HyperNews, a Hypermedia Electronic-Newspaper Environment Based on Agents

Morin, Jean-Henry

HyperNews, a Hypermedia Electronic-Newspaper Environment Based on Agents[1]

Jean-Henry Morin

Abstract

The HyperNews project aims at developing a hypermedia electronic newspaper system, based on agent technology, that will help the readers reduce time spent searching for relevant information and enable information providers to commercialize hypermedia electronic news articles in terms similar to those of printed newspapers. The system enforces copyright control and payment on a "pay per use" basis. Moreover, HyperNews will establish intra- and cross-newspaper hyperlinks according to the interests of the reader and present the information consumer a personalized hypermedia electronic newspaper through a Java-based World-Wide Web browser. The information consumers and providers run an agent execution platform on which the different components of the HyperNews system execute as software agents. The 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.

1 Introduction

As part of the MEDIA project [1], the HyperNews (Hypermedia Electronic Newspaper) project aims at developing an electronic newspaper environment based on agent technology also called mobile object systems[2][3]. 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. Namely, HyperNews will enforce copyright control [4][5][6] and revenue collection. Moreover, it will offer the information consumers the means to reduce the time spent in retrieving the information for which they have expressed interest without being tied to a specific information provider. HyperNews offers the information consumer an interface to an electronic newspaper system which is both *vendor independent* and customized according to his needs[7].

The MIT Media Laboratory has developed an electronic newspaper system called Fishwrap [8][9] used by the MIT community since 1993. It is probably the most relevant related work in this field. 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.

[1] This work was supported by the Swiss National Science Foundation with the SPP-ICS 1996-1998 project "HyperNews" (project number 5003-045333).

From the information consumer's viewpoint, the main idea is to provide him with the means to create his own electronic newspapers customized according to his information interests as well as a structure and layout he has specified. A Java enabled World-Wide Web[10] browser can be used as an electronic newspaper interface to retrieve and read electronic news. The user defines and can manages as many electronic newspapers as he needs (e.g., work, private, etc.). For each one of them, he specifies his information interests from the classification of the various information providers from which information is requested (e.g., the finance section from the one provider, the sport section from an other and anything about the O.J. Simpson trial from a third one). In addition, he can specify the general structure of his electronic newspapers (e.g., a tree like structure with a main page and a set of sections and subsections) as well as the content of these structure elements. Moreover, the user can choose to be notified upon availability of information updates if such services are offered.

The information provider's HyperNews environment offers the means to publish HyperNews articles packaged in agents. The article content is in html format and holds also the embedded elements such as images, sounds, applets, hyperlinks to the Web as well as hyperlinks to other HyperNews articles (e.g., historical evolution). It is then assigned a classification identifier, price, title, author, abstract, etc. (i.e., anything depending on the policy of the information provider). The whole content is encrypted in the article agent. We have designed a scheme for the commercial distribution of electronic documents that enforces copyright control and payment at the time the information is accessed for reading [11]. The articles are encapsulated in software agents together with the code for releasing and protecting their content. The scheme is based on public key encryption and requires a trusted third party, which may be a bank, a credit institution, etc. Both parties trust the credit institution to authorize the unlocking of the article against payment from the information consumer. Upon successful payment the article key is released and a nominative receipt is given to the information consumer for subsequent access. The scheme satisfies the security and distribution requirements of the HyperNews system. Namely, articles are paid on a usage basis, can be held and distributed freely and can be accessed subsequently through a proof of purchase. Moreover, it offers the information providers the means to protect their copyrights, intellectual property and to collect revenue from their electronic publishing activity.

The most important idea of an environment such as HyperNews is that there is no central authority. Information consumers and providers are inherently distributed and fully independent while their needs and requirements are accounted for.

In this paper we present an architecture for the HyperNews system based on agent technology. Section 2 presents the general layered architecture of the HyperNews system. Section 3 describes the agent system we have chosen to use. Section 4 presents the HyperNews core environment. Section 5 presents the HyperNews application components and tools. Finally Section 6 presents some concluding remarks and future work directions.

