.html

[108] Systemics, “The Cryptix cryptographic package“, Systemics Ltd., http://www.systemics.com/software/cryptix-java/

[109] Stingray Software, “Objective Blend Whitepaper“, Stingray Software, http://www.stingsoft.com/obj/whitepaper.asp

[110] Stingray Software, “Objective Grid / Java Whitepaper“, Stingray Software, http://www.stingsoft.com/ogj/whitepaper.asp

[111] Rogue Wave Software Inc., “Stingray Software Products”, Product List, http://www.roguewave.com/company/stingray/products.html

[112] R.L. Rivest, “The MD5 Message Digest Algorithm”, RFC 1321, April 1992.

[113] CCITT, *Recommendation X.509*, “The Directory-Authentication Framework”, CCITT, ITU, Geneva, 1989

[114] Pascal Montjovent, “HyperNews: L’info transparente”, *Webdo Mag* No 4, avril 1998, p 25, http://www.webdo.ch/webdo_mag/webdo_mag_04/hypernews_04.html

[115] Pierre-Philippe Cadert, Jean-Marc Sandoz et Noël Tortajada, “Village Global”, *Radio Suisse Romande La Première*, May 16, 1998, http://www.rsr.ch/LaPremiere/pages/villageglobal.htm

[116] Adrien Bron, “Un Genevois bouleverse les mdias électroniques”, *Tribune de Genève*, June 10, 1998, p 38.

[117] Gianpaolo Cugola, Carlo Ghezzi, Gian Pietro Picco and Giovanni Vigna , “Analyzing Mobile Code Languages“, in Mobile Object Systems, J. Vitek and C. Tschudin (Eds.), Springer-Verlag, LNCS 1222, 1997, p101.

[118] Christian Collberg, Clark Thomborson and Douglas Low, “Breaking Abstractions and Unstructuring Data Structures”, in proceedings of 1998 International Conference onComputer Languages, Chicago, Illinois, May 14-16, 1998.

[119] Christian Collberg, Clark Thomborson and Douglas Low, “Manufacturing Cheap, Resilient, and Stealthy Opaque”, in *proceedings of 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, San Diego, California, January 19-21, 1998.

[120] Christian Collberg and Clark Thomborson, “Software Watermarking: Models and Dynamic Embeddings”, in proceedings of 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Antonio, Texas, January 20-22, 1999.

[121] W. Wayt Gibbs, “The Reinvention of Paper”, *Scientific American*, Sept. 1998, pp22-23.

[122] Acorn Computer Limited, NewsPAD, http://www.acorn.com/acorn/technology/sheets/003-NewsPAD/

[123] JoAnne Robb, “E-Books: The End of the Guttenberg Era?”, *PC World*, Aug. 3, 1998.

[124] Object Management Group, “The Common Object Request Broker: Architecture and Specification”, Revision 2.0 July 1995.

Appendix

Appendix A: Summary of Information Consumer Requirements

Requirement Number	Requirement
IC-1	*information consumers should be able to use any widespread rendering interface as an electronic publishing reader.*
IC-2	*information consumers should be able to choose their information providers.*
IC-3	*information consumers should be able to specify their information interests for every information provider (information profile).*
IC-4	*information consumers should be able to specify the general structure of their electronic publishing application (presentation profile)*
IC-5	*information consumers should be able to specify the content of each structure element of their electronic publishing application (presentation profile)*
IC-6	*information consumers should be able to have multiple instances of an electronic publishing application (information contexts)*
IC-7	*information consumers should be able to be notified when information updates are available on issues of interest (active information)*
IC-8	*information consumers should be able to access easily any point of the historical evolution of an information*
IC-9	*information consumers should be able to access directly any referenced material both inside and outside the domain of the information provider*
IC-10	*information consumers should be able to retain full anonymity when desired*
IC-11	*information consumers should be able to pay on a usage basis*
IC-12	*information consumers should be able to hold and redistribute freely self contained, self secured electronic documents*

