e-commerce for interactive video database

R. Asenjo, I. Navarro, M. Rodríguez, M. Ujaldón, E.L. Zapata

Abstract— **We have recently developed an e-commerce module to complete an interactive video database tool (Tarsys) [4], [2] designed in our group. Tarsys allows several clients to display and edit any part of one or several video archives in digital form, and now it is also possible to commercialize the archive content via Internet. Existing solutions for digital video e-commerce are simple video servers like MediaBase [11] which are not designed on top of a database management system (DBMS) and therefore lack of powerful query capabilities and other well-known drawbacks of the file-oriented information systems. Instead, our application is builded in top of a relational DBMS, allowing customers to search the archive by several criteria such as a GUI interface or SQL queries. The requested videos fulfilling the query conditions can be previewed with RealPlayer [8] and later purchased using a secure web page. The tool retrieves all the video information from Tarsys, which may archive news items in a cable television as well as bacterium videos for scientific purposes. The development has been carried out in Java, using Servlets [7] and JDBC [5].**

Keywords— **e-commerce, video database, query capabilities, Java.**

I. Introduction

WE are being witnesses of the conversion of video archives to a digital form, just another step in the digitalization race (voice, music, images and now video archives). Sooner or later, the analog formats will be relegated to old devices to finally disappear. There are many reasons supporting this fact, but we would stress here the reliability and speed than digital video provides. This is true not only due to the random access, but mainly to the possibility of extending the video stream with additional information (data-stream) either at conversion or production time. That way, we can easily locate certain frames through a fast context search in the specific data-stream.

Those ideas led us to build a computer aided system to convert, catalog and retrieve digital video that we named as Tarsys (Television ARchive SYStem)[4], [2]. Tarsys relies on the client-server paradigm where the server comprises the digital and analog capturers and a video database accessing huge disks as well as a robotized tape library. On the other side, the clients can take several forms depending on their function. For example, one of the designed clients is a GUI tool which provides friendly access to the server in order to insert, edit, catalog and search video. Currently, Tarsys supports and stores two different video formats, each one devoted to a particular function [3]: MPEG-2 4:2:2 provides high quality and MPEG-1, which has lower quality but also consumes much less bandwidth, is used for fast previewing in the intranet. These two formats support streaming of either the whole or a section of the videos. The main advantages of streaming are that the client do not need to store the video, saving disk space, and on the other hand, the user avoids the download idle time, since the video is displayed as it is received.

Computer Architecture Department. University of Malaga. E-mail: {asenjo,inma,luz,ujaldon,ezapata}@ac.uma.es

Summarizing, Tarsys is a flexible, modular and scalable video archive management system which provides fast and context-sensitive access to digital video. It can be used either by the departments of redaction, production or post-production in an audio-visual company, or by a research group interested in building and search an archive of bacterium videos recorded from a microscope. However, Tarsys can not be accessed through Internet, partially because of the security layer which restricts the access to unauthorized users. Nonetheless, Internet access can be of great importance to allow video e-commerce or video-on-demand capabilities. For example, regarding this last issue, Tarsys can be the repository of videos for a video server or video profiler which feeds a cable video on demand network. In such a case, it may be desirable for a customer to search through the Internet the video server for available films of a certain actor made in a particular year. Another simpler motivating example is a TV producer company which offers some of their videos, like football matches, via Internet. An overseas customer would be able to search particular matches fulfilling some criteria, preview a fraction of the videos, and later decide whether or not to buy them via Internet.

Currently, MediaBase [11] is an approximation to the requirements we need, offering an Internet interface and providing streaming capabilities. However, we discarded such solution for three basic reasons: First, its limited query resources, searching videos only by keywords, second, the lack of e-commerce functionality, and third, its design targeted to MPEG-2 and MPEG-1, which are too bandwidth consuming for Internet purposes. Therefore, MediaBase stands as a good streaming source for intranet, but we need a better solution to cover Internet requirements.