2 Architecture of the HyperNews System

In order to gather experience and gain knowledge of the field, we have implemented 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"[12]. It also helped identify the issues and the requirements for an electronic newspaper system based on agents [13]. 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 and are currently implementing a second HyperNews prototype which architecture is described in this paper. It is based on the Java language [14] [15].

The whole agent based HyperNews system is composed of three layers shown graphically in Figure 1. The lowest gray shaded layer is the agent layer representing the underlying agent infrastructure over which the HyperNews system is implemented. The second layer, the *core environment*, is the basic building block of the HyperNews system. It enables the safe and secure distribution and exchange of electronic documents (i.e., HyperNews articles). The top layer, the HyperNews application layer, implements the various tools and applications bound to the HyperNews system. These tools are the interface to the users. This layer has two components depending on the role played by the user, either as an information consumer or as an information provider.

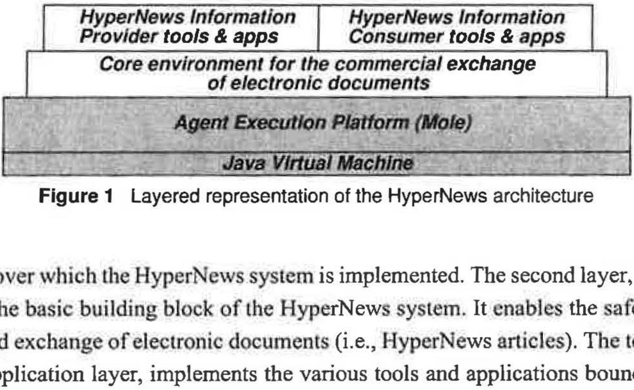

HyperNews Information Provider tools & apps
HyperNews Information Consumer tools & apps
Core environment for the commercial exchange of electronic documents
Agent Execution Platform (Mole)
Java Virtual Machine

Figure 1 Layered representation of the HyperNews architecture

The layered approach offers the advantage of a clear separation between the HyperNews application and the core environment enabling the commercial exchange of electronic documents. Such an approach leads to 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 applications sharing many common characteristics such as copyright protection, revenue collection and security among the major.

3 The Underlying Agent Execution Platform

Mole [16] is a Java based agent system designed and implemented at the Institute of Parallel and Distributed High-Performance Systems (IPVR) of the University of Stuttgart. A central notion of the Mole system is *Location.* Locations 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 of an agent system. 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. Mole offers two types of agents: *user agents* and *system agents.*

User Agents are granted very limited rights to undertake any action in the system and no rights to act outside the agent system unless they acquire prior authorization from system agents. User agents can communicate with other agents, migrate and create new user agents.

System Agents have more privileges than user agents. 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 can not migrate to other platforms.

Mobility in Mole can be achieved in two ways: by remote execution and by migration. As stated in [16]: "Remote execution is a rather simple mechanism. An agent is transferred to the desired location where it is activated and executed. This can be done in two ways: (i) the program code is transferred to the destination location and there, a new agent is created and started directly, (ii) a new agent is created at the source location but not started. It migrates immediately to the destination location and is started there." Migration involves stopping an already running agent and sending it to the remote location to resume its execution. Although the mobility by remote execution has been fully implemented in Mole, the mobility by migration is not yet fully operational. This is due to the problems encountered in stopping threads while they are executing, transporting their state and resuming their execution remotely.

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. Message passing can be both synchronous or asynchronous, and allows transfer of messages between agents.

4 The HyperNews Core Environment

The agent execution platform can be decomposed into three main *areas* for holding agents depending on their privileges and roles: the *HyperNews entrypoint area*, the *HyperNews restricted area* and the *HyperNews 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 *HyperNews 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 *HyperNews system area*. Foreign agents can never reside into the *HyperNews system area*. Finally, all user interaction and local system resources are managed by agents that reside in the *HyperNews system area*. Figure 2 illustrates the complete HyperNews architecture.

