Logic and Semantic Hypermedia Data Models

Table of Contents

1 Introduction ........................................................................................................ 2
2 Problems with Node-Link Data Model ............................................................. 3
3 Logical Composites ....................................................................................... 7
  3.1 Data Types .................................................................................................. 7
  3.2 Browsing ..................................................................................................... 10
4 Semantic Composites ................................................................................... 15
5 Semantic Nets .................................................................................................. 18
6 Knowledge Domains ..................................................................................... 22
1 Introduction

In accordance with a so-called Basic Hypermedia Paradigm, multimedia documents (i.e. chunks of multimedia data) are called nodes. The cross-references or relations between nodes are called links.

Conceptually, a link is a connection between two documents. Links are directed, i.e. there is a source document and a destination document. In other words, a document may have a number of links emanating from the document, and there may exist a number of links pointing to the document.

Together nodes and links form what is called a hyperweb.
2 Problems with Node-Link Data Model

It should be especially noted that the basic node-link paradigm supposes that information on all links emanating from a particular node is stored as a part of such node definition. This situation is termed as embedded links.

The concept of embedded links has a number of important consequences:

- Links are unidirectional, i.e. can be traversed only in forward direction (i.e. from a source document to a destination document). The target node simply does not contain information on links pointing to it.

- All links emanating from a particular node, cease to exist when the node is deleted.

There are a number of well-known problems resulting from the simplicity of the Node-Link Data Model.

We can identify at least the following problems which are discussed in this section:

1. Editing of links is tedious;
2. Logical integrity is not supported;
3. Links are not context-dependent;
   - The presence of links rapidly leads to reader disorientation (“Getting Lost” syndrome;
   - The reuse of hypermedia materials is unsatisfactory.

In order to illustrate how editing of links is tedious. Consider, a tiny part of the whole hyperweb dealing with an alphabetically sorted list of persons. The documents can be accessed directly or browsed sequentially.

Suppose, a new document (“Information on Maurer”) has to be inserted into the list. In this case, manual editing of at least 3 different HTML files is needed.
In order to illustrate Logical integrity and dangling links, Consider the same part of a whole hyperweb representing an alphabetically sorted list of persons.

Suppose, the document "Maurer" is deleted for some reason. All references to such a non-existent document become so-called "dangling" links which point to nowhere.

In order to illustrate context-independent links. Consider, two lists of persons, the documents in the first list present employees of an institution called IICM. The second list presents members of a soccer team "Lions".

Note that two links emanating from the document "Codd" belong to two different contexts (i.e. "Employees of IICM" and "Members of the "Lions" team") but they are presented to a user simultaneously.
The presence of links completely unrelated to the current context rapidly leads to reader disorientation.

Suppose, a user starts browsing from the document IICM and further accesses a "first" employee "Codd". Now, the user may well "jump" to the document "Maurer" by clicking on the corresponding anchor "Next".

It immediately leads to user disorientation since the document "Maurer" has nothing to do with the current context, i.e. with information on IICM.

The reuse of hypermedia materials is unsatisfactory, because an author can refer to other documents but cannot embed them locally with possibly new or different hyperlinks.
Consider the same list of persons employed by IICM. Suppose all these three documents were developed by a first author "A". Suppose also that another part of the same database is prepared by another author (say, by author "B").

Note that the author "B" can refer to documents prepared by the author "A", but he/she cannot re-use them providing with new links as desired.
3 Logical Composites

Basically, idea of Hypermedia Composites is very simple. Links are separated from documents and stored as special data structures called *Hypermedia Composites* (such structures are also known as Collections and Packages).

3.1 Data Types

Thus, hypermedia documents can be connected into different navigational structures. We can say that an HM composite encapsulates a particular internal structure, and provides a particular context for links.

HM composites serve two purposes:

- defines a topology and context of links connecting its members into a navigable structure;
Logic and Semantic Hypermedia Models

- maintains the **integrity of such links** automatically.

Thus, HM composites automatically support a generic topology of links. Members can be inserted, modified or removed, the link topology is automatically updated.

Moreover, Hypermedia composites **can be reused** as members of other HM composites.

Let us summarize main principles of hypermedia data models based on HM composites:

- Hypermedia Database is made up of HM Composites.

- An HM composite encapsulates a particular **internal structure**, which is simply a set of pages or other HM composites (called **members**), one of which is distinguished as a so-called **head**. Head may be generated automatically as a summary dynamically retrieved from members (e.g. table of content).
• The members of an HM composite are interconnected by a particular link topology.

• Every HM composite has a unique identifier (i.e. name) for addressing.

