A Cognitive Model for Adaptive Hypermedia Systems

Lina García-Cabrera
Depto. Informática. Universidad de Jaén
E.P.S. Avda. Madrid, 35, Jaén, SPAIN
Tel: +34 953 212475 E-mail: lina@ujaen.es

José Parets-Llorca
Depto. L.S.I. Universidad de Granada
E.T.S.I.I. Avda. Andalucía, 38, Granada, SPAIN
Tel: +34 958 243180 E-mail: jparets@ugr.es

Abstract[1]

As other authors, we believe that hypermedia systems and web, specially, can increase and improve their functionality by means of making the semantic of the structure of information systems explicit. In this paper, we shall attempt to justify the need for a cognitive model in the conception of hypermedia systems. 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. Introduction

Many years ago, Vannevar Bush established that *"the process of tying two items together is the important thing"* [1], i.e., models must provide an abstraction of functionality from structure instead of abstracting the hypermedia connectivity from information content [5]. We share this opinion and consider that hypermedia systems must be information systems, which offer support to the structuring and access processes, according to *conceptual associations* that can be established between their different information items. In our opinion, the *"primacy of structure over link"* [4] could convert information systems based on links between information chunks into real *knowledge systems* based on structured information items [3]. This structuring process requires the identification of its components and a set of rules that tell us how they can be organised from a conceptual point of view, i.e., the building process for hypermedia must necessarily be based on *cognitive models*.

2. Why a Cognitive Model?

Why should the semantic structure of hypermedia systems be made explicit? Because this explicit representation benefits the users during the development and use activities: construction, maintenance and navigation.

[1] This research is supported by a project by the Spanish CICYT (TIC97-0593-C05-04) which is a subproject of the MENHIR project (TIC97-0593-C05).

The preparation of hyperdocuments includes a lot of changes, additions and updates, frequently carried out by different authors. This implies the need of sharing a cohesive conceptual world between them. On the other hand, for readers, hyperdocuments may be sources of information that facilitate their knowledge comprehension. Finally, a good maintenance of information systems is only possible if the author represents the Design Rationale, i.e., the process of underlying reasoning and decision-making carried out. Furthermore, a hypermedia model must be *adaptive*, i.e., it should allow the representation of the design process and of the hypermedia activity that led us to 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* and *navigation* possibilities in a hypermedia system.

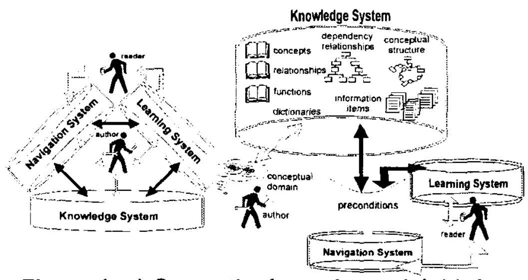

Figure 1. a) Semantic-dynamic model, b) Architecture of knowledge system.

3. A Semantic-Dynamic Model

A Hypermedia System can be conceived as being made up by three interrelated and interacting systems: The Knowledge System, the Navigation one and the Learning one (fig. 1a). 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 change in this structure throughout its evolution. Navigation System helps the reader in his interaction with the information system. Using the knowledge base and the reader activity over time in a dynamic way, this system determines –firstly- the accessible information and –secondly- its interaction possibilities. Finally, 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.

In order to clarify the possibilities for the approach, we use an example extracted from a concrete conceptual domain: *Definition and Control of Processes* in operating systems. Figure 3, shows a part of the conceptual graph.

3.1. Towards an Adaptive Hypermedia

In order to highlight the adaptive aspects of the model we will start clarifying the meaning of the four most important concepts of our model: *information items, conceptual structure, preconditions* and *actions*. We will see how conceptual structure and preconditions stress the cognitive and evolving aspects of a hypermedia system (fig.4).