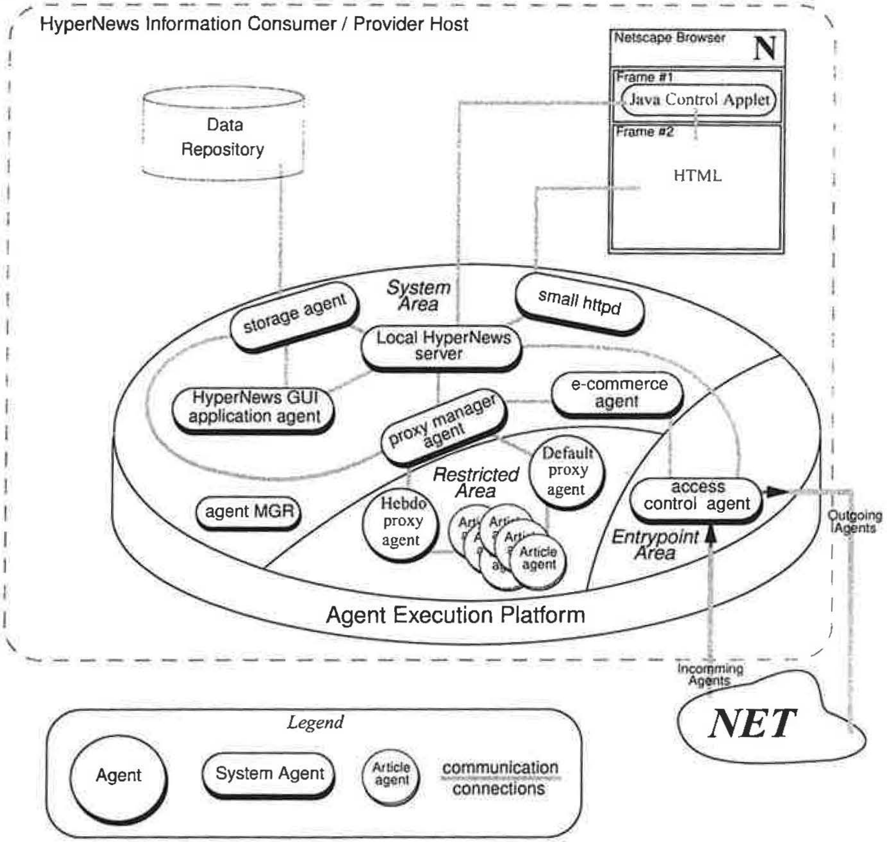


Figure 2 The architecture of the HyperNews system based on agents

4.1 The HyperNews Entrypoint Area

Although security is taken into account at the agent platform level, another level of security needs to be introduced from the HyperNews system point of view to enforce application level security policies. The HyperNews system will provide mechanisms to master and control agent traffic between the platform and the outer world. This is done by introducing the *HyperNews entrypoint area*. The role of this area is to be a placeholder of incoming and outgoing agents before they acquire authorization for either entering or leaving the platform. In addition, this area holds a unique system agent: the *access-control agent.*

The *access-control agent* has high privileges on the platform. It provides at the HyperNews 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 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 placed in the HyperNews restricted area. The HyperNews system agent (the local HyperNews 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 the right to leave the platform. All the agent traffic can be monitored, traced and logged by the access-control agent.

4.2 The HyperNews Restricted Area

The *HyperNews restricted area*, is the placeholder for all the foreign user agents when granted access rights. The user agents have restricted rights and most of the actions they undertake have to be accepted or processed by HyperNews system agents. A common characteristic of the user agents is that they all reside in the HyperNews restricted area since they are considered foreign agents. We have four user agents: the *information provider proxy agents*, the *HyperNews article agents*, the *default proxy agent* and the *proxy-update 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. It also handles the interaction with the local HyperNews articles of the provider it represents. It is installed when a user requests information for the first time from an information provider. However it 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 users platform. It can be updated by the information provider who owns it, through a proxy-update agent. It communicates with the HyperNews 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.