Instead, we pose a solution based on the "RealSystem" [8] streaming format to provide video via Internet, an SSL [10] module included in a web server to provide secure e-commerce, and a DBMS to allow complex searches in the video archive and store other bookkeeping information (customers, statistics, and so forth). This three components can be accessed from the Internet through a complete web interface developed in Java.

Actually, the video and related data that will be available via Internet are some of those included in the Tarsys system (only a selected set of videos will be publicized). However, we have uncoupled the two applications building an e-commerce module which can be connected to Tarsys and has a different database storing only the public videos with their related information. This may seem to violate the database principles, basically replicating information with the associated coherence problem. The reasons be-

hind supporting such an uncoupled design are:

- Our e-commerce module can be sold independently of Tarsys, only to Tarsys customers interested in their video e-commerce.
- Tarsys security and integrity problems would increase significantly if it would be accessible through Internet. The e-commerce module does not need the Tarsys edition capabilities, and therefore, it is unnecessary to expose the Tarsys system to the net. Moreover, the e-commerce module can be installed in a different computer, which can be the web server and also assumes the role of a firewall to avoid any other access to the intranet.
- The Tarsys database server could be overwhelmed if it would serve intranet as well as Internet clients.
- The Tarsys database does not need to mess up with customers and selling information.
- It is always in our minds to avoid any interference with the development of Tarsys.

In any case, the e-commerce module can keep the information coherence thanks to its ability for periodically connecting the Tarsys server to retrieve new public videos and delete or modify video information which has been changed in the server.

The rest of the paper is summarized next. Section II describes the whole system giving its functionality and a first approximation to the developed architecture. The database design and the coherence protocol is presented in section III. A brief description of the Java interface and access to the database as well as the security constrains are addressed in section IV. Finally, conclusions and future work close the paper.

II. System overview and architecture

The goal in this section is to describe the functionality and organization of the e-commerce module we have developed. For the sake of completeness, a brief introduction to Tarsys will also be given.

A. Functionality

The e-commerce module is conceived as an additional client of Tarsys which provides an Internet interface. Tarsys server functions are multimedia data insertion, semi automatic content-data-stream extraction from digital video, multimedia data retrieval and query management. All these capabilities are available to intranet clients, but not from the Internet. That way, the e-commerce module removes this limitation increasing the functionality of Tarsys. More precisely, this module provides:

- Web environment to show the multimedia offer. Customers can make queries and powerful searches through the video DB using a friendly interface. The customer can also browse the whole catalog. After selecting a video subject (films, music, sports, ...) the customer can look for certain videos fulfilling some criteria. These criteria are specific values of the video subject attributes, like for example, date less than 2/4/99 and/or genre equal to drama.
- All the information regarding this video whenever extracted from the database. A fraction of the video can be previewed with a RealPlayer via a streaming broadcast in which the download time to start the video playing may be skipped, for the player displays the video "during the download" instead of "after the download". The Real format also provides an adaptive quality depending on the available Internet bandwidth.
- E-commerce tools using the SSL (Secure Socket Layer) encryption to provide a safe environment in which personal information transactions and shopping can be carried out. Customers can open an account and append items to their shopping cart. Customers information, statistics and shopping information are also stored in the e-commerce database.
- A Tarsys client, which keeps the coherence between the Tarsys DB and the e-commerce one. Since not all the Tarsys videos are going to be sold, this client searches into the Tarsys database only for commertializable items still not included in the e-commerce database. Modified information in the Tarsys database is also updated in the e-commerce one. Besides, this client handles the retrieval of a fraction of the MPEG-1 copy as well as the later conversion to the Real format.

