Formal Foundations for the Evolution of Hypermedia Systems

Lina García-Cabrera*, M.José Rodríguez-Fórtiz**, José Parets-Llorca**

* Depto. Informática. Universidad de Jaén
EPS Avda. Madrid, 35, Jaén, SPAIN
Tel: +34 953 212475 E-mail: lina@ujaen.es

** Depto. L.S.I. Universidad de Granada
ETSII Avda. Andalucía, 38, Granada, SPAIN
Tel: +34 958 243179 E-mail:<mjfortiz, jparets@ugr.es>

Abstract[1]

In this paper, we shall attempt to justify the need for an evolving conception of hypermedia systems and its formalisation. We propose graph theory, predicate logic, temporal logic and Petri nets to support evolution in hypermedia systems. A semantic-dynamic model based on these formalisms is presented. It provides a complete, adaptive and evolving control of development and maintenance of hyperdocuments and an understandable navigation.

1. Introduction

Hypermedia systems are an special kind of Information Systems constructed over a conceptual domain. Because they include the knowledge captured by their authors, they are continuously changing. Changes can be carried out in the concepts offered by them, in the relationships between concepts, in the way of presenting the information and in the documents (information items) which explain the concepts.

Bieber [1] says, "Currently, developers and authors must build all hypermedia representations and navigation using single-step links without semantic or behaviour typing." and "Fourth-generation hypermedia features would provide sophisticated relationship management and navigation support." In our opinion, we must face two challenges. Firstly, we must assume the dynamic and evolving nature of hypermedia systems. A hypermedia system represents some aspects and relationships of a conceptual domain explained by a set of authors. But there are very different ways of representing, structuring and browsing it. Secondly, the bulk of the hypermedia systems, and web in particular, only considers the final hypermedia documents and, sometimes, the navigation performed by the reader. Nevertheless, the design, construction and evolution processes –the whole life-cycle- of hypermedia is not sufficiently considered [10]. However, this development process is very important because it implies a structuring process that is implicit, diluted and unaffordable inside the documents [5].

1.1. Our Approach

In order to provide dynamic, flexible, robust and understandable hypermedia systems we propose an approach based on four main assumptions:

- Following the Theory of the General System [7], a hypermedia system can be conceived as a set of interacting systems in continuous evolution.
- The following elements should be provided: mechanisms for representing the information system; a representation of the conceptual domain or ontology [12] that information belongs to; useful ways of browsing and remembering the memorised knowledge.
- The process of construction of information systems, conceptual domains and routes –ways of navigation- should be flexible.
- Information systems, conceptual domains and navigation routes are exposed to continuous changes and updates which should be integrated in the development process.

In order to provide an operational view of these assumptions, our approach distinguish two abstraction levels in the design of a hypermedia system. The first level, called *memorisation system*, includes the representation and management of information semantics [4], i.e. the conceptual domain. The second level, called *navigation system*, extends this semantics adding dependence and order relationships which allows navigation over the conceptual domain. This distinction is useful because allows a separation of concerns both in the development and the evolution processes. In addition

[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).