The *HyperNews article agent* is the wrapper of two elements: the article content (i.e., the *raw material)* and the code defining the behavior to access this content. The article is encapsulated in an agent both for security reasons and for distribution convenience. When the article agent is unable to find its local information provider proxy agent, the default proxy agent takes care of interacting with that *orphan* HyperNews article. Article agents, can be freely copied, forwarded and held since content access is bound to successful payment.

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 article agents. For example, when receiving a forwarded article, the issuing information provider of that article can be unknown. In such a case, the default proxy agent will be able to undertake the role of the missing proxy agent to interact with the *orphan* article 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.

4.3 The HyperNews System Area

The *HyperNews system area* holds the remaining HyperNews system agents. Namely, the *agent-manager agent*, the *proxy-manager agent*, the *electronic-commerce agent*, the *storage agent*, the *local HyperNews server agent*, the *HyperNews 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 HyperNews resources on the platform. Moreover, due to their nature (i.e., system agents) they can not migrate. The system agents that lie in this area represent the core elements of the HyperNews system.

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 will be either loaded from or saved to secondary storage, created, started, stopped, destroyed, etc. It will help monitor agents on the platform and 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 HyperNews system agent could request that a particular user agent be terminated or a user agent might want to retrieve information concerning its own execution.

The *proxy-manager agent* 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 HyperNews system, a request for payment to open and decrypt the content of an article agent, the local storage and retrieval of HyperNews article agents, the request for creating a user agent, any migration or communication request among other possible actions.

The *electronic-commerce agent* is responsible for handling payments and other monetary transactions. It provides a unified interface to electronic payment independent of the underlying payment scheme used. It can be contacted either by the *proxy-manager agent* on behalf of the *proxy-agent*, or by the *local-HyperNews-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 the right to migrate.

The *storage agent* is responsible for all the actions requiring access to a storage system. It can be a database, a plain file system or even a caching system for HyperNews. This agent provides a unified interface to data management. It can be contacted by any HyperNews 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.

The *local-HyperNews-server agent* is the core element of the HyperNews system. It almost stands in the middle of everything that happens in the HyperNews environment. This agent is always contacted by the access-control agent upon arrival of any HyperNews 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 and dispatches if necessary, all the HyperNews requests issued through the HyperNews reader (i.e., the Web browser) or any other HyperNews GUI application agent residing on the platform.

The *HyperNews GUI application agent* implements all the GUI aspects of HyperNews except for the HyperNews reader which is a standard Java enabled Web browser such as Netscape for example. Mainly, it communicates with the local HyperNews server agent, the storage agent and with the user directly. Examples of HyperNews GUI application agents are profile managers, user information update notifiers, HyperNews data warehouse managers, electronic wallet user interface among others which are described in the next section.

The *small-httpd agent* implements a local reduced web server in order to process all the HyperNews requests issued by the local HyperNews user. This server is only contacted by the web browser when HyperNews links are selected. Other Web links go directly to the Internet.

5 The HyperNews Application Components and Tools

Given the core environment described above and the distribution scheme, we now consider the specific HyperNews tools and application with respect to both the information consumer and provider. The HyperNews GUI application system agent will be used and derived in order to implement the different HyperNews tools. As such, they will run as system agents in the *HyperNews system area*.

5.1 The Information Consumer

The basic needs of the information consumer is to be informed according to his specific interests using a personalized layout. The information consumer will receive the information via the network. However, it is most probable his computer will not be connected all the time to the network. The system has thus to provide means for dealing with off-line information consumers in case they requested to be notified when information updates are available. In the case where the user is part of an organization, the solution is rather straightforward: a local hypermedia newspaper server buffers client data as they arrive. In the case of a stand-alone information consumer, the buffering will have to take place on a remote site (like for example the information provider or the service provider) until the information consumer restarts his machine.

