Hypermedia Systems: the Need for Cognitive Hypermedia Models[1]

Lina García-Cabrera [1], María José Rodríguez-Fórtiz[2], and José Parets-Llorca[2]

[1] Departamento de Informática, E. P. S., Universidad de Jaén
Avda. Madrid, 35. 23071 –JAEN (Spain)
lina@ujaen.es
[2] Departamento de L. S. I. ETS Ingeniería Informática. Universidad de Granada.
Avda. Andalucía, 38. 18071 –GRANADA (Spain)
jparets@ugr.es

Abstract. Most of the current hypermedia systems models focus more on edition and document navigation through prefixed links than on the dynamic construction of the document. We consider that hypermedia systems should be information systems, which offer support to the structuring and access processes, according to the *conceptual associations* that may be established between their different information items. That is to say, the building process of hypermedia must necessarily be based on a cognitive model. It allows an explicit representation of the semantic content that facilitates the development of tools, which support development, maintenance and navigation activities. In this paper, we shall attempt to justify the need for a flexible cognitive model in the conception of hypermedia systems. An introduction to a semantic-dynamic model is presented that provides a complete, adaptive and evolving control of the development and maintenance of hyperdocuments and an understandable navigation.

1 Hypermedia Systems are Ill

It is true, hypermedia systems *are severely damaged* [11, 23], *but their injuries are not deadly* [26]. A certain collapse in hypermedia research today has been provoked by the blurred-line between the models used to represent information and the models for presentation or edition. On the one hand, the existent reference models [6, 12, 17, 20, 10], as Schnase says [30], *"tend to focus on abstracting the connectivity of hypermedia* (links) *from its underlying information* (nodes) *rather than abstracting structure from functionality"*. On the other hand, the static representation of the information

---

[1] This research is supported by a project –MEIGAS- by the Spanish CICYT (TIC2000-1673-C06-04) which is a subproject of the DOLMEN project (TIC2000-1673-C06).

network used by today's hypermedia systems make them unusable in authoring systems and frameworks which attempt to gain some advances in structuring, presenting and accessing complex, dynamic and interrelated information.

From our point of view, models which support the abstraction of the structure, information and behaviour of hypermedia systems by means of semantic techniques are needed. In addition, these models must hold the current and predictable characteristics of a high-level hypermedia system (which Bieber et al. [3, 4] call hypermedia systems of the third and fourth generation). These high-level architectures support both author and reader activity. From the author's perspective they should provide a flexible control of the development and maintenance of the document. From the reader's perspective, navigation should help to understand, access and choose the presented material.

This conclusion will be justified in the next section stemming from an analysis and criticism of the suggested models for developing hypermedia system in the research literature. Section 3 presents the different approaches used in structuring hypermedia systems. Their main advantages and drawbacks will be discussed. In section 4 the previous conclusions will be applied in obtaining the characteristics that a model for developing Hypermedia System must have. Section 5 will present and describe a general architecture of the proposed Model based on systems as an alternative to Models based on a set of hierarchical levels. Finally, section 6, summarises the conclusions about the analysis and criticism made.

2 Hypermedia Models

Hypermedia systems are older and more intense than the World Wide Web. In fact, three hypermedia generations can be distinguished [16, 29]. The first one comprises hypertext systems based on mainframes. These offered support for multiple collaborative users but, at the same time, presented serious limited navigational help and graphics were not supported. In the second generation we found systems based on workstations and PCs, for single users or small groups including advanced user interfaces, multimedia information and graphical navigational help.

Current hypermedia systems belong to this second generation, resulting in closed systems with storage mechanisms and no interpretability. In 1987 the third generation began research oriented to the development of prototypes that tried to include the conceptualisation of hypermedia systems by means of abstract models. The reference models HAM, Dexter and Trellis belong to this effort. In this stage prototypes were created with the aim of supporting structuring mechanisms with composed nodes. The main objective of last generation, in which we are now immersed, is to achieve the incorporation of hypermedia features into software and information systems in order to provide their users with an associative way of accessing, analysing and organising information, i.e. the integration of hypermedia functionality [1].

In this section, we present and analyse two ways of conceiving a hypermedia model. The first , from a more classic point of view, models a hypermedia system as a hierarchy of levels. The second and more ambitious attempts to take into account the problems presented by the second generation of hypermedia systems and tries to achieve an interpretation of their structure.

2.1 Hypermedia Reference Models

From the third generation of hypermedia systems (late eighties) different Reference Models have been proposed with the aim of converting them to open systems and integrating their functionality in whichever framework or application. These models describe every conceptual element that, from the point of view of their authors, a hypermedia model contains. The five more extended ways of understanding and modelling these systems are described below.

HAM or **the Hypertext Abstract Machine (1987)** [6] (*figure 1*) was the first attempt to express a hypermedia system by means of an abstract model. The HAM does not describe the full hypertext system, only the HAM objects and their applicable operations. The HAM is the abstract level, which is at the top of the storage system, and it manages and provides the hypermedia information to the applications and user interfaces. The HAM model is based on five types of objects: *graphs*, *contexts*, *nodes*, *links* and *attributes*. The HAM maintains a history of these objects, allows selective access through filtering mechanisms (by means of expressions based on objects attributes and their values), and includes data access restriction mechanisms based on ACL (*Access Control List*). A *graph* is the highest level object and it contains one or more contexts. *Contexts* are subsets of *nodes* connected by *links* to a hyperdocument. *Attributes* can be attached to contexts, nodes, or links representing application-specific properties of objects or containing information that further describes an object. The model only describes two of these attributes: identifier and version based on the creation or updated time of an object.