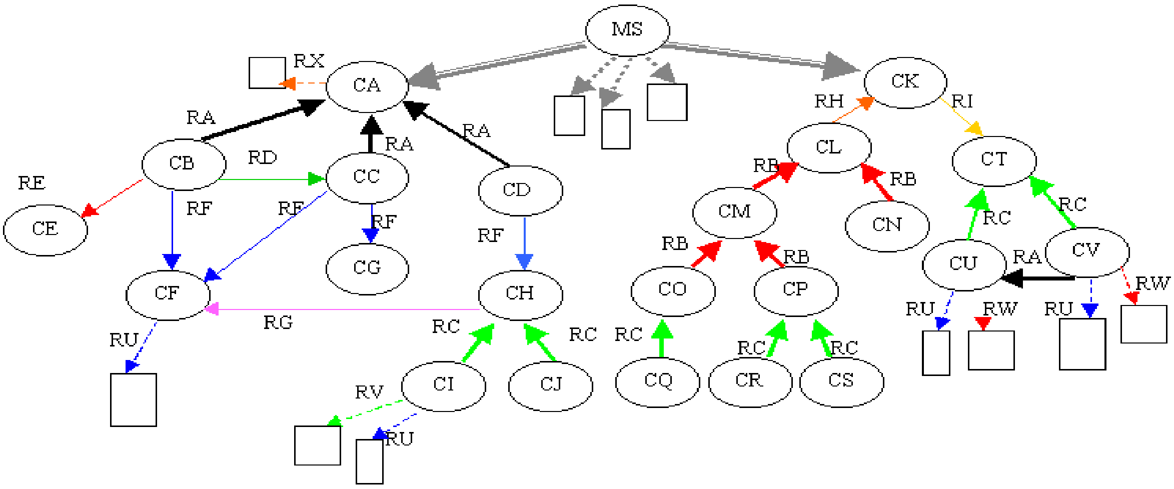


Figure 1. Examples of CSs of the Memorisation System.

different navigation 'styles' can be performed using the same semantic structure.

Different formalisms will be used in representing these systems which will allow to manage the development and evolution in both of them. They will be presented in the next sections.

2. An Evolutionary Model Based on Systems

A Hypermedia System can be conceived as being made up by two interrelated and interacting systems (for a complete description of the model, see [2]):

1. Memorisation System (*MS)* is in charge of the storage, structuring and maintenance of the different pieces of information –pages or documents-. 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.
2. Navigation System (*NS)* helps the reader in his interaction with the information system. Using the memorised knowledge and the reader activity over time in a dynamic way, this system determines –firstly- the accessible information and –secondly- the interaction possibilities.

A concrete and complete example of the use of the formalism to specify the structure and evolution of an hypermedia System can be seen in appendix..

2.1. Formalisation of the Hypermedia Systems

As stated above, two systems are distinguished in the model. The formalisms associated and the modelled aspects of each system are summarised in table 1.

In the *MS*, which mainly includes the semantic structure of an information system, graph theory [13] and temporal logic are used. The second system, *NS*, specifies the order relationships between concepts when navigation will be performed. Petri nets and temporal logic are used in this case [8][11].

The *MS* provides the necessary instruments which allows a representation of the information system by means of a directed graph [4], in which, nodes and links are labelled with semantic meanings –a semantic net-. The graph represents the conceptual domain –concepts and relationships between concepts- of the information system, named Conceptual Structure (CS*)*. The different information items –documents- can be associated –labelled- with one or more concepts of the CS. These items are also nodes of the CS. In order to allow provisional and incomplete development, items which are no related to any concept can also be included. Figure 1 shows an abstract example where *MS* is an artificial node which is the root of the represented information systems. Two conceptual structures are included (CA and CK). A conceptual structure for the Solar System is explained in the appendix example.

Therefore CS is defined as: CS = (C, II, A_c, A_i), where C is the set of concepts, II is the set of information items, A_c is the set of labelled conceptual associations, A_i is the set of labelled associations between concepts and information items.

Because CS is constructed by the authors in a dynamic way, some evolution operations as add-concept, delete-association, modify-association, add-item, etc. have to be included. The operations must verify a set of restrictions in order to maintain the consistency of the CS. These restrictions can be basic ones, defined as a functional part of the *MS*, or can also be defined by the author. Some examples of basic restrictions are:

- Each association of the *CS* must connect two concepts or a concept and an item.
- Each arc and node of the *CS* must be labelled.
- Two nodes in a *CS* cannot have the same label.

The author can also include additional restrictions which determine what associations between concepts are possible. In order to represent these restrictions, formulas in temporal logic are used. This formalism also allows to check if the CS is valid at any moment. Some examples are:

- Concept-A can be connected with concept-B by means of the relationship-A.
- The relationship-B must be acyclic.
- Concept-C can be connected with concept-G if concept-C is reached from concept-B.