B. System architecture

Figure 1 schematically displays the whole system organization. The Tarsys server lies on the left side of this figure, containing the following elements:

- The digital/analog capture subsystem, which converts the input signal to MPEG-2 and MPEG-1 format and stores the videos in the video DB. At the same time, some video technical information (bit rate, aspect ratio, resolution, etc.) is stored on the database.
- The video database has very large capacity requirements, for which a tertiary storage unit (like a robotized tape library) is a must. Hard disks behave like caches for the tapes.
- The information stored in the data DB comprises the technical issues, video description (title, abstract, subject, participant and other attributes) and data-stream information which allows us to retrieve a given scene through a description, like keywords, data about the contents of the scene, events, places or colors.
- This data stream information is partially extracted by the "Automatic content extraction" subsystem, builded on top of specific modules devoted to a particular goal (scene detection, world transcription, text recognition, bacterium movement and so on).
- Finally, the "Query management" and "Video access management" subsystems provide the interface to serve several kind of Tarsys clients. One of them is a GUI tool for allowing friendly access to Tarsys videos, data completion and query.

We now focus on the right side of the figure, where we can see an additional Tarsys client providing video e-commerce capabilities. Its subsystems are the following:

- The e-commerce database, which stores a subset of the video description and other information barely relevant to Tarsys for being tightly coupled to e-commerce issues (i.e.,

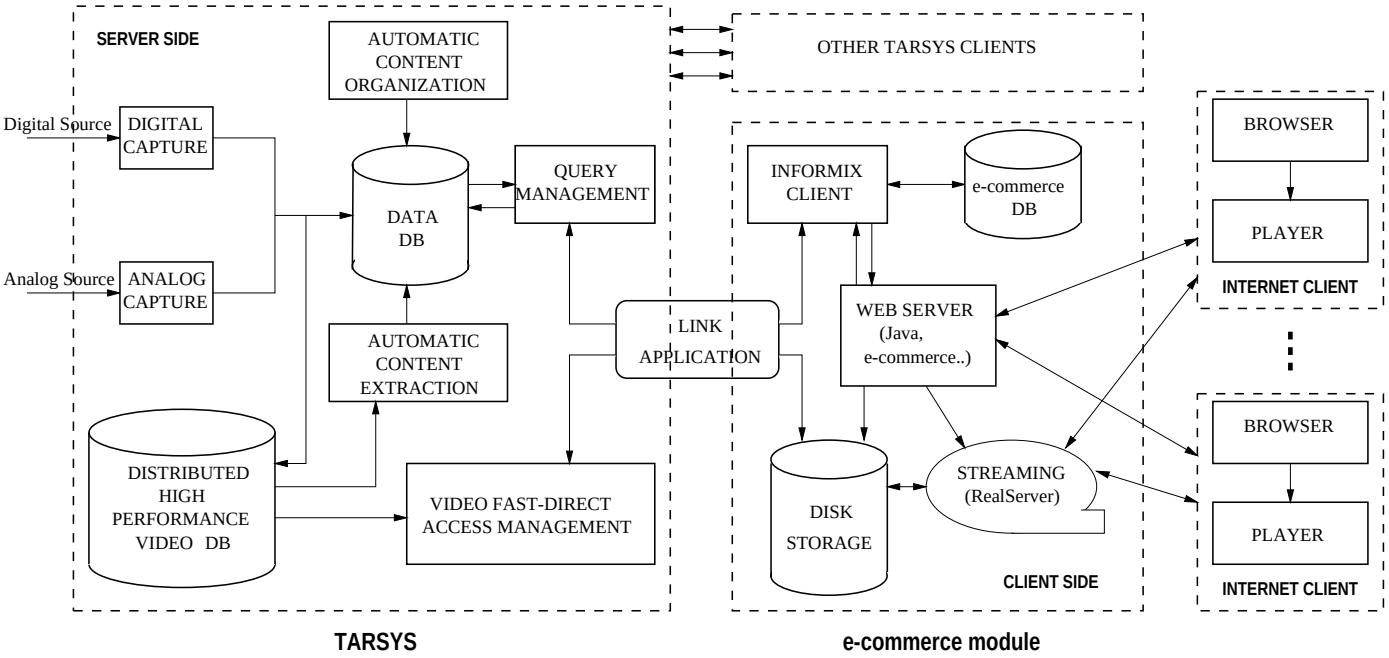

Fig. 1. Tarsys and e-commerce module global architecture.

video price or customer information). This database dynamically adapts to the organization of the Tarsys database thanks to additional metadata tables. In our implementation, we have recourse to the Informix DBMS.