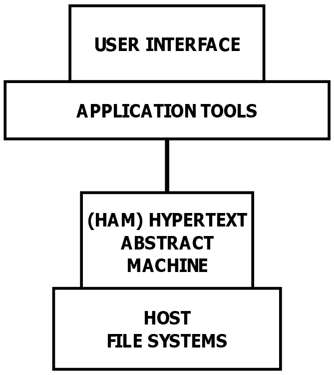

Fig. 1. HAM: Hypertext Abstract Machine [6].

Richard Furuta and P. David Stotts propose a meta-model for the **Trellis System** [12] (*figure 2*) based on Petri Nets. This meta-model distinguishes five logical levels (*Abstract Component Level*, *Abstract Hypertext Level, Concrete Context Level, Con-*

crete Hypertext Level y *Visible Hypertext Level*). Within each level one or more representation of part of the hypertext can be found. In contrast to the HAM, here the levels represent levels of abstraction, not components of the system.

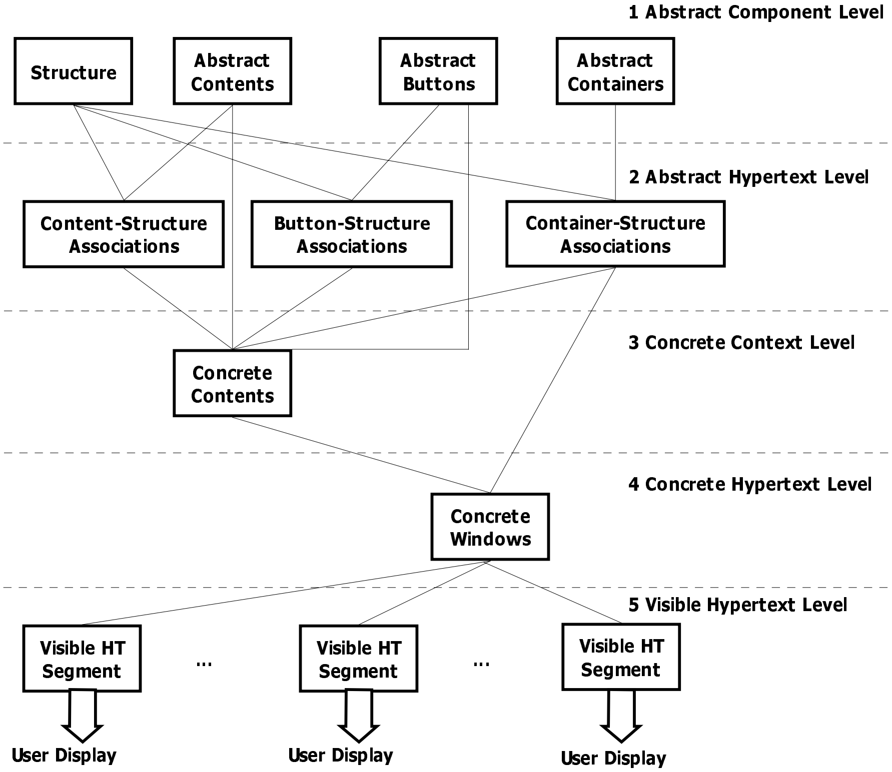

Fig. 2. Trellis' Metamodel [12].

The first level presents the components (*structure*, *abstract contents*, *abstract buttons and abstract containers*) that will be associated in the second level to form the hypertext. In addition to traditional nodes (*abstract contents*) and links (*abstract buttons*), the Trellis system supports two more elements: *structure* and *containers*. The first merely describes a skeleton of the graph which provides *placeholders* that will be associated with the hypertext's abstract contents and relationships. The containers are an abstraction of how the information pieces of the hypertext will be aggregated and combined for display together. The associations between the elements of the Abstract Component Level are made in the Abstract Hypertext Level. They can be content-structure, buttons-structure and containers-structure associations.

The Abstract Hypertext Level describes these associations but does not describe how these associations will be displayed. The mapping between the abstract hypertext and windows will be made in the Concrete Context Level. This indicates how a particular piece of information will be displayed, or the details of operations derived from a link navigation. Finally, the fourth and fifth levels specify visible presentation of the document on a particular user interface.

In this model the separation between components, and associations between components, on one hand, and their implementation by Petri nets, on the other hand, achieve a dynamic adaptation of the appearance and behaviour of the hyperdocument when it is navigated. According to Stotts and Furuta, a hypertext document has two layers -a fixed underlying information structure that is created by the hypermedia author and a flexible structure that is superimposed on the former and is tuned to each user's requirement-.