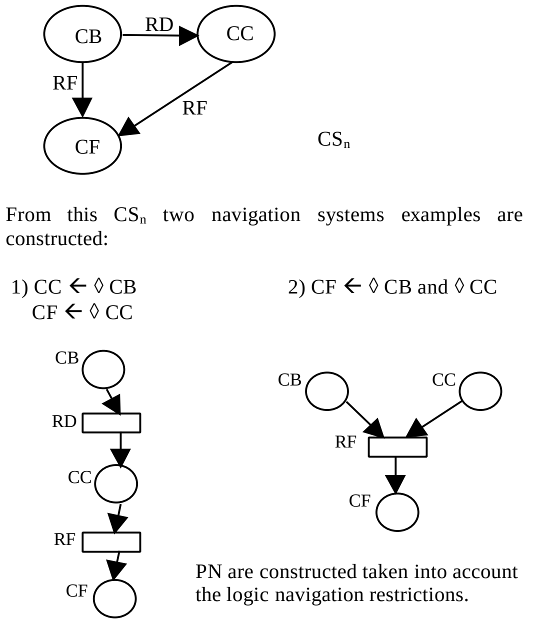
Therefore, the Memorisation System is defined as *MS* = (CS, RT, AC_e), where CS is the previously defined directed and labelled graph weakly connected that represents the conceptual domain of a hypermedia system, RT is the set of restrictions that must verify the CS –defined by the system RT_s and by the author RT_a- and AC_e is a set of evolutionary actions (see next section).

Memorisation System	**Graphs**	**Temporal Logic**
Concept (C)	Labelled node	Proposition
Item (II)	Labelled leaf node	Proposition
Relationship between concepts (A_c) or concepts and items (A_i)	Labelled arc	Formula with temporal and logic operators
Navigation System	**Petri Nets**	**Temporal Logic**
Concept or item	Place	Proposition
Order relationship between concepts or items	Transition and arcs	Formula with temporal operator
Dependence relationship between concepts or concepts and items	Transition and arcs	Formula with logic operator
Navigation	Firing transitions	Instantiation of formulas

Table 1. Formalisms used in specifying the structure of a hypermedia system

The Navigation System, using as basis the CS of the Memorisation System, allows a selection of a subset of the concepts and associations included in CS. This graph, CS_n, being a subgraph of CS, $CS_{n\ =}$ (C_n, $II_{n,}$ A_{cn}, A_{in}), will be presented to the reader. In addition, some navigation restriction can be added in order to follow more restricted paths in the subgraph. These restrictions or navigation rules are expressed using temporal logic. Considering the CS_n and temporal restrictions, a Petri net is automatically constructed. As demonstrated in [3] and in [8], Petri nets give an operational semantics to temporal logic formulas allowing an operational navigation. The algorithm which transforms temporal logic formulas in Petri net is explained in [3].

Therefore, the Navigation System is defined as *NS* = (CS_n , RT_n, PN, AC_e), where RT_n is the set of restrictions specified by the author by means of temporal logic, PN is the Petri Net and AC_e is the set of evolving actions to adding, deleting or modifying navigation restrictions (see next section).



From this CS_n two navigation systems examples are constructed:

PN are constructed taken into account the logic navigation restrictions.

Figure 2. Construction of the Navigation Paths

An example of the specification of the navigation possibilities is shown in figure 2. It gets a subgraph based on the left CS of the example of figure 1. The appendix presents the navigation system of part of the CS of the Solar System, having only into account the Earth relationships

2.2. Formalisation of the Hypermedia Evolution

Both systems, *MS* and *NS*, include a set of evolving actions, $AC_{e,}$ that allow to make and propagate changes in the hypermedia system. An evolving action can belong to three different types:

1. Actions that redefine some aspects the system. Obviously the basic restrictions discussed below, $RT_{s,}$ cannot be changed.
2. Actions that control the propagation of these changes inside of the system itself.
3. Actions that control the propagation of these changes outside the system, i.e. in the other system

When these actions are carried out they change the corresponding elements of the hypermedia system. Because integrity should be guaranteed in any case, these operations should be carried out following a set of meta-restrictions. The specification of these meta-restrictions implies a meta-level in the definition of the *MS* and *NS*.

Formalisms of a higher abstraction level should be used. See figure 3.

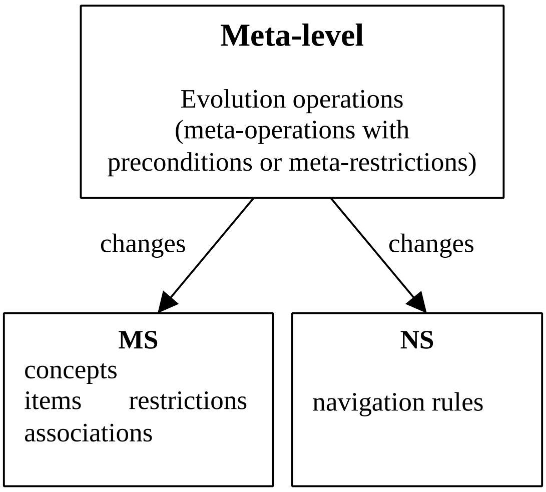


Figure 3. The Meta-level in evolving the Memorisation and Navigation Systems

Table 2 summarise the formalisms used in specifying meta-restrictions in both systems. Lets describe how they are specified for each systemm *MS* and *NS*.

