Information Management Tool

Christian Götze, Kay-Uwe Graw

Fraunhofer Institute of Computer Graphics
Julius-Junius-Str.11, 18059 Rostock, Germany
tel. +49 381 4024 110, fax. +49 381 4024 199
chris@egd.igd.fhg.de, kugraw@egd.igd.fhg.de

Abstract

This article describes an Information Management Tool (IMT) which allows a better use of the vast amount of information available on the Internet. The IMT supports simultaneous use of information from the Internet and the local office area by integrating multimedia documents from different sources into a consistent local information base. An efficient retrieval and the classification of documents with relevant information is possible by means of additional meta information. Furthermore, the user can define relations and associations between any documents within the local information base.

Keywords: Information Management

1 Introduction

A huge amount of information can be found at the numerous WWW-pages which are available through the Internet, the global information space. Current Internet browsers allow an easy access to this vast information pool and are able to display all kinds of multimedia information. However, the current use of these information consists mainly of bookmarking WWW-pages, downloading printable documents or archiving print outs of WWW-pages. Unfortunately, found information can get lost because the information from the browsing process are not stored permanently and information can be removed by the WWW providers anytime without notice. This results in many dead links on the WWW. Print outs from the Internet have the disadvantage that they may become out of date and that they can not be used for information retrieval. Furthermore, the reuse of bookmarked information requires a renewed download of these information from the Internet. This is a waste of bandwidth and results in additional costs especially for Internet user who don't have a permanent Internet-connection, e.g. people using the Internet via ISDN.

Therefore, important information from Internet searches should be stored locally in order to classify and evaluate them for later reuse. This means storing the whole WWW-page locally including as many embedded objects as possible. However, the advantage of local
availability has the disadvantage that information may become out of date. Thus, we need some kind of database where we can store meta information about the documents, e.g. the original WWW-address (URL) and the download date of an imported WWW-page, which can be used for an automatic or user requested update.

Furthermore, the display of WWW-documents from the Internet and the modification of local documents, e.g. documents from the local office area, is only possible with different applications. Current applications still handle information from the Internet and the local office area separately and applications for the office area can not handle the HTML-format sufficiently. Therefore, there is only little interaction possible between these applications and it is hardly possible to use documents of different formats within a similar context in a consistent manner, for instance associate a WWW-page with a local Word-document or a local video. There exist some tools which allow the iterative storage of whole hierarchies of WWW-documents with all their embedded objects but these tools also do not support the use of documents from other sources.

In the following we want to introduce the Information Management Tool (IMT) which solves many of these problems by storing and integrating WWW-documents and documents from the local office area into a consistent local information base. The IMT offers a common interface for a consistent access to different kinds of multimedia information and has a wide range of functionalities to collect, structure and associate all kinds of multimedia documents within the information base. Furthermore, it offers search facilities to find information on the Internet and within the local information base.

2 Concepts

In the following we describe the main underlying concepts and ideas for the Information Management Tool although not all of them have been completely implemented yet.

2.1 Local Information Base

Assume, we want to collect information on a certain topic which will be found in a large number of documents from either the Internet or the local office area. In order to allow an easy retrieval and reuse of found information we need some kind of common container where we put all documents with relevant information on the topic. Furthermore, we want to add some kind of meta information and relations to the documents, e.g. keywords, attributes or links between documents, to increase the overall information value of the information collection. In our context we want to call this combination of documents and meta information, which are stored in an extra database, the local information base (see Fig. 1).

Usually, a user will create several information bases which contain information on different topics. The IMT offers the interface to create and manage the information bases and
has all necessary functionalities to collect and import relevant documents. It stores and manages additional information, e.g. keywords and relations, in a consistent database and structured information from different information bases can be imported in another information base.

![Diagram of Local Information Base]

Figure 1: Local Information Base

### 2.2 Associating Information

The accumulation of a large amount of unrelated single documents into a single container is not suitable because it makes the retrieval of relevant information very difficult. Furthermore, a single document usually contains different kinds of information and the relevant part of these information depends on the context in which the document is presented. For example, if a picture of a soccer team is presented together with an article about a certain player only the part of the picture which shows the particular player is relevant in this context. Furthermore, the personal background of a user also determines which part of the information will be considered as relevant and different people will define different relations between documents.