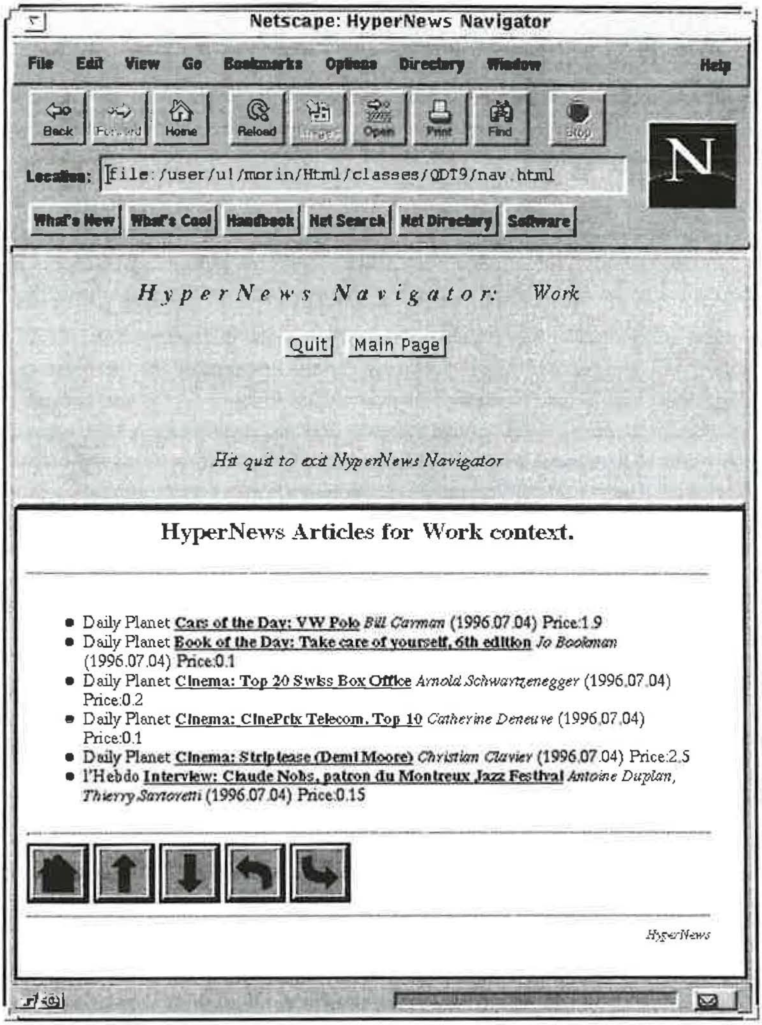
In order for the user to manage his environment a number of tools are needed. These tools should allow the user to manage contexts of information (i.e., electronic newspapers), specify where information is to come from, what are his information interests (i.e., information profile), how the information is to be presented in terms of layout (i.e., presentation profile), to retrieve, manage and read the HyperNews articles through a hypermedia newspaper reading tool. The major tools are outlined below:

The *context manager tool* provides the information consumer a way to manage the different information contexts according to his needs. For example, a user having a private context and a set of professional contexts might need to create new contexts, merge them into a new one or even to create a new one by tailoring an existing template context according to specific needs.

The *information profiler tool* provides three components: the *information source selector and manager*, the *classification browser and user defined information interest selector* and the *information profile builder and manager*. The *information source selector and manager*, allows the user to select, add, remove and modify the users personal information source address directory. In other words, to manage where the information is to come from. It should offer the possibility to retrieve general information and pricing policies from a given information provider as well as general on-line subscription and registration possibilities. The *classification browser and user defined information interest selector*, allows the user to specify explicitly his information interests by browsing the classification and selecting the relevant sections, topics, subjects, keywords etc. The *information profile builder and manager* finally, allows the user to generate, update and merge the information profiles.

The *presentation profiler tool* is also composed of three elements: the *structure editor*, *the abstract-information presentation merger* and the *presentation profile builder and manager*. The *structure editor* allows the user to define *how* the information is to be shown. For example, the structure of the presented information can be linear, tree based, cue card style, menu driven, etc. The *abstract information presentation merger* is more concerned about "where to put what". It actually links together the information profile and the structure to build the wireframe of the hypermedia electronic newspaper ready for *instantiation* (i.e., retrieval). The *presentation profile builder and manager* allows the user to generate, update and merge the presentation profiles.