The Memorisation System always must guarantee its consistency. Two aspects of this system can change, the CS –the graph- and the restrictions defined by the author. Graph Theory is used to represent the evolution operations of the graph and their associated meta-restrictions. Changes in restrictions defined by the author, RT_a, must be defined by means of meta-restrictions.

When the author changes the CS –add, delete or modify a concept, item or association- the system must check:

1. CS verifies the restrictions defined by the system and associations satisfy the set of restrictions defined by the author. RT acts as a set of restrictions for the operations, only if the operation match restrictions, it will be carried out (internal propagation of changes).
2. The subgraph used by the *NS*, CS_n, is consistent with changes in CS. If a concept or relationship have been deleted in CS, the *NS* must also delete this concept or relationship in CS_n (external propagation of changes).

When the author redefines –add, delete or modify- one associative restriction RT_a, the system must check:

1. The set of axioms about associations is valid, by means of predicate temporal logic.
2. CS verifies the new set of restrictions, using the graph theory. The system must detect the associations that not satisfy one or more restrictions and delete them (internal propagation of changes).
3. The CS_n verifies the new set of restrictions by means of graph theory. The system must detect the associations that not satisfy these restrictions and delete them (external propagation of changes).

Memorisation System	Graph Theory	Predicate Temporal Logic
Operation	Set operation	Predicate
Meta-restriction	Reachability function	Temporal formula
Modified aspect	Set	Variable

Navigation System	Predicate Temporal Logic
Operation	Predicate
Meta-restriction	Formula
Modified aspect	Instantiation in the variable of a predicate

Table 2. Formalisms used in specifying the evolution meta-restrictions of a hypermedia system

Navigation System models evolution using predicate temporal logic. It provides a meta-level with evolution operations which manage and change the navigation restrictions. Navigation rules can be added, deleted or modified, and the meta-restrictions of these operations can be established.

In a similar way that the Memorisation System does, the consistency must be guaranteed during the evolution of the Navigation System. In this system, changes can be produced in the subgraph selected CS_n and in the navigation restrictions, RT_n, defined by the author, and therefore, in the PN obtained from them.

When CS_n is changed –the author select another set of concepts and relationships- new navigation possibilities are defined. In this case, the author must define again the navigation restrictions. This change is not a real evolution, the author is designing new navigation possibilities, but if these possibilities are defined in an incremental way, the system can aid the author in the design process.

When the author redefines –add, delete or modify- a navigation restriction, RT_n, the system must check:

1. The set of restrictions that establish the order of navigation is consistent. Predicate temporal logic is used to specify the evolution operations over the restrictions, and their associated meta-restrictions.
2. The navigation restrictions have changed. Changes in a restriction can imply the modification of other restrictions. The PN based in the navigation restrictions must evolve, generating it again (internal propagation of changes).

Figure 4 sums up the evolving changes described above and the interactions between the systems Restrictions defined by the system, RT_s, or by the author, RT_a, are associated to the conceptual structure CS (1). Author selects only a subset of concepts and relationships from the CS in order to establish the navigation routes, creating the CS_n (2). Navigation restrictions, RT_n , are added (3) and a Petri net, PN, is created from them (4).

Evolution can be carried out in the conceptual structure, CS (5), in RT_a by means of predicate logic (6) and in RT_n using predicate temporal logic (8). When RT_a is modified CS could also change (7). PN evolves being reconstructed from RT_n (4). The evolution in the Memorisation system is also propagated to the Navigation system (2).