Table A 1: Information Consumer Requirements Summary

Requirement Number	Requirement
IC-13	*information consumers should be able to access subsequently documents they have already paid for (provided the policy allows it)*
IC-14	*an electronic publishing system should be open to existing and future electronic commerce systems*
IC-15	*information consumers should be able to access documents off-line*
IC-16	*information consumers should be able to move their electronic publishing environment between hosts easily*
IC-17	*information consumers should be able to publish new documents embedding documents of other information providers together with their own added value*

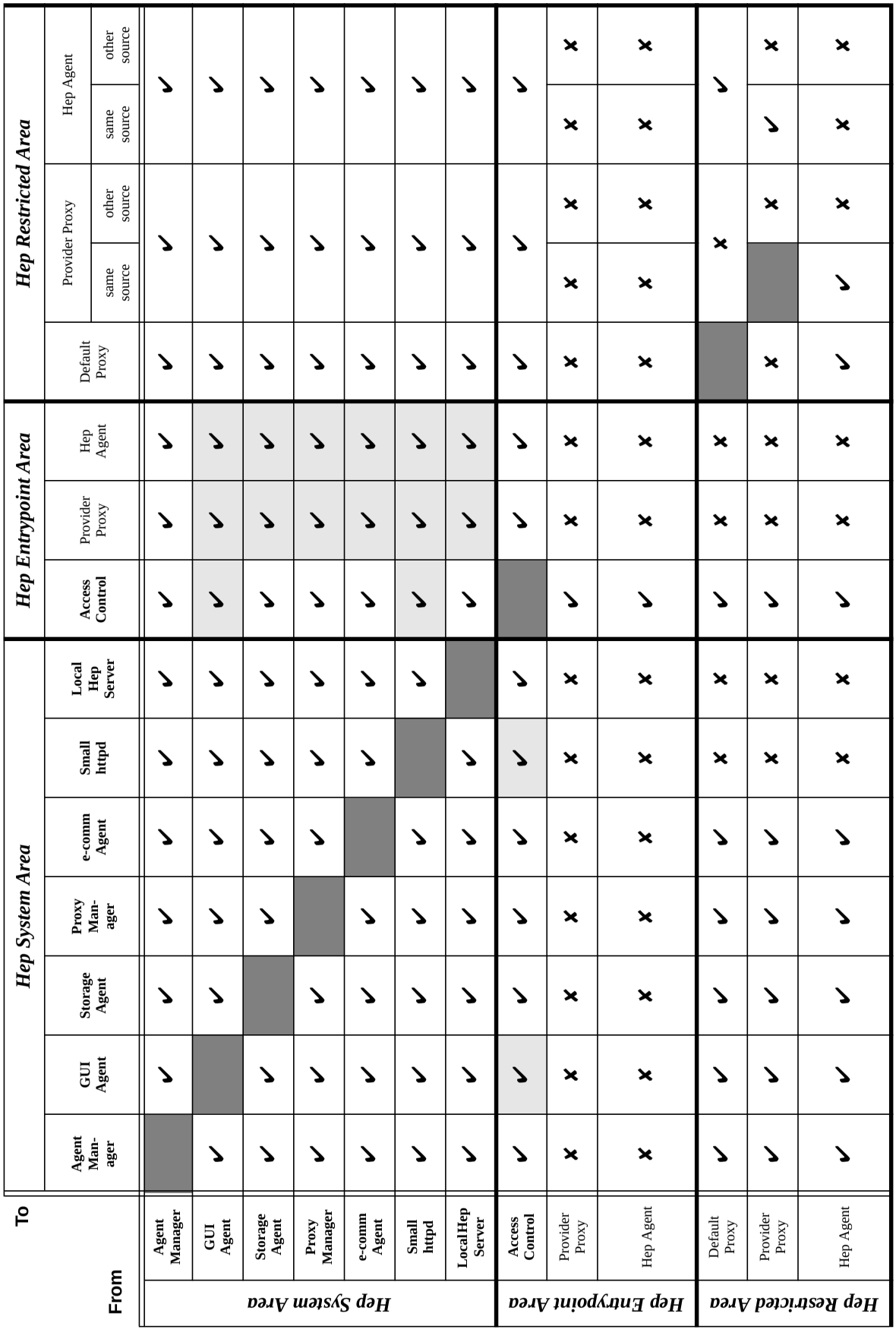
Table A 1: Information Consumer Requirements Summary