The value of a document collection can be increased by defining relations between documents and sets of documents. In many cases we can gain more information if we put several documents in a certain context because we get the 'whole picture'. Relations between documents can differ in degree, e.g. a document which contains important information or only less important additional information in context with another document. Therefore, the user should be able to define different kinds of relations and associations between documents which are flexible and represent different degrees of relations. The IMT allows the definition of relations by means of a hierarchical container structure, a
2.2.1 Hierarchical Structure

A common approach to define relations between documents is mapping documents to a hierarchical container structure based on a set theoretical approach. Here, relations between documents are defined by grouping documents together in sets of hierarchical containers. The hierarchical structure consists of different container objects which contain either other container or documents. We use the following container objects and rules based on Hyper-G ([1]) to create the hierarchical structure:

- **Root**
  The root object is the main container of the local information base and its name is the name of the information base. The elements of the root object can only be other collection or guided tour objects. Every root contains also exactly one bucket.

- **Collection**
  A collection object is a container which elements can be collections, clusters, sequences or documents.

- **Cluster**
  A cluster object is a container which elements can only be documents (an instance of a document) which are elements of the same collection as the cluster.

- **Sequence**
  A sequence is a container which elements can only be documents which are elements of the same collection as the sequence. The documents in a sequence are ordered whereas the documents in a cluster are not.

- **Guided Tour**
  A guided tour is a container which elements are instances of documents. These documents are ordered as well but they can be from any collection.

- **Bucket**
  The bucket is a special container which is not really part of the hierarchical structure as such and contains only documents. It works as a temporary container for documents before they are put into the hierarchical structure of the local information base. Documents in the bucket do not have any relations to objects in the hierarchical structure.
The basic objects of the hierarchical structure are the document objects. They can be any kind of multimedia document of any format, e.g. a HTML-document, an image, a video or a text document. In our context a HTML-document including the embedded objects is treated as one document object, e.g. embedded images are not included as independent image documents into the local information base. Not all HTML-documents should be stored locally, e.g. WWW-pages with a database query interface. A document can only be an element of exactly one collection but several instances of the document can exist at the same time as elements of various clusters, sequences or guided tours.

If we reduce the hierarchical structure only to collections and documents we have the same hierarchical structure as the directory structure of a file system. Most people are familiar with such hierarchical structure and the additional container objects allow a more precise definition and presentation of relations between documents. The data about the hierarchical structure are stored and managed in the meta information database of the local information base to ensure a consistent hierarchical structure. If documents are moved or removed all corresponding structure information are automatically updated, e.g. if a document is moved from one collection to another its instances are automatically removed from clusters or sequences and updated in guided tours.

2.2.2 Link Structure

Although most people are familiar with structuring information hierarchical this approach is not flexible enough for some purposes. Therefore, the hierarchical structure of the documents within the local information base can be extended with a link structure. In our context a link is a bi-directional object which consists of a source object and a destination object. Links can be defined from a document to any other object of the hierarchical structure, except the root, the bucket and documents in the bucket. The position of a link at the source and destination document are defined by the anchor position. The complete link information are stored in a database independently from the linked objects. This approach has the following advantages:

1. Linking of any documents with any other document or container

   Unlike in HTML, link information are not part of the actual document and therefore documents of any type can be linked with any other container object or document type, e.g. a link from a soundfile to a text document or a collection.

2. Consistent link structure

   All link information are stored and managed in an independent database. Link information are automatically updated when a document or container object is moved or removed by the user. Therefore, a consistent link structure can be ensured and dead links which are a common problem on the WWW do not occur.

3. Following links backwards
Since a link consists of source and destination object the user can follow a link backward from the destination object to the source document. Therefore, a user can get a list of documents which are linked to a certain container or document and thus get an overview in which other contexts they are also used.