Perhaps the most referenced model is **Dexter's** model [17] (*figure 3*). Its main goal "*is to provide a principled basis for comparing systems as well as for developing interchange and interoperability standards*". The model is divided into three layers: *runtime layer*, *storage layer* and *within-component layer*. It focuses on the *storage level*, which describes structure as a finite set of components. A component can be a node, a link or a composed entity formed by others components (in fact acyclic directed graphs). Their internal structures within the components are responsibility of the w*ithin-component layer*, the interpretation of which must be made by the applications that provide them (Dexter's model treats within-component structure as being outside the hypertext model *per se*). It only is concerned with the definition of the interface between this level and the storage level, named *anchoring*. This intermediate element takes charge of addressing items within the content of a component in an indirect way. The model also establishes an interface between the storage layer and the runtime layer, named *presentation-specifications*. This contains information on how a component/network is to be presented to the user. The way in which a component is presented depends on the specific hypertext tool that is doing the presentation, but can also be a property of the component itself and/or of the access path which arrives at that component.

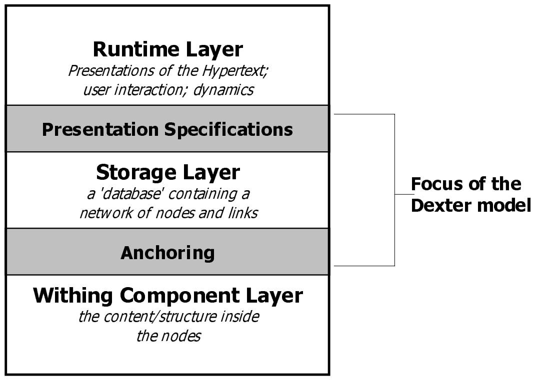

Fig. 3. Dexter's Model [17].

Like Dexter's model, the **Formal Hypertext Model** of Danny B. Lange [20] emphasises the data structure of hyperdocuments. This data model defines nodes, links and network structures of nodes. The model goes further than Dexter's model in looking inside the nodes of a hyperdocument. It defines slots – which can be compared to records of data in a programming language- and fields –sequences of characters inside a slot-. The model allows the possibility of referring to nodes, slots inside a node or fields inside a slot of a node. The model also considers the possibility of distinguishing between node types and link types by means of attributes and values. But this model does not include the representation and the interpretation of navigation. From the point of view of the author, this work should be conducted by the applications that operate with the hyperdocument using specific operations. In addition, the data object does not have to be reliable about these applications and their semantics. The data model is implemented as a persistent object-oriented database, solving, in this domain, issues like distribution, version management and access control.

De Bra, Houben and Kornatzky propose a more general object-oriented model that they call **Extensible Data Model for Hyperdocuments** [10] or Tower Model. This model "*is based on separation of concerns between the fixed aspects of a hyperdocument that would always be present, and the variable part that is extensible*". The authors distinguish two layers. The *lower layer* defines kinds of first-class objects as nodes, links, and anchors. The *second layer* is in charge of modelling constructors which allows more complex information representations. There are three ways in which a complex object can be built from its components:

- The *composite object constructor* models information units from collections of nodes, links, and anchors. Essentially it builds graphs and, "*intuitively, a composite object can be viewed as a template containing holes into which the components are plugged*".

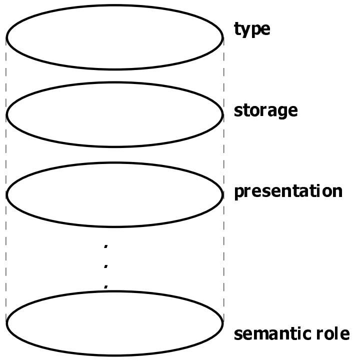

Fig. 4. Tower Constructor [10].

- The *tower constructor* (*figure 4*) packages together the multiple levels on which an object within a hyperdocument is described. For example, a node would have a structural dimension consisting of its content (e.g. text) and the operations to manipulate it (e.g. a text editor), and a presentation level describing its appearance on the screen. Another level would be a semantic role of its attributes. The number and nature of the levels of a tower object is arbitrary, depending on the information which is supported by applications outside the hyperdocument.

- The *city constructor* binds together multiple views or perspectives of an object. That is to say, views correspond to different roles of the same object. The difference between the tower and the city constructors is that the first gives different descriptions of one object, whereas the second provides different descriptions of basically the same information.

Using these constructors, the model offers an interface level that supports the integration of different information sources and, in addition, the creation of virtual or computed structures from descriptions of its components.

As can be observed, all these models describe the possible conceptual elements that can be found in a hypermedia system. They consider different abstraction levels although many of them only describe completely the storage level. In general, all of them agree in the definition of:

1. a first storage layer in which nodes, links and anchors are represented in a separated way. Some of them also include composed nodes, structures and composed structures;
2. a second layer which maps the *front-end* or hypermedia system elements with the models, variables and calculations of the *back-end* or application objects; and
3. finally, the representation level of the hypermedia structure by means of a user interface.

From our point of view two important consequences can be concluded from the previous discussion:

1. Hypermedia reference models focus more on edition and document navigation through prefixed links than on the dynamic construction of the document.
2. Most of them shyly admit the possibility of defining some attribute type associated with nodes and links that in a certain way describe its behaviour. But none of them establishes, in a definitive way, what the attributes are (HAM, Trellis, Tower Model) nor considers that this description is the responsibility of the applications that provide information objects (Dexter, Formal Model of Lange). In the best case, they define attributes that are directly related to the presentation of the document (i.e. a link to the top of a page, or inside a page).