The process of hypermedia development follows a bottom-up approach. At first, multimedia pages are created, for which all of the different types of media objects can be utilized. These bottom-level pages then build the basis for the definition of complex HM composites -- and therefore the overall structure of the hypermedia database.

In order to create an HM composite the author chooses the type of a generic link structure, assigns a name and selects a page which is assigned as a head.

In addition, there are several optional parameters which give the author the capability to refine the behaviour of the HM composite.

From what has been explained so far, it should be clear that the model provides sophisticated structuring facilities. Hypermedia resources can be kept modular and are easy to reuse. All collections
are created independently, but they are fully compatible via the flexible nesting mechanism. Therefore, multi-user development of hypermedia databases is greatly facilitated.

Let us now take a look at the main steps in editing of a hypermedia database:

- Step 1: Editing of pages
- Step 2: An HM composite is created
- Step 3: Previously created HM composites are re-used
- Step 4: The new HM composite is stored into database

Any specifications for an HM composite can be altered at any time and the deletion of a whole HM composite from the database triggers the automatic removal of this HM composite from all other HM composites of which it was a member.

Any set of HM composites can be saved separately from the actual database and incorporated into other hypermedia database -- thus providing an easy way of reusing and exchanging HM composites between databases.

3.2 Browsing
At any particular moment in time, the user can only navigate through a single, specific HM composite called a current container. Only members of the current container can be accessed during navigation.

If a page is accessed, it results in presenting media objects (i.e., some text, picture, audio, video clips etc.) on the screen.

There is a number of solutions for visualization of a HM composite

- it can present a special document called "label"
- it can present its head
- it can present a dynamically generated overview of its content

Orthogonal to the navigation by access and links in the current container is navigation by means of the operations **Zoom_In**. If a current member is an HM composite, it can be Zoomed in.

In this case the HM composite becomes the new current container. The head of this new current container becomes a new current member and link-based navigation within the new container becomes available.

Thus, links encapsulated within an HM composite, become available for navigation only when the HM composite has been "entered" by means of the ZOOM_IN operation.
As an example, consider a situation where HM composite "IICM" reuse a composite "Lions" as a member.

In this case, accessing the "Lions" as a member of "IICM" would provide just "Welcome" page with a possibility to open (i.e. ZoomIn) the "Lions" composite. When a user opens the "Lions" composite, for example, a list of the team members is visualized.
The **Zoom_Out** operation is the complement of **Zoom_In**. **Zoom_Out** restores the current container and current member to the state they had before the most recent **Zoom_In**.

Extending the functionality of **Zoom_Out** to give access to any HM composite of which the current HM composite is a member is provided by the operation **Zoom_Up**.

Together, **Zoom_In**, **Zoom_Up** and **Zoom_Out** provide users with the capability of navigating in a direction orthogonal to the conventional plane of link-based browsing.
Logic and Semantic Hypermedia Models

Basically, the Data Model replaces "spaghetti" like view of a hypermedia database offered by the node-link paradigm, with well defined structure consisting of a number of independent but fully-compatible hypermedia modules called HM composites.

The model provides the following advantages:

- **Links are encapsulated** within hypermedia containers called HM composites (they do belong to individual nodes). By definition, links cannot point outside a structured collection, but only between its members; hence structured collections represent well-defined chunks of information, which may be re-used in various contexts without concern for superfluous hyperlinks.

- The model compensates for the restriction of links to local contexts by **orthogonal browsing** semantics to switch local context (operations Zoom_In, Zoom_Out and Zoom_Up).

- Complex **data objects are re-used** rather than referred to. The distinction is apparent within the interface metaphor: a reference implies jumping to another location (with only one way back); re-use implies embedding the external object into the current context (with the possibility of later switching to another of the contexts within which the object is used).

- Authoring is carried out by **memberwise inclusion** of hypermedia chunks (i.e. HM composites) as opposed to "spaghetti like" linking of nodes/composites in other models. Any HM composite can be inserted into any other HM composite as a member. The possibility of creating recursive membership relations gives the entire model the capability to deal with whatever complex situations of hypermedia database modelling might be desired.

- All operations are **addressed to a particular data object** (HM composite) and do not affect link structure of other objects. In fact, this object- oriented granularity of the Data Model paves a new way for supporting logical integrity of hypermedia databases.
4 Semantic Composites

Unfortunately, logical (i.e. application-independent) composites cannot reflect specific features of a particular application what is necessary for building a user-centered hypermedia database. So-called semantic (i.e. application-dependent) composites are often more useful since they take into account specific features of a particular application.

The following components are required to build a particular Hypermedia application:

- a limited number of semantic composite types reflecting peculiar features of this application. For example, if we deal with an e-Learning application, we can speak about Course, Lesson and Chapter.