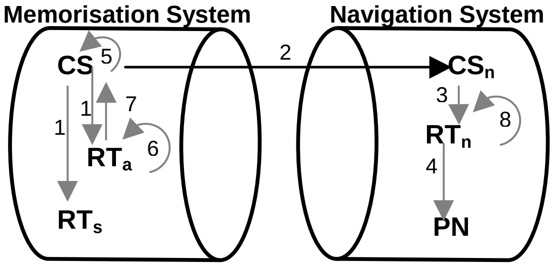


Figure 4. Definition and evolution of a hypermedia system

3. Contributions of the Formalisms

The different formalisms –graphs, Petri nets and propositional and predicate temporal logic- allow to model and distinguish between the information system, the conceptual structure and navigation. The author organises the information of the Memorisation System according to his particular interpretation of the conceptual domain. Therefore, to offer more than one structure –perspective- of the same information is possible. In addition, the model can provide more than one view (CS_n s) of the source CS by means of the Navigation System and different routes of navigation over the same subset of information

In particular, the Memorisation System contains the semantic structure–how knowledge is organised-, therefore, labelled graphs are the more suitable mechanism for representing it. Because restrictions should be also represented, indicating what associations are valid in the CS, temporal logic is a natural way to formulate them.

In the Navigation System, the main objective is to restrict the possible paths that can be followed when information is navigated and the order in which navigation is carried out. Temporal logic allows the specification of order relationships and Petri nets offer an operational formalism which can be executed in order to show these paths and analyse their properties [8][10].

The formalisms used in evolving the systems –graph theory and predicate temporal logic- easily support the changes and its propagation. Changes in the items, the CS, and in the Petri net are possible in an independent way. But, at the same time, the system can propagate these changes in order to maintain the global consistency.

In particular, graph theory is based on set theory, so the evolution operations can be expressed by simple set operations. Predicate temporal logic allows us to modify consistently the restrictions expressed in propositional temporal logic. Predicate temporal logic manage the meta-restrictions treating the propositions of the restrictions as variables, modifying them, and therefore, changing the restrictions.

Predicate temporal logic is used in the Navigation System with the same proposal, but respect to navigation restrictions. Predicate temporal logic is used, as demonstrated in previous papers [8][9], to verify these restrictions and to observe how the evolution is carried out.

The proposal of one such amount of formalisms has a main objective: to represent each evolution problem using the formalism which better fits the evolution possibilities. Obviously these formalism are hidden and the author have not to know them. These formalisms can be hidden inside the tools which implements the *MS* and the *NS* and the author could define its CS and restrictions using a visual -graph- representation of them.

4. The Evolution Formalisms in Other Systems

Although we use the previous formalisms in specifying and evolving hypermedia systems, we consider that they are useful in modelling the functioning and evolution of other types of systems, as reactive systems or temporal databases [8][9].

Graph theory can represent the relationships between agents and their environment in reactive systems. The associations established in the schema of temporal databases can also be defined by means of graph theory.

Due to the nature of both kinds of systems, meta-restrictions about relationships can be expressed using Temporal Logic. The evolution of these relationships and restrictions can be expressed by predicate temporal logic as a meta-level which defines the evolution operations and their meta-restrictions.

5. Conclusions

The separation of hypermedia systems in two abstraction levels allows a specification and management of the semantics of information and its navigation in a separated way, using different formalisms. Evolution operations can be defined independently in each level, but it is possible to determine what changes must be propagated to other components or to the other level.

The most important consideration during evolution is the conservation of the integrity of the system. Each evolution operation must verify a meta-restriction, checking the integrity restrictions associated to it. The meta-restrictions depend on the system (*MS* or *NS*) in which the change will be carried out.

The novelty of our approach about evolution is the incorporation of a meta-level, by means of reflectivity and second order, which allows us to reason about the functioning and structure of an hypermedia system which evolves.

The selected formalisms allow an easy specification and change of the structure of each system. It is very easy to modify a graph, a Petri net or a logic program in order

to change the structure of the system that they represent. Set Theory allows the verification of properties and integrity rules over the graph. Predicate temporal logic represents the evolution meta-restrictions over the memorisation and navigation in a hypermedia system.
These evolution formalisms can also be applied to other kinds of systems with an evolving nature, such as reactive or temporal ones.

6. Appendix

The following example of a hypermedia system shows different concepts related to the Solar System.

First of all the specification and evolution of the Memorisation System will be presented. After that, the Navigation system will be specified and evolved.