The *HyperNews reader* is the interface to the hypermedia newspaper system. This tool should be as much as possible based on existing technology. Due to wide spread availability and platform independence, the use of a Java enabled Web browser, such as Netscape, appears to be the best solution. It offers many advantages like a standard rendering interface and formatting language (html [17]), the ability to use in a transparent way through the same interface both the Web and the HyperNews system through the simple, yet powerful, *http* [18] communication protocol. Moreover, the support for multimedia elements and Java based applets as article embedded components is obtained for free. Figure 3 shows an example of the HyperNews reader presenting a container page of a user's personal newspaper entitled *work*. The top frame contains a simple navigation applet. The bottom frame contains the set of articles relevant to the current page of the work context. Each entry displays the general information about their content (i.e., name of provider, title, author, date, price, etc.). The article hyperlinks when selected trigger the payment and reveal the article content as shown in Figure 4. The bottom navigation links allow



Location: file:/user/ul/morin/Html/classes/QDT9/nav.html

HyperNews Navigator: Work

Hit quit to exit NyperNews Navigator

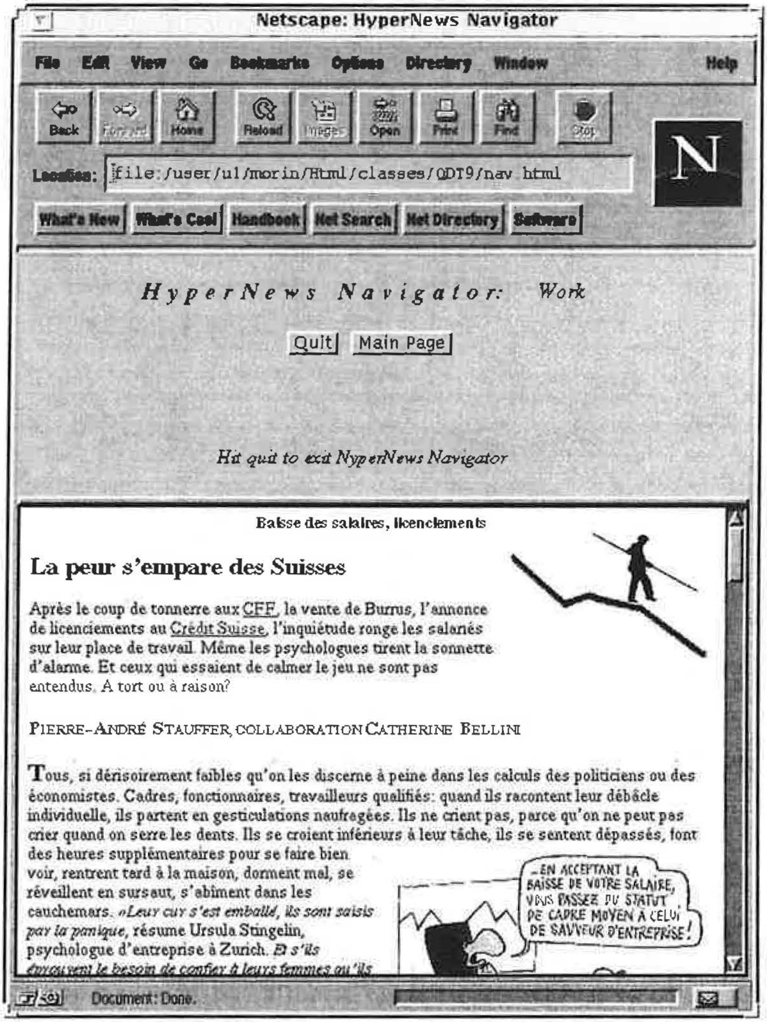
HyperNews Articles for Work context.