2.2 Cognitive Models

As an alternative to the Reference Models there are models whose nodes and links are semantically typed in order to help authors to organise information more effectively and give better navigation for readers. This kind of model is scarce, but many specific framework and applications which include a taxonomy of nodes and links can be found.

The only generic model we have found in hypermedia research literature is the suggested by Rao et al. [28] based on Guilford's Structure of Intellect model. Its general morphology for hypermedia systems classifies nodes into six different semantic types and links into twelve different types. The first ones represent pieces of knowledge or ideas (i.e., *detail, issue, proposition*); the second ones represent relationships between ideas. Links are further classified into convergent links and divergent links. Convergent links (i.e., *specification*, *inference*) are those that focus on or detail a particular thought, whereas divergent ones (i.e., *elaboration*, *opposition*) expand or broaden new ideas.

This model presents two large disadvantages: a) only authors and developers [4] but not readers take advantage of this taxonomy and b) the conceptual world considered is so wide that we have serious doubts about the usefulness of a so generic semantic which finally will always be too ambiguous. These problems are not present in those systems which include particular cognition models such as, for example, the gIBIS system (*graphical Issue Based Information System*) [8] to support argumentation dialogues among a team of software designers. Its conceptual model uses three node types: *issues*, *positions* and *arguments* and also relation types that connect some node types (i.e., between *issue* and *position* relations can exist such as *is_suggested* or *questions*). In such a way, the system keeps design decisions and also underlying design rationale [31]. That is to say, it makes a representation of the processes of reasoning and decision about a set of related issues. The gIBIS model is unquestionably very practical and effective, but only in the conceptual domain for which it was designed.

2.3 Why a Cognitive Model?

As can be seen, the previous Reference Models consider the final hypermedia documents and, sometimes, the navigation conducted by the reader. Nevertheless, the design, construction and maintenance processes of the hypermedia by an author is not considered sufficiently. However, this development process is very important because it implies a structuring process that is implicit and diluted inside the document.

Furthermore, the integration of the hypermedia functionality [13], the main aim of the Reference Models, is not possible without understanding and controlling the information system. Without a consensus about how we can identify the objects of a system, and how these objects can be combined it is not possible to achieve integration with other applications, because we do not know how we can reference them and what can be made with them.

In fact, some authors such as Bieber [2] believe that hypermedia systems are an interesting interface to be integrated into dynamic systems like, for instance, knowledge-based decision support systems. Can this proposal be reversed? Why we don’t incorporate a knowledge system that allows an intelligent control and structuring of the hypermedia system?

We believe that cognitive models can allow the expression of the design rationale making semantics explicit and allowing us to structure the information system. Furthermore, if semantics is made explicit during the building process, it can be shown during the navigation process. Such semantic incorporation makes possible the control of the information system and its maintenance and integration into any computed framework. An argument in this sense will be developed in the remaining sections.

3 Finding Out the Semantic Structure of an Information System

Until now our basic claim is the need to build a Cognitive Model for hypermedia systems. But **how** can we find out the semantic structure of an Information System? Two possible answers to this question can be found in the literature. The first implies that the conceptual structure of a document can be discovered by exploring and analysing it, i.e. the structuring process can be achieved by an examination of the documents. On the contrary, we can consider that the author should represent this process and the documents should be obtained from the cognitive model provided. In the following paragraphs we will examine both assumptions (see table 1).

From Document to Structure	From Structure to Document
Analyses of Spatial/Visual Aspect	Universal Ontology
Analyses of Content/Natural Language	Multi-local polyhedric Ontologies

Table 1. How to find out the Semantic Structure

3.1 From Document to Structure

Two possibilities can be used in finding semantics from hyper-documents: 1) Analysing the visual or spatial aspects of the documents [22], 2) Analysing their content.

The first approach implies a basic assumption: meaning is embedded in syntax. This implies that the edition properties of a hyperdocument (style, fonts, paragraphs, links, etc.) reflect the cognitive schemes and conceptual associations of concrete knowledge domains. Although this assumption can be valid for documents with well known or standard patterns (reports, logistics, timetables, programs, ...) in most cases the information pieces are partial, ambiguous and fuzzy. As Kaplan says [19] visual or graphical representations only show one aspect of a virtual space. Hypermedia allows us to operate in a space different from the geometric space: the semantic space. The architectural space of the documents and the semantic space managed during their conceptual development are weak correspondences. The reasons for this are obvious. Most of the final presentations obey aesthetical, pedagogical, cultural or linguistic aspects, and in the worst case, technical or implementation constraints. The analysis of several complex hyperdocuments shows that this process is very difficult also for humans[2]. This is especially so because the structuring information is not recorded anywhere; we have the consequences of the structuring process but not the causes that provoked them.