6.1. Specification of the Memorisation System.

a) Graph *CS=(C, II, A_c, A_i)* (See figure 5)

C= {Solar System, Planets, Stars, Earth, Venus, Sun, Moon, Countries, Oceans, Portugal}
II={P1, M1, C1, C2, Po1, Po2, O1, Su1, Su2, S1, S2, S3}
A_c= {<Earth, rotate, Moon>, <Earth, part_of, Countries>, <Earth, part_of, Oceans>, <Sun, rotate, Earth>, <Sun, rotate, Venus>, <Countries, is_a, Portugal>, <Solar_System, part_of, Planets>, <Solar_System, part_of, Sun>, <Stars, is_a, Sun>,<Stars, part_of, Solar_System>, <Planets, is_a, Earth>, <Planets, is_a, Venus>}

A_i= {<Moon, photos, M1>, <Countries, list, C1>, <Countries, cities, C2>, <Portugal, map, Po2>, <Portugal, history, Po1>, Oceans, list, O1>, <Sun, photos, Su1>, <Sun, quimical composition, Su2>, <Planets, def, P1>, <Stars, def, S1>, <Stars, nova, S2>, <Stars, supernova, S3>

b) Temporal logic

Examples of restrictions RT over the associations:
- *RT_s : is_a* association is not recursive.
  <c, is_a, c1> ← not à <c1, is_a, c> " c, c1 Î C
- RT_a: If an *is_a* association exist previously between any concept and the *Planets* concept, an association *rotate* must be added relating that concept with the *Sun* concept (every planet must rotate around the sun).
  <c, rotate, Sun> ← ◊ <c, is_a, Planets> ∀ c∈ C

6.2. Evolution of the Memorisation System.

a) Graph Theory

Example of operation: *add_concept: Saturn.*
The meta-restriction of this evolution operation must hold.
- *Meta-restriction*: *Saturn Ï C*
  Meta-restriction holds, so *Saturn* can be a new concept.
- *Internal propagation of the change:* if a concept is added, it must be associated to other concepts. The evolution operation *add_concep_assoc* must be carried out as consequence of the previous.