- Videos are stored on disk using the Real format, which is specially designed to fit the Internet constrains. In particular, the SureStream technology is much less sensible to package loss than MPEG; audio and video quality dynamically adjust to the Internet bandwidth; and finally Real format achieves high speed for the coding and decoding operations involved.
- The web server provides the Internet interface. This subsystem includes the actual server (Apache in our implementation) and the Java programs and Servlets we have developed to provide a friendly interface. The server supports the SSL protocol to encrypt critical information.
- The web interface provides query capabilities by accessing the e-commerce database through an Informix client. This client was developed in Java and connected to the database via JDBC.
- Streaming capabilities are provided by the RealServer, directly accessed by the Internet clients, which reads the requested video previews from disk.
- Finally, the database coherence application keeps track of new or modified multimedia files in Tarsys. This subsystem also accesses to the Informix client to update or modify the e-commerce database. It also handles the retrieval part of Tarsys MPEG-1 videos, and convert them to Real format.

The next two sections go deeply in the description of these subsystems, organizing them into two blocks: subsystems related with the database and subsystems related to the Internet interface.

III. The database and information coherence

The information managed by the database should allow operations such as customer registration, display the shopping offer either by browsing or searching, video preview, selection of one or more videos to buy, payment option, and many more. On the other hand, the video information should come from the Tarsys system, which pose conflicts as the e-commerce database should reflects the changes performed on the Tarsys database. Note that this adaptive behaviour is related to the data within the tables and the ones created in Tarsys itself.

Let us present the problem more precisely. In Tarsys, one video belongs to a certain category and, inside this, to a subcategory (noted from now on as Category-Subcategory). Certain video attributes are independent of the video classification (title, length, keywords), but other exists only for certain categories and subcategories: for example, attributes such as singer, longplaying record and record company in a Music-Pop video; or match date, referee name and tournament on a sport event.

Considering all this, our database should be designed not only to store a certain Category-Subcategory video attributes, but also to automatically create the necessary tables when new categories are considered in Tarsys. This flexibility can be achieved thanks to the use of a Metadata database storing the structure of those database entities that are not present during the whole life of the database (they can appear or disappear). Figure 2 shows the entity-relation (E-R) diagram for the e-commerce database.

From right to left we can see first the “Customers” entity, which may include attributes such as name, address, phone or e-mail. This entity is related (through the “Order” relationship) to the “Shipments” entity, whose attributes are address, payment option, amount, and any other informa-