Appendix B: Summary of Information Provider Requirements

Requirement Number	Requirement
IP-1	*information providers should be assured that content will be protected and secured against illegal access and use thus protecting their copyrights and intellectual property*
IP-2	*information providers should be assured that content access will generate payment to the copyright holder and thus be able to collect revenue from their electronic publishing activity*
IP-3	*information providers should be able to have their own classification schemes*
IP-4	*information providers should be able to easily transform existing electronic material into autonomous electronic documents*
IP-5	*information providers should be able to establish historical evolution links between content elements*
IP-6	*information providers should be able to insert links in content elements to reference other material inside and outside their own domain*
IP-7	*information providers should rely on a standard widely accepted formatting language for electronic content composition and rendering*
IP-8	*information providers should be able to reflect the editorial process within the electronic content element*
IP-9	*information providers should be able to accommodate and customize policy dependent issues according to their needs*
IP-10	*information providers should strive to offer high service availability to information consumers*

Table B 1: Information Provider Requirements Summary

Appendix C: The Inter-Agent Communication Matrix



The rights are expressed by check marks when the communication can occure or by a cross otherwise. The dark grey shaded boxes denote self communication and the light grey shaded boxes indicate that although communication is authorized it is not pertinent for the time being. The target columns for Provider proxy agents and Hep agents have been split in two columns when necessary. This in order to distinguish inter-agent communication between user agents belonging to the same provider or different providers. This is summarized in the following table:

✔	Authorized
✘	Unauthorized
	Non pertinent
✔	Authorized but not considered in current design
Agent	Bold agent names represent system agents

Table C 1: Legend for the Inter-Agent Communication Matrix

Appendix D: HyperNews Preferences and Utilities

Information Consumer

Figure D.1 (a) : General

Figure D.1 (b) : Reader

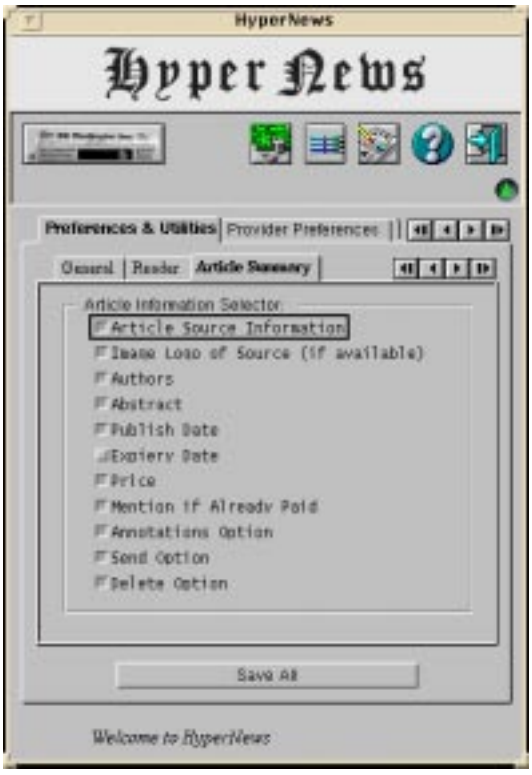


Figure D.1 (c) : Article Summary

Figure D.1 Information Consumer Preferences & Utilities

Information Provider

Figure D.2 (a) : Database settings

Figure D.2 (b) : Classification Manager