- a special rendering engine which process such data types and returns requested documents accompanied with a special environment.

Semantic composite types, by definition, cannot be defined once and forever similar to Logical composite types. Thus, a special authoring component (Data Definition Tool - DDT) is needed.
This tool is applied by a special person (say, WEB-Administrator) to define all types of composites needed by the content authors. Thus, we may speak about two levels of Data Definition:

Thus, the following components are needed to implement the model:

- Data Definition tool to define application-oriented composite types.
- Data Manipulation tool to fill out instances of the previously defined composite types with a particular content.
- Rendering Engine to map the composites onto a plain HTML model what is necessary to comply with existing standards and software packages.

We can also say that the DDT defines a library of semantic composite types to be further used by a DMT and a rendering engine.

Normally, DMT utilizes a most simple and productive drag and drop user interface.
Content authors simply drag and drop existing HTML documents into named cells predefined by the DDT.

The model is highly modular, for example, a library of semantic composite types may be used by a DMT without any knowledge on DDT. DDT may be used by advanced users as an additional tool to adjust the system functionality (if desired).
5 Semantic Nets

Web Based Knowledge Mining (Situation):
- A user needs material on a particular subject to acquire additional knowledge, and is aware about a WEB server containing a relevant information.
- The user access the server to find most relevant material, to work through these material.

Web Based Knowledge Mining (Solutions):
- search (engines and agents)
- semantic nets (browsing meta networks)

Semantic net is a graph consisting of nodes interrelated by arcs.
- **nodes** define particular concepts.
- **arcs** define semantic relationships which hold between concepts denoted by the nodes.

The most popular relationship between nodes is called "is a part of".

A semantic net may be seen as a meta description of a particular repository of documents.
A particular information resources may be attached to a number of nodes (we will call such nodes "Knowledge Cards").
Semantic net consisting of knowledge cards and semantic relationships, supports a resource infer procedure. We can say that
"if a resource is relevant to a knowledge card and this card is "a part of" another card, then the resource is relevant to the later concept as well".

In a general case, users browse the semantic network and identify knowledge cards (i.e. concepts) of interest.
The system automatically infer all the resources relevent to such selected concepts.
Note, authors do not need to search particular knowledge cards to attach their resources. A particular author simply defines a special card "area of contributions" which should be related to knowledge cards describing relevant concepts.

For example, an author dealing with hypermedia applications, may define his "area of contributions" as "is a part of" knowledge card "Hypermedia".

Now, all contributions of this particular author will be automatically inferred as related to hypermedia.

Web Based Knowledge Delivery (Situation):

- A user needs knowledge on a particular subject for a long-term perspective, and is aware about a WEB server containing relevant information, and periodically updated by the subject experts.

- The user accesses the server to configure his/her personal profile in such a way that
  - most relevant training material are automatically delivered to his/her personal desktop
  - he/she is automatically notified about such new materials
  - communication with the subject experts and other learners working on similar materials is possible via the desktop

Similarly, users do not need browse knowledge cards to find relevant resources.
A user can also define a personal knowledge card "area of interest", and refer to other Knowledge Card.

For example, if a user has a permanent interest in "Hypermedia" and "Network, he/she can define a personal knowledge card related to these two concepts. Whenever, the user enters the server, all relevant resources are inferred automatically.
6 Knowledge Domains

Generally speaking, Knowledge Profiling deals with imposing an application dependent data structure on a top of collection of Information Resources.

A typical example would include a company installing knowledge profiles for each of its employment positions. The process may be seen as defining a position accompanied with descriptions of knowledge components required for all positions. Moreover, the knowledge components may be further associated with training resources which provide this prescribed knowledge.

On joining the company an employee needs to be inducted into the practices and procedures of the organisation this will be facilitated by the new employee undertaking the associated training for the prescribed knowledge required for the position they hold.

New or trainee employees may always see what knowledge is needed (i.e. they can browse a position profile) and automatically access the training resources if any additional training is required.

Such "global" relationships between information objects may be defined as a collection of Knowledge Domains.

Each Knowledge Domain is a set of documents belonging to a number of predefined semantic categories.
A **knowledge domain schema** may be seen as a definition of all categories and all possible semantic relationships between them.

The knowledge domain schema defines common properties of all the category instances.

Any resource may be inserted (stored) into a particular knowledge domain as an instance of predefined category.

Thus, a responsible author simply selects an existing Knowledge Domain and a predefined category for a new resource and the system guides the author through the process of defining attributes and necessary relationships.
New "Knowledge"

Description:

Needed:
- Project Manager
- Multimedia Developer

Explains:
- Quality Management
- Project Management
- Multi Media

[Save] [Cancel]