- Daily Planet **Cars of the Day: VW Polo** *Bill Carman* (1996.07.04) Price:1.9
- Daily Planet **Book of the Day: Take care of yourself, 6th edition** *Jo Bookman* (1996.07.04) Price:0.1
- Daily Planet **Cinema: Top 20 Swiss Box Office** *Arnold Schwartzenegger* (1996.07.04) Price:0.2
- Daily Planet **Cinema: CinePrix Telecom. Top 10** *Catherine Deneuve* (1996.07.04) Price:0.1
- Daily Planet **Cinema: Striptease (Demi Moore)** *Christian Clavier* (1996.07.04) Price:2.5
- l'Hebdo **Interview: Claude Nobs, patron du Montreux Jazz Festival** *Antoine Duplan, Thierry Sartoretti* (1996.07.04) Price:0.15

Figure 3 HyperNews article container page

the user to navigate through the structure of the personal newspaper. The hyperlinks in the article content are regular Web URLs offering the user the possibility to access directly any referenced material. Although these figures are taken from our first prototype, they serve the purpose of showing the ideas of the HyperNews reader.

The *electronic wallet* implements the user interface for commercial transactions and management with the financial and credit institutions.


Netscape: HyperNews Navigator
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
N
What's New What's Cool Handbook Net Search Net Directory Software
Quit Main Page
Document: Done.

Location: file:/user/ul/morin/Html/classes/QDT9/nav.html

HyperNews Navigator: Work

Hit quit to exit NyperNews Navigator

Baisse des salaires, licenciements

La peur s'empare des Suisses

Après le coup de tonnerre aux CFF, la vente de Burrus, l'annonce de licenciements au Crédit Suisse, l'inquiétude ronge les salariés sur leur place de travail. Même les psychologues tirent la sonnette d'alarme. Et ceux qui essaient de calmer le jeu ne sont pas entendus. A tort ou à raison?

PIERRE-ANDRÉ STAUFFER, COLLABORATION CATHERINE BELLINI

Tous, si dérisoirement faibles qu'on les discerne à peine dans les calculs des politiciens ou des économistes. Cadres, fonctionnaires, travailleurs qualifiés: quand ils racontent leur débâcle individuelle, ils partent en gesticulations naufragées. Ils ne crient pas, parce qu'on ne peut pas crier quand on serre les dents. Ils se croient inférieurs à leur tâche, ils se sentent dépassés, font des heures supplémentaires pour se faire bien voir, rentrent tard à la maison, dorment mal, se réveillent en sursaut, s'abîment dans les cauchemars. *«Leur cœur s'est emballé, ils sont saisis par la panique*, résume Ursula Stingelin, psychologue d'entreprise à Zurich. *Et s'ils éprouvent le besoin de confier à leurs femmes qu'ils*

–EN ACCEPTANT LA BAISSE DE VOTRE SALAIRE, VOUS PASSEZ DU STATUT DE CADRE MOYEN À CELUI DE SAUVEUR D'ENTREPRISE!

Figure 4 HyperNews article example

5.2 The Information Provider

The role of the information provider is to provide information consumers the information for which they have expressed interest. A major difference between consumers and providers is that the provider will have to be available at all times for client requests. This means that it will be his responsibility for buffering information updates in case the information consumer is not able to do it himself.

The information provider environment requires a set of tools in order to publish HyperNews articles over the network on a commercial basis. Three major tools are anticipated for this environment. First, for creating and managing the publisher's information classification, second, for composing, packaging and publishing HyperNews articles and third for allowing the inclusion of hyperlinks in the HyperNews articles.

The *information classification builder tool and provider description* supports two modes: creation mode and update mode. In creation mode upon start-up it gathers general information about the information provider, the general pricing policy, revenue collection methods used, etc. It is also used to create the information classification which will be available to information consumers to define the information profile bound to this specific information provider. In update mode, it is used whenever changes are made to the information classification thus leading to appropriate notification to the information consumers for synchronizing classifications.