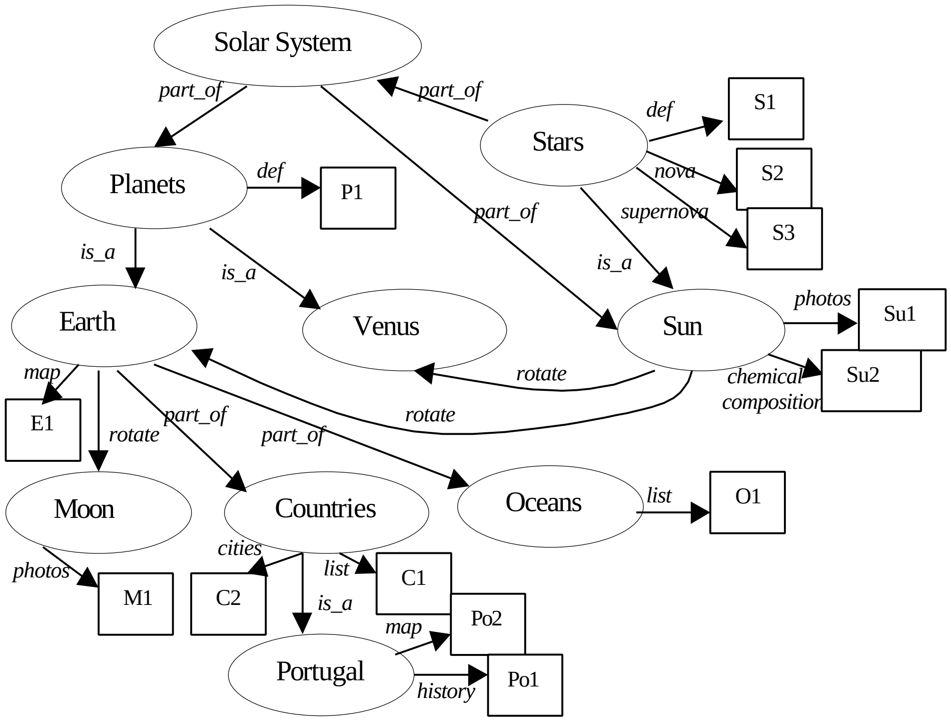


Figure 5. CS from a Solar System hypermedia

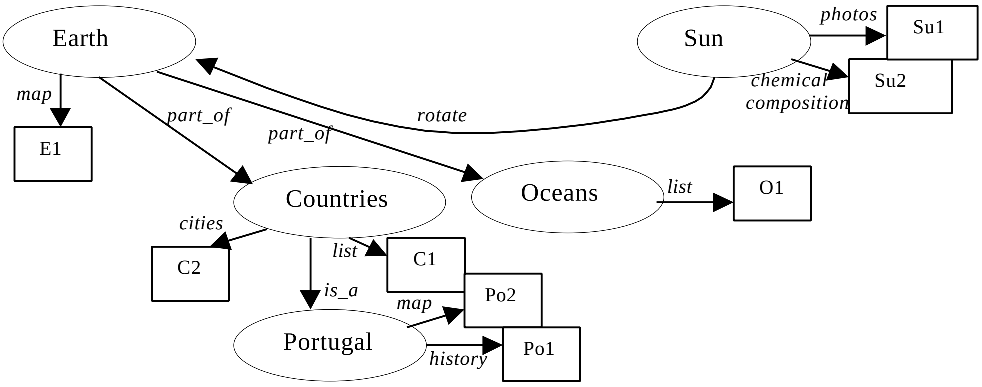


Figure 6. CS$_{n:}$ selection of CS

In this case the operation: a*dd_concep_assoc: <Planets, is_a, Saturn>* will be carried out. Its meta-restriction must also be verified:

- *Meta-restriction*: *<Saturn, is_a, Planets> Ï* A_c

This meta-restriction holds. It can be also verified proving the logic restriction:

<c, is_a, c1> ← *not à <c1, is_a, c>* with *c = Planets* and *c1= Saturn.*

After these changes, the graph which represents the Memorisation System has evolved: CS CS'

CS' = (C', II', A_c*',* A_i*') ; C'= C È {Saturn}; II' = II ;*
A_c*'=* A_c *È {<Planets, is_a, Saturn};* A_i*' =* A_i

b) Predicate Temporal Logic

Restrictions RT over the associations can be also changed. Predicate Temporal Logic is used as a meta-level to manage and evolve these restrictions.

Example:

As previously stated, cycles in concept associations are not allowed. An association *ac2* can be included in the restriction to establish an association *ac1* if previously *ac1* is not included in the restriction to establish the association *ac2*.

The meta-restriction which describes this restriction is:

addRest(ac2, ac1)← *not à isRest(ac1,ac2)*
ac1,ac2Î A_c

This clause can be instantiated:

addRest (<c, rotate, Sun>, <c, is_a, Planets>) ←
not à isRest(<c, is_a, Planets>, <c,rotate, Sun>)

If *c* is *Earth*, the restriction can not be added because the meta-restriction does not hold (see 6.1.b)). *Earth* is a planet, and this is the restriction to rotate around sun. If we stated that the restriction of being a planet is that previously it rotate around sun (inverse relationship), a not desired cycle situation is being produced.

6.3. Specification of the Navigation System

A part of the Memorisation System, CS_n , is chosen to navigate (See figure 6). In that case the navigation restrictions are expressed in Temporal Logic.

a) Temporal Logic

Example of definition of navigation restriction:

c.Portugal.map ← *à c.Countries.list and a.is_a*

It expresses that the *map* of *Portugal* can be shown if previously the *list* of *Countries* has been presented and there is an association *is_a* between both concepts. Letters *c* and *a* in the propositions represent the concepts and associations respectively.

Using the previous CS, the rest of the navigation restrictions can be constructed automatically. For example:

c.Portugal.map ← *à c.Countries.cities and a.is_a*
c.Portugal.history ← *à c.Countries.cities and a.is_a*

b) Petri nets

A Petri net can be constructed from the navigation restrictions, as the figure 7 shows.

6.4. Evolution of the Navigation System

Predicate Temporal Logic is used to define the meta-restrictions associated to the evolution operations of this System.

a) Predicate Temporal Logic

Adding, modifying or deleting navigation restrictions is possible if each concept and item selected from the CS can be reached. A navigation restriction can be modified or deleted if the concepts and items that they reference are referenced in other restrictions because, in other case, these concepts and items will be unreachable.