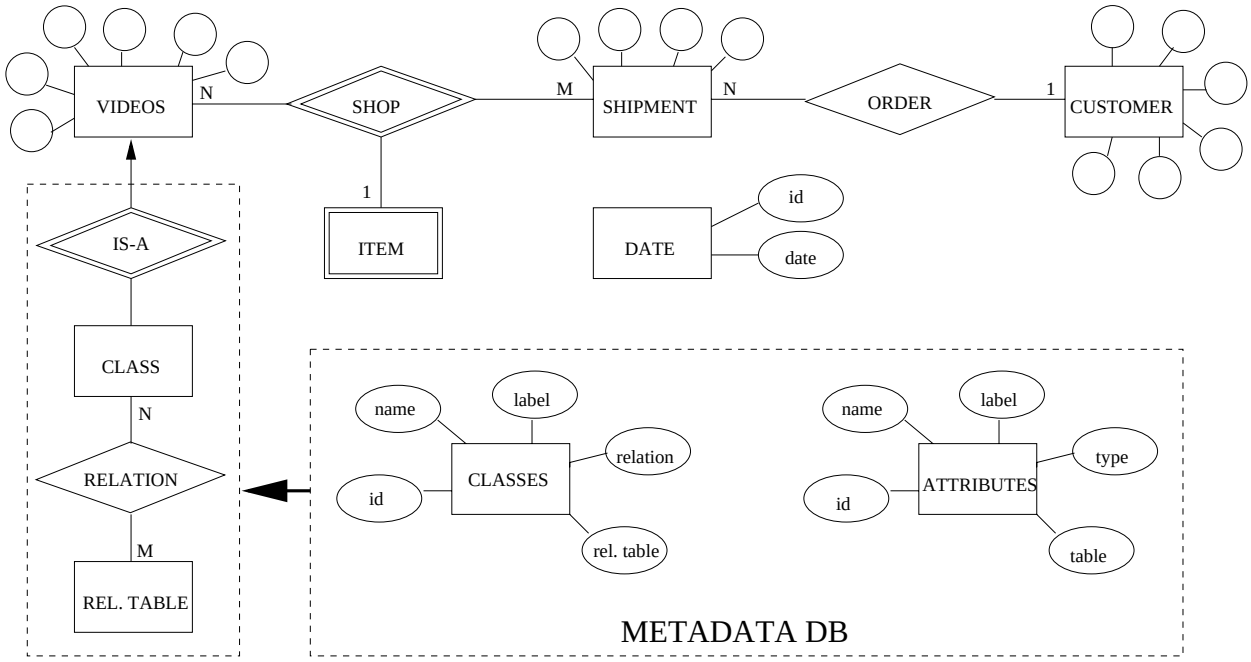

Fig. 2. E-commerce DB entity-relation diagram.

tion related to the ordering procedure followed by the customers. At the same time, the "Shipments" are related to the "Videos" entity through the relationship "Shop" (i.e., several videos can be included in a shipment). The "Item" entity can also be seen like an attribute of the "Shop" relationship, thus specifying the number of a certain video which are included in a shipment.

The videos entity has attributes like title, length, price, keywords, places, path to the video file, etc, as well as the attributes depending on the Category-Subcategory classification for the video. If we consider an example with three different Category-Subcategory video classes: Music-Pop, Sport-Football and Films, all of them will be "Class" entities related to the Video entity through the "Is-a" relationship. Depending on the class, certain attributes may be present: song name, longplay, record company,...; match date, stadium, referee,...; director, producer, genre, etc. Additionally, some of these entities can be related to an additional entity (i.e., "Actors" which play films or "Musicians" which play songs), and these new entities have additional attributes. Figure 3 presents the resulting E-R diagram for this example.

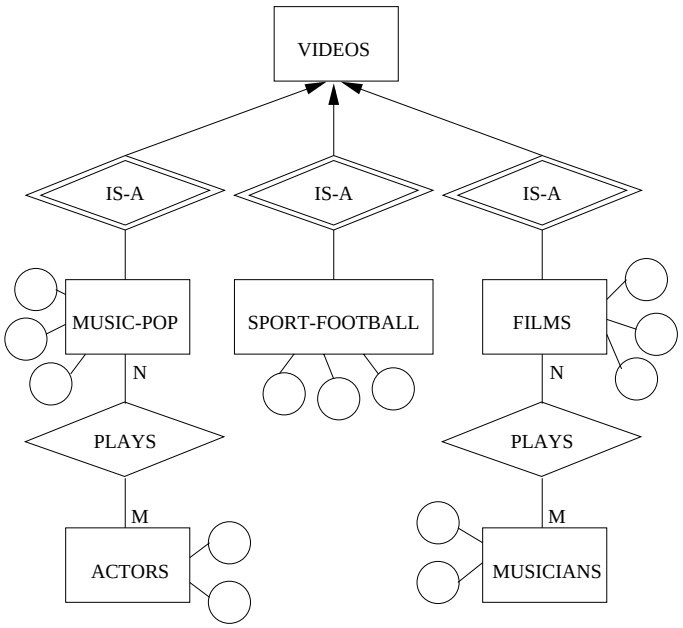

Fig. 3. A possible instance of the video E-R diagram.