---

[2] We have studied the network of links of the Virtual Memory Tutorial (CNE Modules Tutorial Central, http://cne.gmu.edu/modules/modules.html) in order to interpret relationships between the different information pages. We can not do anything because: a) the only information that we can met about pages is the name of the archive but anything about his content; b) relationships, many times, make reference to a portion of the page and not to the whole content of the page.

The second approach implies the understanding of natural language enriched with graphical features. Although this solution can be considered, we believe that a complete representation of knowledge in all human domains is not possible, at least in the near future.

3.2 From Structure to Document

This alternative implies that the document authors build and structure their own conceptual domains. According to some kind of conceptual nets they can characterise their information domains and tailor their documents. This implies that authoring systems should provide open ontologies [33] which allow the semantic structuring and description of the hypermedia information system.

What should be the scope of one such ontology? Do we have to search for a universal, standard and formal conceptualisation which integrates the possible ontologies? Unfortunately, "*only a local context can motivate a given ontology which aims at schematizing a conceptualization*". "*Knowledge in a domain could be extracted by understanding and generalizing the conceptualizations inherent in the lexical repositories of a domain*". "*Taxonomic sources and context, select some issues of the global conceptualization, like pointing the finger to a site in the mind of an ideal, intersubjective expert*". "*Put differently, we need to move from local heterogeneity of intended meanings to multi-local, polyhedric intended meanings*" [13].

The conceptualisation of an information domain is only possible in a concrete context. Apart from linguistic and cultural aspects, a knowledge domain depends on the use of the information. And like Hendriks and Vriens [18], we consider that due to the dynamic nature of knowledge assets we must take into account tacit knowledge, involving such intangible factors as personal beliefs, perspectives and values, and embedded in individual experience. Furthermore, the same information organised in different ways or used with different objectives will require different conceptual structures. Specifically, the use of information will determine:

- The set of relationships and concepts registered.
- The degree of structuring of these concepts
- The type of checks which should be carried out in a semantic space.

And, as a consequence, the representations (more or less formal), which are required in a knowledge domain.

4 Basic Ingredients in Building a Hypermedia Model

According to the Collins Cobuild English Language Dictionary [7] *a model of a system or process is a theoretical description that can help you understand how the system or process works*, or how it might work. This implies that if we want to model hypermedia systems we cannot forget the raw material we are working with, and the characteristics that describe its behaviour and that help us to understand it.

Hypermedia systems are information systems which offer support to the structuring and access processes, according to conceptual associations that can be established between their different information items, i.e. as Vannevar Bush announced, *“the process of tying two items together is the important thing”* [0]. Unfortunately, much of the current tools and authors do not take into account this fundamental basis. They have forgotten the politics of the structuring process, marvelled by the mechanism to establish and to follow links. The *primacy of structure over link* [26] could convert information systems based on links between information chunks into real knowledge systems based on structured information items [24].

But you cannot structure –and, if necessary, change the structure of– any particular piece of information if there is no identification of their components and a set of rules that tell us how to organise them. The building process of a hypermedia model must necessarily be based on a cognitive model and only semantics can allow us the establishment of links between the different pieces of information in a conscious –not rash or capricious– way. Also, making the context and the relations of any item of information explicit avoids *disorientation* and *cognitive overhead* [9] and allows updates and adaptations to respect and clarify the underlying conceptual structure.

Tasks	Reasons for Cognitive Model	Users
Development	Incremental design process Collaboration between authors	Author
Maintenance	Representation of Design Rationale	Author
Navigation	Provide a contextual access Support human understanding	Reader

Table 2. Reasons for Adoption of Cognitive Models

From a more tangible point of view, let us think about the users of hypermedia systems and hear their demands (see table 2):

1. *For the author,* the elaboration of hyperdocuments is a difficult task, which includes a lot of changes, additions and updates. In addition, it is frequently carried out by different authors. This implies that hypermedia models *“must support the designer’s incremental and opportunistic activity all along the design process”* as Nanard [25] says. A good maintenance of information systems is only possible if the author saves information about how the document has been conceived and how it can be explored according to the author’s criteria. Furthermore, a representation of the Design Rationale [31] or process of underlying reasoning and decision should be carried out.
2. *For the reader,* hyperdocuments are information systems that may be navigated. Readers are interested in knowing what kind of information is being offered at each moment, what has been offered until now, and what can be offered in the future. For them, hyperdocuments are –or may be– sources of information that facilitate their knowledge comprehension.

Previous argumentation implies that a hypermedia model must be *dynamic*, i.e., it should allow the representation of the design process and of the hypermedia activity that let us structure the information net in a concrete way. But this dynamism is not possible without making the semantic structure explicit, because it determines the *evolution* [27] and *navigation* possibilities in an hypermedia system.

But, by running away from absolute permissiveness in establishing links among two nodes, we cannot fall in the opposite corner, i.e. absolute rigidity. The use of fixed structures that respond to a particular cognitive model (gIBIS [8], SEPIA [32]) make its use unworkable by the author outside the intended domain because these structures impose serious limitations. Following Marshall et al. [22], we assume that a descriptive system, where the author was able to characterise his domains of interest and could use his own abstractions in order to limit and structure his information systems, is more interesting. As a consequence the system must provide mechanisms which allow the incorporation and representation of the author's ontologies [33] or conceptual domains.