3.1.1. Information Items. A hypermedia system is an information system made of different pieces of information –the *information items* (fig. 2a)– that can be referenced, used and composed. Each of the information items has a set of *properties* that tell us the kind of information we are working with and the function that it could represent in a context. The most important property is the set of associated concepts, i.e., an information item will be identified by one or more *concepts*. These concepts will part of a *Conceptual Domain*. Thus, the whole set of information items labelled with concepts which belong to the same Conceptual Domain implies *an Information Domain*.

Concepts are dependent on the Conceptual Domain, i.e., they have to be created by the author/s or expert/s in the topic the document is devoted to. These kinds of *dictionaries* of concepts are built along the document elaboration and maintenance processes. But many information items can refer to the same concept or set of concepts. In this case each information item will play a different role in a context. For instance, the *create-a-process* concept (one of the *Process-Operations*) can be referenced by different items, which plays the roles of *definition, explanation, algorithm* or *example.* This property is called *role, function* or *intention of an information item*. The roles of an information item in the context are applicable in any case. The model provides some generic roles played by a piece of information and supports the addition of new functions that the author considers appropriate in the domain.

Apart from concepts and functions, an information item has additional properties: type of *language of the content* (such as text, sound, graphic, image, animation, execution or hypermedia), those related to the *edition aspects* (author/s, edition date, organism, quality, visitors,...), and *specialisation level.* This property allows that more than one definition of the same concept can be provided attending to the specialisation level of the future reader.

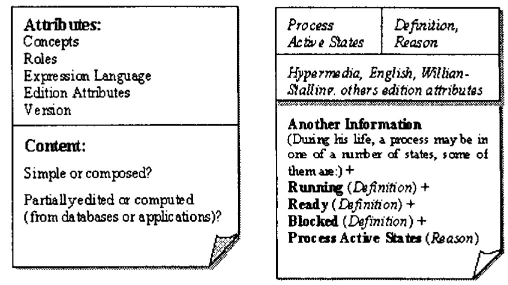

Figure 2. a) Information item, b) An example of composed information item.

An item may be made up by others if a generic dependency relationship (section 3.1.2.), which suggests that they are elements of the same set exists. If the information item is a composition, it will be labelled with a more generic concept and it will aggregate the roles of the component items. For instance, *Running, Ready* and *Blocked* have the relationship *aKindOf* with *Process-Active-States* (fig. 2b). Then, a composed item, which is labelled with the generic concept, can be constructed by grouping all four items. In addition, a set of information items labelled with concepts and specific role/s can be viewed under another concept. In our example, a new information item which groups together the information items about operations over processes, can be elaborated: **Process-Operations** (*Analysis*):= **Process-Creation** (*Algorithm*) + **Process-Creation** (*Explanation*) + **Process-Termination** (*Explanation*) + **Process-Suspend**, **Process-Resume** (*Definition*, *Explanation*).

Finally, an information item can be labelled using more than one concept. In this case, roles are applied to each one of them. In the example, a paragraph which explains the suspend and resume operations is labelled as: **Process-Suspend, Process-Resume** (*Explanation*).

As mentioned above, information items are created and edited by the author, but they can also be computed from others pieces of information. Additionally, another application or information database could provide information items. More than one semantic model in research literature just only build documents as composition of data represented and presented as style template, i.e., as a database form. But, a lot of documents cannot be adapted to this simple skeleton. The model presented here allows different kinds of information which forms a conceptual entity, in an independent way of the origin of the information items.

3.1.2. Conceptual Structure. 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*. *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 [6]. 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.

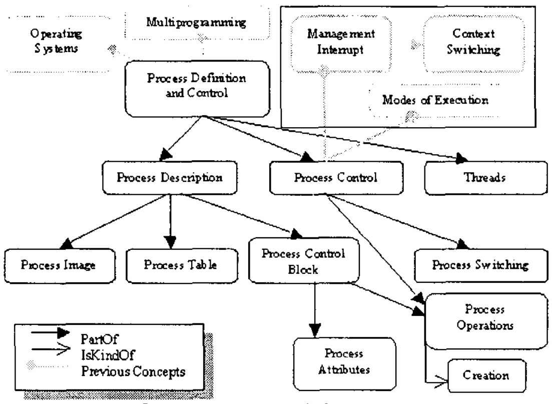

Figure 3. Concept-map of dependencies.