All these optional entities and relationships are dynamically managed thanks to the Metadata database. This database is implemented by two Metadata entities: `Classes` and `Attributes`. `Classes` stores information about the name of the table (for example, class1), the label of the entity (for example, Films), the id of the related table (for example, the id of Actors) and the name of the relationship table (for example, the id of Plays). `Attributes` provides the name of an attribute (for example, atr1), its label (for example, director), its type (for example, char) and the id of the table containing such attribute (for example, the id of the table Films). That way, as new Category is included in Tarsys, for example, News, it is possible to update the "Classes" and "Attributes" entities to reflect the particularity of the new kind of videos. This also implies to create new tables ("News", "Presenters" and the "Presented by" relationship) with the new corresponding attributes (date, hour,...; name, specialty,...).

Now, we can describe the application in charge of upgrading the Metadata and e-commerce databases when data change in Tarsys. This link application was written in ESQL/C, and is willing to establish a connection to the Tarsys database as well as our module database. Depend-

ing on the available information in Tarsys, this link application can run with or without operator assistance. The assistance is necessary only when Tarsys does not provide the preview MPEG-1 video and the video price, whereas by doing so, the link application can run automatically (for example on a daily basis, or even triggered by Tarsys after any change).

We now summarize the set of operations that should be carried out by the link application:

- The establishment of a connection to both Tarsys and the e-commerce databases. These connections can be local or remote depending on the machine used for running the link application, which besides has to login with access permission to Tarsys and update permission to the e-commerce database.
- The video information retrieval from Tarsys. We must first search in the "Date" entity of the e-commerce DB the date in which the last synchronization was made, and, with this information, make use of some of the library routines provided by the Tarsys development team to _open, _fetch, and _close a search operation. After that, the link application updates the e-commerce "Date" entity with the corresponding system date.
- To insert the new video data and to merge the subsequent changes in Tarsys. Eventually, this operation may imply to create new tables and attributes, changing the metadata database as stated above.
- The video search and retrieval. Tarsys provides some library calls to edit MPEG video to build the preview video which will be later downloaded and converted to Real format. Finally, the link application just proceeds to close all connections.

IV. Interface to Internet

The main page when entering the system through http is a typical HTML page with two options: registering into the system and accessing the main video catalog (for which you must already be registered).

A. Registering into the system

The user registers by filling a simple application form, an applet written in Java built using classes from the AWT (Abstract Window Toolkit). We prefer this rather than Swing for being simpler and faster. The applet running on the client side communicates with a server written in Java using sockets to poll a port for which the client requests are performed.

As soon as the server detects the connection of a new client, a new thread is generated to deal with its requests and interact with the database. For example, the server contacts the database to verify the chosen login and insert in the `Customer` table all the information obtained. Then, it contacts the customer via e-mail for providing him the actual login and password, for which we use the SMTP protocol [1]

Fig. 4. The Web page showing the main catalog.

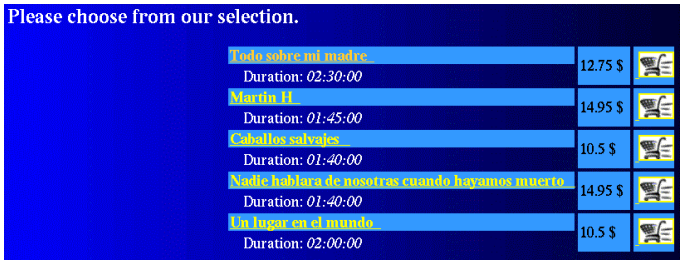

Fig. 5. The Web page showing the films video catalog.