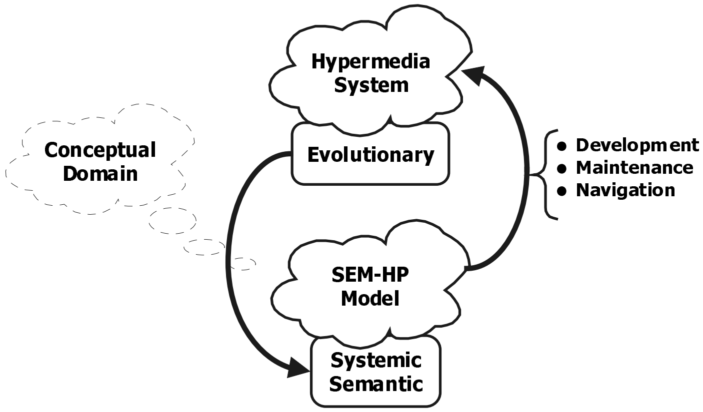

Fig. 5. Basic Ingredients of a Model for devolopment, Maintenance and Navigation of Hypermedia Systems.

From the previous discussion the following conclusions can be deduced (*figure 5*):

- The building of a hypermedia model requires us to know:
  - Which elements integrate it,
  - What is the functionality or types of services offered by these elements and,
  - What kinds of communication and material must flow between each one of these elements.
- The model must help and make flexible the construction, maintenance and navigation of the hypermedia systems. These three key aspects are exposed to continuous changes and updates and the model can be able to integrate them.
- An explicit semantic representation must permeate the model. The possibilities of structuring and further changes, adaptations or evolution will depend on the level of explicitness of this semantic representation.
- The model must offer a flexible semantic representation that allows the author a characterisation of his own information domains

5 A Semantic-Evolutionary Model based on System Theory

In our opinion, the traditional skeleton of hypermedia models, based on a set of hierarchical levels that can be translated into a sequential and static methodology, is not the best approach in representing a complex and changing reality, where construction, maintenance and use (navigation) are confused by their strong interrelationships. In these models there is no correspondence between structure and functionality. From our point of view, a functional systemic perspective is more suitable. A hypermedia system can be conceived as a set of interacting systems in continuous evolution.

Le Moigne [21] defines a system as: "something (no matter what, if we can identify it) that inside something (environment) in order to something (finality or project) makes something (activity = operation) by means of something (structure = stable form) that is transformed along time (evolution)". This definition and the further developments of the author imply that offering different representations of a certain complex reality is necessary, as a functional approach is one of the most important in understanding the activity of a complex system.

We assume that any Hypermedia System can be modelled by means of interrelated and interacting systems. Each one of those systems will recover some of the characteristics of the model, and thus will offer certain functions to the rest of the systems and, as a result, to the users -authors and reader- that interact with them. One such approach allows the construction, maintenance and navigation of information systems in continuous evolution and makes these activities more feasible, understandable and flexible. Semantics will make this possible.

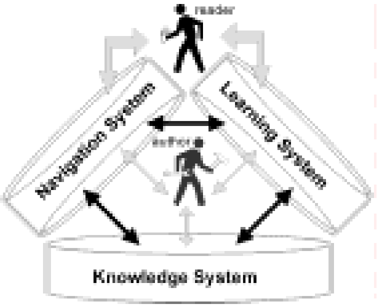

Fig. 6. Semantic-Evolutionary Model based on Systems. Different systems interact among themselves -black arrows-. The Reader interacts with the Navigation and Learning Systems, while the author interacts with the three systems -grey arrows-.

A Hypermedia System can be conceived as composed by three systems: The Knowledge System, the Navigation system and the Learning system (Figure 6). The Knowledge System is in charge of the storage, structuring and maintenance of the different pieces of information. It memorises the knowledge acquired about the information system that is represented. This knowledge will guide the design and structuring processes of the information system. It will determine the possibilities of transformation and change of this structure throughout its evolution. The Navigation System helps the reader in his interaction with the information system. Using the knowledge base and the reader activity through time in a dynamic way, this system determines –firstly– the accessible information and –secondly– its interaction possibilities. Finally, The Learning System optimises the knowledge acquisition process from the hypermedia system adapting navigation to the information needs and to the knowledge gained by the reader.

5.1 Making the Hypermedia Adaptive and Evolutionary

In order to highlight the adaptive aspects of the model we focus on those artefacts, which stress the cognitive and evolving aspects of a hypermedia system: conceptual structure and preconditions. The complete model can be found in [14, 15].

The set of concepts in a Conceptual Domain, which identify the information items, constitute a graph which contains the relationships and dependencies between concepts. We will call this graph *Conceptual Structure*, with *information items* being the building blocks for information used in the information system. *Relationships* between concepts are domain dependent and must be defined by the author for each particular conceptual domain, i.e. the author provide his own ontologies [33]. These ontologies (concepts & relationships between concepts) define a dictionary of keywords which is used by the author in order to provide the structure and by the reader in order to select material. In addition, *generic dependencies* between concepts, which are domain independent and have a generic character, may be considered: *aggregation (partOf)*, *instantiation (isA)* and *specialisation (aKindOf)*. The dependency *partOf* allows hierarchies between concepts. The dependency *aKinfOf* allows the composition of information items. Relationships and dependencies between concepts allow the definition of the *concept environment*, i.e. the set of concepts which has relationships or dependencies with another concept. The notion of environment allows some interesting operations which are known in the literature as *queries based on the structure*. The previous dependencies allow the dynamic creation of computed documents, i.e. the readers can construct new documents by means of this explicit semantics. Relationships and dependencies also guide the authors during the construction and maintenance because they can suggest some structures and associations in a concrete information domain, i.e. it helps the author in Validity and Relevance checks.