In addition to these domain dependent relationships, *dependencies between concepts* (fig. 3), which are domain independent, can be considered: aggregation (p*artOf*), instantiation (i*sA* or i*nstanceOf*), and specialisation (a*KindOf*). The dependency *partOf* allows hierarchies between concepts. The dependency *aKindOf* allows the composition of information items as shown in previous section (fig. 2b). The dependency *isA* allows the definition of a concept using other more generic concepts. Less formal dependency relationships can be established: *complementarity* and *priority* of concepts. The first one implies a relationship which adds or improves a concept. The second one establishes a temporal order between concepts and is important in the navigation order. For instance: (((**Management-Interrup***t*) Some) Produce) **Process-Switching** (in a postfix notation).

These dependencies and relationships can be expressed by means of concept maps. A Concept Map is a visual graph made up by nodes –concepts- and arcs –relationships-. 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. In the example (fig. 3), the environment of *Process-Control* is made up by concepts such as *Magnagement-Interrupt*, *Process-Switching* or *Process-Description*. The notion of environment allows some interesting operations which are known as *queries based on the structure* in the literature. For instance, which concepts complement another concept; which concepts are derived from another concept; which concepts produce or cause another concept; which concepts are one level higher or lower in the conceptual structure; which concepts are separated from another concept by a distance *d*; which documents are related to some conceptual domain.

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 (see below) in a concrete information domain. The concepts *Management-Interrupt* and *Modes-of-Execution* belong to another Conceptual Domain but have strong relationships with the concept *Process-Control* (fig. 3). In this case, the authoring tool could suggest the inclusion in the hyperdocument.

3.1.3. Preconditions. They guide development, maintenance and navigation of the hypermedia system. They are provided from the different Systems, and are always applied by the Navigation System. They 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 *item framework*. Two types of preconditions may be distinguished:

1. *Derived from the semantic structure* of the information system. Obviously, navigation will be restricted inside the *conceptual world as* designed by the author. The aspects which are useful in establishing preconditions are: The conceptual structure of a conceptual domain of which an item is member; generic dependencies between concepts; functions that an information item may play in the context of an information item, and the language of an item.
2. Topics *derived from the navigation* itself and which provide a better adjustment of the structuring process: a) the type of navigation: a group of preconditions that characterise certain types of navigation; b) the navigation carried out by the user over time or functional history -the set of operations performed or information items selected by the reader and their order; c) considerations about security and access control: user identification, restrictions in accessing the Conceptual Structure, item functions and item versions.

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 as: *"if this and that then..."*, *"if before of ... and after of... then show..."*, *"take into account if reader knows this and that concepts..."*, *"if the reader has made this tour.... then that items can be shown"*. As Stotts and Furuta [2] we consider that a hypertext is an *interactive document* which provides a *dynamic structure*. This assumption implies the need for temporal logic in expressing what link sequences can be followed during browsing. These authors propose temporal logic as formalism in checking the properties of a hypertext. In our approach, we use temporal logic as an inherent way of expressing preconditions. Consequently, this kind of rules determines, at all times, which pieces of in-