B. Accessing the main video catalog

When the customer selects to enter the system, a typical window asking login and password is shown. After validation, the Web page in Figure 4 is loaded. This page shows a list of video categories for which the customer may either see its catalog or perform a query. In any case, an application form must be filled and, upon completion by clicking on the submit button, a Servlet serves the new page (as well as the remaining pages we'll see from now on). This is faster compared to download pages built with Applets, and, moreover, protects the system as the client side does not execute any program which may be decoded and misused: Through the net are solely transferred requests and answers, carrying all the processing and database accesses upon the shoulder of the server.

If the user selects to see the catalog, the new page loaded is shown in Figure 5. There we can see all the videos currently on the database for the selected category together with information like title, duration and price. At this point, the server has started a new session for the client to move inside the application, creating its shopping cart and allowing him to insert and/or delete items. From this page, choices are: (a) Return to the main catalog (upper left icon), (b) add the video to the shopping cart, (c) ask for additional information for the chosen video and (d) watch a small fraction of the video by clicking on its name.

C. Queries

If the customer wants to perform a query instead of seeing the catalog, the page downloaded is shown in Figure 6. The query fields labeled `title`, `keyword` and `places` are common to all the categories, and then each will have its own list, though it is the same Servlet who serves all the pages.

The queries do not have to fill all the fields, but only those interested in matching. Also, inside a single field, the system extracts all the results containing the string entered

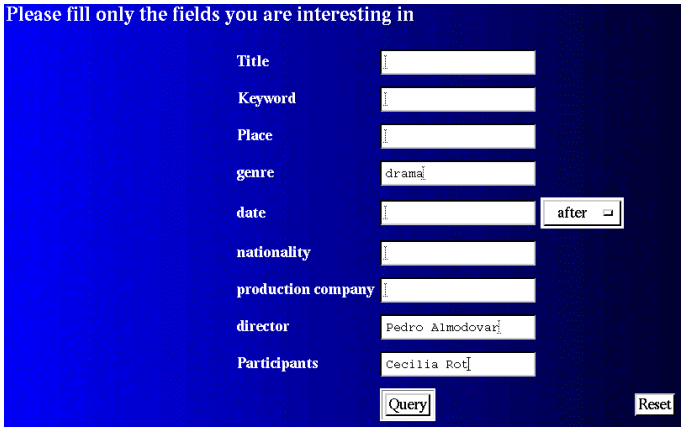

Fig. 6. The Web page to perform a query.

for querying. With this, the list of outputs consists of all the database registers fulfilling at least one of the fields filled.

D. Watching a video preview

Whenever the customer click on a video name, a new page is downloaded showing additional information and allowing him to watch a preview of the video (Figure 7). This fragment is embedded on the page, though it is necessary to install RealPlayer [8] and set up the corresponding plug-in on the navigator. Should the user lack of any of these, it might watch the video by clicking on the `See it` link, which executes RealPlayer on a separated window.

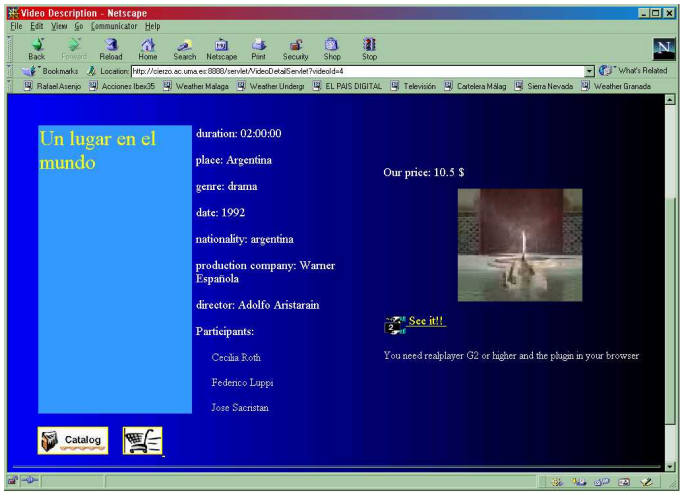

Fig. 7. The Web page with information regarding the video selected by the customer.