Preconditions guide the development, maintenance and navigation of the hypermedia documents. They are provided by the different Systems and are always applied by the Navigation System. They limit or constrain the *associations* between information items that can be used during navigation. In a dynamic way a set of preconditions will hold for each information item and they will limit the set of associated items. We will call this set the *information item framework*. Two types of preconditions can be distinguished:

1. Derived from the semantic structure of the information system. Obviously navigation will be restricted inside the *world* conceived and designed by the author. The aspects which are useful in establishing preconditions are: The Conceptual Structure of a Conceptual Domain of which an information item is

member; Generic Dependencies between concepts; functions that an information item may play in the context of an information system, and the language of an item.

2. Topics derived from the navigation itself and which provide a better adjustment of the structuring process: the type of navigation, the navigation carried out by the user over time or considerations about security and access control.

The possibility of adding preconditions implies adaptations and changes in the hypermedia system. These preconditions are described in a temporal-descriptive logic language which supports expressions such as: "*if before ... and after... then show...*", "*take into account whether the reader knows this or that concept", "whether the reader has made this tour.... then this and that item can be shown*". These kinds of rules determine, at all times, what pieces of information can be activated and what are the information items that can be searched. These rules are provided by the hypermedia author and are indirectly selected by the reader when he specifies a navigation type or navigates in the system. In addition, browsing implies feedback information for the Navigation System and, sometimes, also for the Learning System. That is to say, the applicable preconditions or information item framework is dynamic over time and depends on previous navigation carried out by the reader.

In the model presented, each System redefines, to some extent, the knowledge base provided by the Knowledge System which is stable for the reader but dynamic for the author or authoring tool. Each System is supported by itself and contributes with additional information. This information will say what information items can be consulted and under what prism. The different systems interact between them and their interaction produces, in a dynamic way, adaptations within them.

6 Conclusions

The analysis of different models of hypermedia systems proposed by the research literature shows that they are not able to represent the development, maintenance and navigation processes of an information system in continuous evolution. Hypermedia reference models focus more on edition and document navigation through prefixed links than on the dynamic construction of the document. Most of them shyly admit the possibility of defining some attribute type associated with nodes and links that in a certain way describe its behaviour. But none of them establishes, in a definitive way, mechanisms to define and represent the semantic structure of hypermedia systems.

Therefore, hypermedia systems should be based on cognitive models. They allow an explicit representation of the semantic content that benefits authors and readers and facilitates the development of tools, which support development, maintenance and navigation activities. These cognitive models, in a flexible way, provide, incorporate and represent the author's ontologies and support the adaptive and evolving activities of the hypermedia system.

Consequently, a Hypermedia System must be based on a semantic-evolutionary model. This model is a set of interrelated and interacting systems called Knowledge

System (made up by the Memory and Presentation Subsystems), Navigation System and Learning System. These systems allows: a) an easy and flexible development and maintenance of hypermedia documents, b) providing a representation of the conceptual structure and dependencies between them, c) more than one representation of the information system –of a set of possible representations - and, d) a dynamic navigation where is possible multitarget, multiproposal navigation with structural contextualisation.