Definition 3.1 An **information item** is any piece of *identified* information, which represents a conceptual unit in the information system. Each information item has a set of properties describing the type and functionality of the contained information. **Definition 3.2** A **property** of an information item is an associated attribute which describes the type, function or behaviour of the information that the information item contains. **Definition 3.3** A **concept** is an idea, thought or abstraction which can be labelled by the author in order to make explicit his knowledge and understanding, i.e., a labelled idea. **Definition 3.4** A **conceptual domain** is the set of concepts which the different information items in a hypermedia system may refer to. **Definition 3.5** An **information domain** is the set of information items identified by concepts belonging to a Conceptual Domain. **Definition 3.6** A **role** is one of the possible roles that the information item may play in the context of an information system. From the author's point of view, an item may play a certain role in the context, but for the reader, it follows a link with the aim of reaching a certain type of information about a specific concept. **Definition 3.7** The **specialisation level** is a property of an information item that determines the level of specialisation of the information contained in the item. **Definition 3.8** A **Conceptual Structure** of a Conceptual Domain is a network or graph of labelled concepts which maintains information about: a) relationships between concepts, and b) generic dependencies between them. **Definition 3.9** A **relationship** between concepts is a labelled association between two or more concepts, members of a conceptual domain, that maintain some type of connection. **Definition 3.10** A **generic dependency** between concepts is a connection or subordination relationship that is independent of the considered Conceptual Domain. **Definition 3.11** The **concept environment** is the set of concepts that have some sort of relationship or dependency with a concrete concept. **Definition 3.12 Preconditions** are the set of conditions that constrain the information items that can be associated with another information item. **Definition 3.13** An **association** between information items is a connection relationship among two or more information items. **Definition 3.14** An **information item framework** is the set of preconditions that hold when an item is achieved. It limits or constrains the set of information items that can be further associated with this item. **Definition 3.15** An **action** is an operation that is activated when certain association between information items is followed. This association depends on the expression language used by the target information item.

Figure 4. Terminology of the semantic-dynamic model.

formation can be activated and which 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 navigates through the system.

3.1.4. Actions. During the navigation process, the reader reaches information items and the Navigation System will apply some *Action* to this item. The action carried out will depend on the language used to express the item. Following an association and carrying out the action will imply feedback information for the Navigation System and, sometimes, also for the Learning System. This implies that each new action will provide new conditions which will restrict the associations that can be applied to the next information item. So, the *information item framework* is dynamic and depends on previous navigation.

4. Knowledge System

The main objective of the Knowledge System is the storage, structuring and maintenance of the different pieces of information. It is made up by a Memory Subsystem and a Presentation Subsystem.

The *Memory Subsystem* allows the storage of selected knowledge for each information system. It memorises information concerning the whole *Conceptual Domain* (def. 3.4) which is managed in a particular information system and establishes the *raw material* used in constructing the Hypermedia Information System, its possible components and their behaviour. The elements to be managed are:

1. The *Conceptual Structure* (def. 3.8) which allows *information items* (def. 3.1) to be catalogued. It includes a dictionary of *Relationships,*
2. A dictionary of *Functions* (def. 3.6) which can be played by an item in a document.
3. The *Information Items*: these information items can be used to construct hyperdocuments, will be expressed in one or more possible language/s and will have to be catalogued under one or several concepts. They will also be labelled with one or several functions in a particular context and have certain edition properties.

The *Presentation Subsystem* determines the set of possible presentations of the hypermedia information system. To some extent, it establishes the possible hypermedia documents which can be built with the items in the Memory Subsystem. For example, the concept *Process-Operations* has a dependency relationship with the concept *Process-Control-Block* and with the concept *Process-Control*. This implies that the items labelled with *Process-Operations* can be shown with former items, latter items or both, i.e., three presentations are possible. The Presentation Subsystem use filters in order to determine what item associations are admissible. These filters come from: the generic dependencies between concepts (def. 3.10); the set of roles; the conceptual structure and, finally, the language of an item. This last property obviously determines what actions (def. 3.15) can be applied to the information item.

5. Navigation and the Learning Systems

The model is not complete if we forget basic functionality of a hypermedia document: navigation. Then, the *Navigation System* allows browsing and remembering the memorised knowledge adapting it to the characteristics and interaction of the reader. The Navigation System selects a subset of the possible presentations. In other words, only a selection of preconditions (def. 3.12) offered by the Presentation Subsystem will be applied. In addition the Navigation System (section 3.1.3.) will apply the preconditions, which depend on previous navigation.