The *information composition and publishing tool* has two main components: the *composition browser* and the *information unit maker or packager*. The *composition browser* is a browsing-viewer style interface to the storage system (i.e. database, file system, etc...) allowing the publisher to compose an information unit by assembling independent elements (e.g. a picture in gif format, an article in html and a Java applet). It is important to mention that the actual article composer can be any html editor. Thus, this tool is not mandatory in the system. However, it can be used as a helper application generating the corresponding html file prior to packaging it. It can also be used as a specialized article browser although this task can be handled by installing the HyperNews consumer environment on the information provider's site. We can also anticipate the use of other helper applications such as external translators for example. The *information unit maker or packager* allows the publisher to classify the information to publish by *tagging* it with the corresponding classification signature and other information such as pricing information, authorization scheme, expiry date and any policy dependent information relevant to the published information. Moreover, this tool provides data encryption and key management according to the proposed distribution scheme. It represents the article agent wrapper and packager of the article to publish.

The *information hyperlinker tool* this tool offers the possibility of hyperlinking published information. We anticipate two types of links for the moment from the point of view of the information provider. The first type, the *historic hyperlink*, concerns the evolution of the information (e.g. previous or next in the sequence of articles, first or initial article, etc.). This type of link is of prime importance for access to the historical evolution of an information. The second type, *the semantic hyperlink*, will be used to establish either intra or extra information source links to related material, for details, encyclopedic information, museums, digital libraries, archives etc.

6 Conclusion

In this paper we have presented the architecture of the HyperNews electronic newspaper system based on agents. We have adopted a layered approach in order to provide a clear separation between the different layers of the system. Namely, the agent infrastructure, the core environment enabling the safe and secure distribution and exchange of electronic documents and finally the specific HyperNews application components and tools. We have then discussed separately each layer and the interfaces between them. Moreover, the proposed architecture is a contribution in the direction of publisher independent, commercially viable electronic publishing while leaving enough freedom for customized publisher dependent policies.

We plan to evaluate our system in a commercial environment. According to our schedule the installation of the HyperNews system at our industrial partner's site "L'Hebdo" is planned for the end of 1997. Based on the results and the evaluation we will receive, we will draw a first set of conclusions and we will provide an updated version of the system.

Upon completion of the HyperNews project, we plan to address the issue of hypermedia electronic publishing in a broader perspective. In addition to electronic newspapers and magazines, we consider digital libraries, electronic books and many other types of value added electronic publishing services as different classes of electronic publishing applications. These application classes share similar requirements from the point of view of an underlying electronic publishing infrastructure. Thus, we will define a framework for commercial hypermedia electronic publishing.

References

[1] 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.

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

[3] K. Rothermel and R. Popescu-Zeletin (Eds.), "*Mobile Agents*", Springer-Verlag, LNCS 1219, 1997.

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

[5] The Cyberspace Law Institute, http://www.cli.org/

[6] Pamela Samuelson, "*The Copyright Grab*", Wired Magazine, Issue 4.01, January 1996

[7] Jean-Henry Morin 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.

[8] 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.

[9] Fishwrap Documentation center, http://fishwrap-docs.www.media.mit.edu/docs/

[10] 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

[11] Vassilis Prevelakis, Dimitri Konstantas and Jean-Henry Morin, "*Issues for the Commercial Distribution of Electronic Documents*", *Proceedings of CMS'97, Communications and Multimedia Security Joint Working Conference IFIP TC-6 and IFIP TC-11*, Athens, Greece, September 22-23 1997. (A shorter version is published in the *Proceedings of ERSADS'97, Second European Research Seminar on Advances in Distributed Systems*, Zinal, Switzerland, March 17-21 1997.

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

[13] Jean-Henry Morin, "*Requirements for a Hypermedia Electronic-Newspaper Environment based on Agents*", Working paper, Technical Report, Centre Universitaire d'Informatique, University of Geneva 1997.

[14] James Gosling, Bill Joy and Guy Steel, "*The Java Language Specification*", The Java Series, Addison-Wesley, 1997.

[15] Ken Arnold and James Gosling, "*The Java Programming Language*", The Java Series, Addison-Wesley. 1996.

[16] 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.

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

[18] T. Berners-Lee, R. Fielding and H. Frystyk, "*Hypertext Transfer Protocol-HTTP/1 0*", RFC 1945, May 1996.