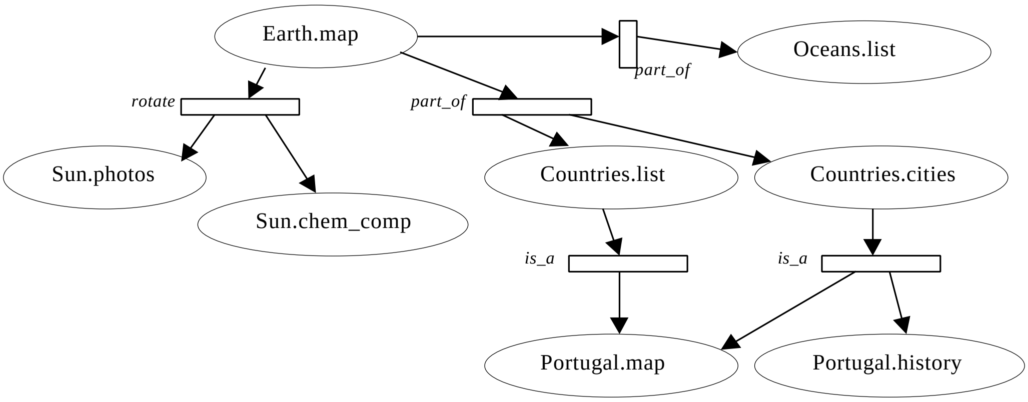


Figure 7. Petri Net of the Navigation System

Example:

The meta-restriction of the evolution operation *delRest* (deleting a restriction) is: to get the concept and the item of the head of the restriction rule by means of another navigation restriction is possible, or another navigation restriction which includes a reference to that concept and item in its body exists.

delRest(c.i, nav_rest) ← à existRest(c.i, nav_rest1) or à (existRest(c1.i1, nav_rest2) and ref(nav_rest2, c.i)
" c Î C , " i Î II,

nav_rest is the restriction for navigating to the item *i* of the concept *c: c.i ← nav_prec*

If *c.i* is instantiated with *Portugal.map***,** the meta-restriction holds, then the navigation restriction can be deleted. Navigation restriction:

c.Portugal.map ← à c.Countries.list and a.part_of

can be deleted because there are another restriction which allows to reach that item:

c.Portugal.map ← à c.Countries.cities and a.is_a

As navigation restrictions have changed, Petri net must be modified to deleting the transition *is_a,*, and their arcs, which link the places *Countries.list* and *Portugal.map*.

7. References

[1] Bieber, M.; Vitali, F. 1997. Toward Support for Hypermedia on the World Wide Web. IEEE Computer, January 1997.

[2] García-Cabrera, L.; Parets-Llorca, J. (2000) A Cognitive Model for Adaptive Hypermedia Systems. The 1st International Conference on WISE, Workshop on World Wide Web Semantincs. Hong-Kong, China, June 2000, pp 29-33.

[3] Lin, C.; Chaudhury, A.; Whinston, A. B.; Marinescu, D. C. "Logical Inference of Horn Clauses in Petri Net Models". IEEE Transactions on Knowledge and Data Engineering, vol 5,3. pp: 416-425. 1993.

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

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

[6] Paderewski, P; Parets-Llorca, J.; Anaya, A; Rodriguez, M.J; Sanchez, G; Torres, J; Hurtado, M.V. (1999) A software development tool for evolutionary prototyping of information systems". IEEE CSCC'99. Computers and Computational Engineering in Control. Electric and Computer. Engineering Series. World Scientific and Engineering Society Press, pp: 347-352.

[7] Parets, J; Anaya, A; Rodríguez, M.J.; Paderewski, P. (1994) A Representation of Software Systems Evolution Based on the Theory of the General System. Computer Aided Systems Theory, EUROCAST'93. Lecture Notes in Computer Science vol.763. Springer-Verlag. pp: 96-109.

[8] Rodriguez-Fortiz, M.J, Parets Llorca, J. (2000) Using Predicate Temporal Logic and Coloured Petri Nets to specifying integrity restrictions in the structural evolution of temporal active systems. ISPSE 2000. International Symposium on Principles of Software Evolution. Kanazawa, Japan, November 2000, pp 81-85.

[9] Rodriguez-Fortiz, M.J. Software Evolution: A Formalisation Based in Predicate Temporal Logic and Coloured Petri Nets. (in Spanish). Ph Thesis. October 2000.

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

[11] 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.

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

[13] Wang, W., Rada, R. Structured Hypertext with Domain Semantics, ACM Trans. Information Systems, 16(4), October 1998.