A link is determined by the source and destination objects and their anchor positions. Depending on the source and destination object, the anchor positions of the link can be either the whole object or a position within a document, e.g. a phrase in a text document or an area of an image. The source anchor position can be of type from and from within and the destination anchor position can be of type at or into. Therefore, we have 4 types of links as combination of the anchor position types. Since the anchor positions are stored independently from the documents it takes a greater effort to keep the anchor positions consistent when a document is changed. For example, editing a text document with links from within and/or links into this document requires a permanent update of all corresponding anchor positions until the document is finally saved.

Additionally, the user can also define a reference, e.g. a directory path or an URL, to an extern multimedia document which is not part of the information base. A reference is an unidirectional link where the referenced document can be removed without using the IMT. Therefore, we don’t have complete consistency here but some consistence checks can be implemented. For instance, every time an information base is opened dead references will be removed automatically when the corresponding documents no longer exist.

2.3 Information Retrieval

Apart from the hierarchical container structure and the link structure the local information base contains more additional meta information about the documents which are inserted manually by the user or created automatically by the IMT. These meta information allow a better retrieval of relevant documents from the local information base. Relevant keywords can be assigned to any document of the information base, e.g. keywords for images, sounds and videos. Additionally, a full text index stores the information from text and HTML-documents. Furthermore, meta information like creation date of a document or container, last access date to a document or object name are automatically created and managed in the information base. Special object types have additional attributes, e.g. the original URL of an imported WWW-document.

By means of these meta information we have the following options to search for relevant information in the local information base:

- Key Word Search

  Documents can be searched by certain keywords which have to be entered by the user. Since any document can have keywords, we can also search for images or sounds by keywords.
• Full Text Search

The integrated full text index allows a full text search in all text and HTML documents of the local information base.

• Attribute Search

Documents can be searched by their attributes, like last access date, document name, document type etc.

In order to find relevant information on the Internet which then can be imported into the local information base we need search facilities for the Internet as well. Instead of implementing yet another search engine it is appropriate to use existing search engines, i.e. the user defines the search query which is then sent by the IMT to existing search engines and meta crawlers on the Internet, like Altavista or HotBot. The results of these search engines are combined and evaluated and are presented to the user who then decides which WWW-documents should be imported into the local information base. This approach yields better results because the strength of the search engines can be combined.

2.4 Consistent Information Access

For an effective use of the local information base the user needs a consistent access to the documents and meta information. Therefore, all documents and meta information from the information base can be accessed through the interface of the IMT. Because new formats are developed and existing formats may change in the future we use a plugin concept to provide viewers and editors for different types of documents. The plugin concept allows a more flexible integration of viewing and editing capabilities for a wide range of different formats. Existing viewers can be exchanged by the user with his own viewers and viewers for new formats can be easily added. Especially the integrated WWW-browser who allows the import of whole WWW-pages into the local information base can be modified easily when there are new changes to the HTML-standard.

Existing applications which also offer viewing and editing capabilities for multimedia documents can be used within the IMT interface as well if they have an inplace interface, like OLE. Thus, the familiar interface of these applications can be used if desired and some problems with certain formats can be avoided. In order to keep the local information base consistent all imported documents and the meta information in the database can only be modified through the IMT interface.

3 Current Realisation

A first prototype version of the IMT has been implemented and serves mainly as a test bed to try out different approaches for the practical realisation of the concepts. The import
of documents from the local office area and the Internet into a local information base has been fully implemented. At the moment the IMT provides viewers and editor plugins for RTF-text, several image formats and HTML. Any other format which is supported by the OLE 2.0 server can also be viewed and edited by using the corresponding application within the IMT-interface. Documents in a guided tour or a sequence are presented with a dia-show approach which automatically chooses the appropriate viewer.

The user can structure the imported documents using the hierarchical container objects introduced in section 2.2.1 and create links from any documents to documents or containers. Local search facilities have been implemented whereas Internet search is still future work.

The prototype has been implemented for Windows95/WindowsNT with Borland Delphi 3. The meta informationen are kept in a Paradox database which is accessed via the Borland Database Engine (BDE) which can be freely distributed.

A (German only) version of the prototype can be downloaded from http://www.egd.igu.fhg.de/fhg.igu/abteilungen/a1/prototypen/imt.html

References