Author and reader interact with the Navigation System. In fact, some edition properties will be updated by this system: number of visits, type of visitors, and people and organisms who recommend it. The Navigation System has to take into account the following information at all times: First, the information item where the document reader is located at any moment; second, *concept/s environment* (def. 3.11) of the information item, and third, item *information framework* (def. 3.14), i.e., the preconditions set that is true for an information item. Using this information, the Navigation System can determine and show the conceptual structure related to an item as a set of multiple associations that can be followed from it. This means that navigation is not only possible through document, but also through the visual representation of his structure.

The *Learning System*, which modifies navigation taking into account the type of information that the reader wants to achieve (the *goals* of the reader) and/or knowledge that he wants to acquire, or better, learn (*achievements*).

As can be seen each System redefines, at some extent, the knowledge base provided by the Memory Subsystem which is stable for the reader but dynamic for the author. 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 one another and produces, in a dynamic way, adaptations within them. The explicit representation of the semantic structure drives the development, maintenance and navigation processes of information systems.

6. Knowledge System Functionality

A fifth key element of the model is the set of *operations* that provide a conscious support of every one of the previous components. The most important operations are those that allow the development and maintenance of the Knowledge System. This group includes operations which adds, erases or modifies an information item, a role, a concept, a relationship between concepts, etc. In general, these operations produce a record of changes. Others operations are the *checks* based on preconditions which determine the updating repercussions into each System. In particular, the model easily integrates *Validity* and *Relevance checks.* Examples of the former are: Is this concept a member of the Conceptual Domain of the document?, or is there any simple or composed information item labelled in this way?. Examples of the latter are: Can this link be followed from here?, is it suggested that the tool offers all the information labels with this concept?, do knowledge, navigation or, in addition, learning system, demand any constrain type?, what is type of?, why?.

Nevertheless, the model allows a certain degree of inconsistency and freedom (i.e., information items partially labelled, concepts with no relationships in the conceptual graph...). In this case, the Hypermedia System could detect incoherence but with no possibilities of performing an automatic management. This automatic activity implies that the system has rules which determine the understanding of information, and of the operations that can be carried out on it. This means that the model offers a flexible mechanism -acts and helps in an automatic way when it is possible and, always, facilitates a manual maintenance-.

As shown in fig. 1b, the author represents his complex domain/s of knowledge using the Memory Subsystem. He creates his concepts, relationship dictionary and function dictionary. He builds the conceptual and dependency structures. This knowledge will be used in characterising the different information items. As a result, these items will be labelled with concepts, functions, language, edition properties and version. The Presentation Subsystem, using the Memory Subsystem knowledge, allows compositions between information items and a subset of the possible presentations, which are possible with the whole set of information items. This subsystem helps the author in Validity and Relevance checks. These checks will control and drive the development and maintenance of the hypermedia information system. Furthermore, the Presentation Subsystem provides preconditions to the Navigation System in order to help the reader in a dynamic way.

7. Conclusion

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. A semantic-dynamic model is presented that provides: 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; and d) a dynamic navigation where is possible multitarget, multiproposal navigation with structural contextualisation.

8. References

[1] Bush, V. As We May Think. *The Atlantic Monthly*, 176: 101-108, July 1945.

[2] Stotts, P., Furuta, R., Ruiz-Cabarrus, C. Hyperdocuments as Automata: Verification of Trace-Based Browsing Properties by Model Checking, *ACM TOIS*, 16(1): 1-30, January 1998.

[3] Nanard, J., Nanard, M. Using Structured Types to Incorporate Knowledge in Hypertext. *Hypertext'91 Proc.*, ACM Press: 329-343.

[4] Nürnberg, P.J., Leggett, J.J., Schneider, E.R. As We Should Have Thought, *Hypertext'97 Proc.*, ACM Press: 96-101.

[5] Schnase, J.L., Leggett, J.J, Hicks, D.L., Szabo, R.L. Semantic Data Modeling of Hypermedia Associations, *ACM Trans. Information Systems*, 11(1):27-50, January 1993.

[6] Uschold, M. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2), June 1996.