References

1. Ashman, H.L.; Balasubramanian, V.; Bieber, M.; Oinas-Kukkonen, H. 1996. Proceedings of the Second International Workshop on Incorporating Hypertext Functionality Into Software Systems (HTF II), Hypertext'96 Conference. [On-line] http://space.njit.edu:5080/HTFII/Proceedings.html
2. Bieber, M. 1991. Issues in Modeling a "Dynamic" Hypertext Interface for Non-Hypertext Systems. Hypertext'91 Proceedings ACM, Dec 1991, pp. 203-217.
3. Bieber, M.; Vitali, F. 1997. Toward Support for Hypermedia on the World Wide Web. IEEE Computer, January 1997.
4. Bieber, M.; Vitali, F.; Ashman, H.; Balasubramanian, V.; Oinas-Kukkonen, H. 1997. Fourth Generation Hypermedia: Some Missing Links for the Word Wide Web. [On-line] http://www.cs.unibo.it/~fabio/bio/papers/1997/IJHCS97/issuev57.html
5. Bush, V., 1945. As We May Think, The Atlantic Monthly, July 1945, 176, 101-108. [On-line] http://www.isg.sfu.ca/~duchier/misc/vbush/.
6. Campbell, B., Goodman, J.M. 1988. HAM: A general purpose Hypertext Abstract Machine, CACM 31:7, July 1988, pp 856-861.
7. Collins W. 1992. Collins Cobuild English Language Dictionary. HarperCollins Publishers.
8. Conklin, E. J.; Begeman, M. L. (1989). gIBIS: A Tool for All Reasons. Journal of the American Society for Information Science 40(3), 200-213.
9. Conklin, J. 1987. Hypertext: An Introduction and Survey. IEEE Computer, 20(9), 17-41.
10. De Bra, P., Houben, G.J., Kornatzky, Y. 1992. An Extensible Data Model for Hyperdocuments, 4th ACM Conference on Hypertext, Milan, December 1992, pp 222-231.
11. DeYoung, L. 1990. Linking considered harmful. Proceedings of ECHT 90, (Versailles, France, Nov), 238-249.
12. Furuta R., Stotts, P.D. 1990. The Trellis Hypertext Reference Model, NIST Hypertext Standardization Workshop, February 1990, pp 83-93.
13. Gangemi, A.; Steve, G.; Giacomelli, F. 1995. ONIONS: an ontological methodology for taxonomic knowledge integration. Working notes of the workshop Ontological Engineering. ECAI'96, 29-40.
14. García-Cabrera, L.; Rodríguez-Fórtiz, M. José; Parets-Llorca, J. 2000. Toward a Formalisation of Evolutionary Hypermedia Systems based on System Theory, Computer Aided Systems Theory, EUROCAST'2001. In Proc. of EuroCAST'2001 (Eight International Conference on Computer Aided Systems Theory), LNCS 2178. Springer-Verlag, 2001.
15. García-Cabrera, L., Rodríguez-Fórtiz, M. José, Parets-Llorca, J.: Formal Foundations for the Evolution of Hypermedia Systems. 5th European Conference on Software Maintenance and Reengineering, Workshop on FFSE. IEEE Press. Lisbon, Portugal, March (2001) 5-12.

16. Halasz, F. 1988. Reflections on NoteCard: Seven Issues for the Next Generation of Hypermedia Systmes. Communications de ACM, July 1988.
17. Halasz, F., Schwartz, M. 1990. The Dexter Hypertext Reference Model, NIST Hypertext Standardization Workshop, February 1990, pp 95-133.
18. Hendriks, P.H.J., Vriens, D.J. 1999. Knowledge-Based Systems and Knowledge management: Friends or Foes?. Information and Management, February 1999, Vol. 35, Nº2, pp 113-123.
19. Kaplan, N. Where no Mind Has Gone Before: Ontological Design for Virtual Spaces. ECHT 94 Proceeding, 206-216.
20. Lange, D.B. 1990. A Formal Model of Hypertext, NIST Hypertext Standardization Workshop, February 1990, pp 145-166.
21. Le Moigne, J-L. 1990. La modélisation du système génèral. Théorie de la modélisation. Paris. Presses Universitaires de France. p 62.
22. Marshall C.C., Shipman III F.M., Coombs J. H. 1994. VIKI: Spatial Hypertext Supporting Emergent Structure. Xerox Palo Alto Research Center. ECHT 94 Proceeding.pp 83-93.
23. Meyrowitz, N. 1991. Hypertext-does it reduce cholesterol, too? From Memex to Hypertext (Kahn et al.,ed.) Academic Press, Boston.
24. Nanard, J., Nanard, M. 1991. Using Structured Typed to Incorporate Knowledge in Hypertext. Proceeding of the Hypertext'91, (San Antonio, TX, ACM Press), 329-343.
25. Nanard, J., Nanard, M. 1995. Hypertext Design Environments and the Hypertext Design Process. Communications of the ACM, 38(8), 49-56.
26. Nürnberg, P.J., Leggett, J.J., Schneider, E.R. 1997. As We Should Have Thought. Proceedings of Hypertext 97, (Southampton, UK, ACM Press) 96-101.
27. Parets, J.; Torres, J.C. 1996. Software Maintenance versus Software Evolution: an Approach to Software Systems Evolution. IEEE Symposium and Workshop on Engineering of Computer Based-Systems (ECB'96). Friedrichshafen, Germany, March 1996, pp 134-141.
28. Rao, U., Turoff, M. 1990. Hypertext Functionality: A Theoretical Framework, International Journal of Human-Computer Interaction, Volume 4, Number 2, pp 333-358.
29. Salampasis, M. Hypermedia. [On-line] http://osiris.sund.ac.uk/~cs0msa/hyp0.htm
30. Schnase, J. L., Leggett, J.J, Hicks, D.L., Szabo, R. L. 1993. Semantic Data Modelling of Hypermedia Associations. ACM Transactions on Information Systems, Vol. 11, Nº1, January 1993, 27-50.
31. Shipman III, F.M., McCall, R. J. 1997. Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale form Design Communication. Artificial Intelligence in Engineering Design, Analysis, and Manufacturing (AIEDAM), 11, 2 (April, 1997), pp. 141-154.
32. Streitz, N. A., Haake,, J. M., Hanneman, J., Lemke, A., Schuler, W. Schutt, H., Thuring, M., "SEPIA: A co-operative Hypermedia Authoring Environment," ECHT 92 Proceeding, (Milan, Italy, Nov.30–Dec. 4), 11-23.
33. Uschold, M. 1996. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, vol. 11, n. 2, June 1996.