E. Shopping cart

Once the shopping cart contains at least one item, two more icons are available, one for seeing its contents and another for ordering the selected products.

The first action brings him to the page showed in Figure 8, which offers a list of the videos selected, an accounting for the set, and the total amount to pay. Once here, the customer may return to the catalog, buy the videos, delete an item or even empty the shopping cart.

Fig. 8. The Web page showing the shopping cart.

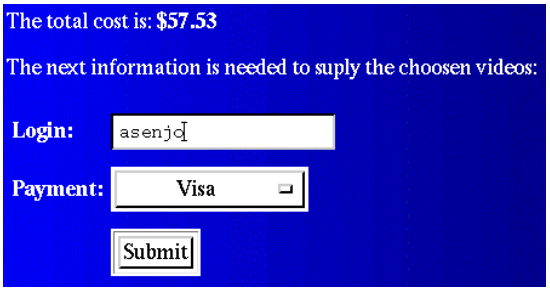

Fig. 9. The Web page to choose the payment method.

If the customer decides to buy the videos, it will be conducted to the payment method page showed in Figure 9. Once there, the payment may be satisfied through credit card, back transfer or at the postal office upon reception; once the selection is completed and the form is submitted, the system will inform about the personal data (name, address, ...) that are on the database regarding the customer. At this point, we have already entered a secure place on the web, as the next operation is to enter the credit card number if required.

F. Checkout

The customer can modify any of its personal data or change the person and/or address the purchased items are going to be sent. The credit card number is sent to the bank, which will be responsible for authorizing the transaction and inform the company about the results.

At this point, the whole query-and-buy cycle has been completed and the only operation remaining is a checkout which confirms the used the purchase he made, and returns him to the main page in the system. A Servlet is responsible for closing all the customer connections, invalidate the session and empty the shopping cart, finally showing a reminder about leaving a secure page.

V. Conclusions

We have developed an e-commerce module to register, catalog and commercialize videos through Internet. We provide a broad range of services to companies connected to our Web pages, including the possibility of visualizing

and editing either the whole or a fragment of such videos stored in both MPEG-1 and MPEG-2 digital formats as well as purchasing them through a secure interface.

Our system consists of:

1. A Web environment to show the videos and perform specific queries. The Web server uses Java for making the web pages more interactive and appealing to the user.
2. Tools for e-commerce making use of the SSL cryptography module to provide secure transactions for personal and financial data.
3. Database manager to deal with the products description, customers data and all the purchasing process.
4. Applications to update the database and maintain its data coherence with the information stored on the Tarsys tool.
5. A streaming system allowing the customer to see video fragments at the same time it's being transferred from the server, using for this purpose the visualization tool on the client's side (player).

In the future work, we plan to extend the system with a graphic application to perform a manual updating of the database, carry out the video edition through a more selective process, and include set of data related to the selling process for marketing purposes.

Acknowledgments

The authors would like to acknowledge the valuable suggestions of Guillermo P. Trabado, Rafael Larrosa and the Tarsys development team.

References

[1] Comer, D.E. *Internetworking with TCP/IP*. Vol. 1. Prentice-Hall, 1991.
[2] R. Larrosa, G.P. Trabado, E.L. Zapata. *Tarsys: An Integrated System for Video Archive Management, Browsing and Retrieval.* IEEE International Conference on Multimedia and EXPO. New York City, USA, July 30 – August 2, 2000.
[3] Sikora, T. *MPEG Digital video-coding standars.* IEEE Signal Processing. Vol. 14, No. 5, September, 1997.
[4] `http://www.andaluciadigital.es/TARSYS`
[5] `http://java.sun.com/products/jdbc/index.html`
[6] `http://java.sun.com/products/jfc/tsc/index.html`
[7] `http://java.sun.com/products/servlet/index.html`
[8] `http://service.real.com/help/library/index.html`
[9] `http://www.informix.com`
[10] `http://www.openSSL.org`
[11] `http://www.sgi.com/software/mediabase`