Computer Science Journal of Moldova, vol.2, no.2(5), 1994

The intellectual hypertext for knowledge transfer

G.Andrienko N.Andrienko

Abstract

Keywords: hypertext, intellectual indexing, human–computer interaction, knowledge based systems, knowledge elicitation methods

1 Problem

We see the difference of the hypertext as an approach to the information management from other approaches (e.g.DBMS) in the following: the user usually accesses it to study a certain subject rather than to find the specific information. The study is carried on by viewing a number of information fragments associated in accordance with their meaning, the sequence of viewing being governed by the user's objectives. The possibility to vary the sequence of viewing the contents of the hypertext, as opposed to the linear text, results from the information being broken into fragments (stories) contained in hypertext nodes and the links being established between them which, as a rule, allow the user to pass from the currently viewed node to one of the several nodes linked to it. Evidently, the more links there are in hypertext, the better is the adaptability of the hypertext to the various user's objectives.

However, the growing number of links makes the work of the user with the hypertext more complicated. While viewing each of the hypertext nodes the user has to make the decision to which of the related stories he should pass to proceed with the study of the subject of interest. An improper decision can lead to the lack of understanding of the subject or to the misunderstanding of it. In fact, to choose properly

©1994 by G.Andrienko, N.Andrienko

the way of studying the hypertext material it is necessary to know this material beforehand! This means that the task of finding the proper root through the hypertext cannot be left to the user who is supposed just to start acquainting himself with the material. There is the reason to pass such functions to the intellectual hypertext management system (further referred to as the HTMS).

The HTMS should be based on the knowledge of the material contained in the hypertext (domain knowledge) and of the ways of representing this material (knowledge of the hypertext structure). The tasks of the intellectual HTMS:

- to identify the user's problem or to define the area of his information interests;
- to pick a set of stories from the hypertext the contents of which correspond to the user's needs or can be useful for the solution of the problem;
- to classify the nodes picked into basic and auxiliary (the latter are the nodes contained comments, examples, discussions, detailed consideration of the subject represented in the former ones). The nodes should be given a proper appearance to attract the user's attention to the significant concepts, features, facts and so on;
- to supply the nodes with the necessary links in accordance with the logically motivated order of studying the material. The links with the auxiliary nodes should not lead away from the basic line of studying, therefore the only way from the auxiliary node must be the return to the basic node left just before.

2 Approach

The task of building the intellectual HTMS is contrary in certain sense to the conventional idea of a hypertext as a collection of arbitrary textual, graphical and other types of items connected with arbitrary links.

First, the intellectual HTMS necessitates the knowledge base on the domain structure. The knowledge base must contain the basic concepts of the given domain and the semantic links between them, thus the convenient way of representing the knowledge is the semantic network.

Second, the intellectual management of the hypertext requires formalization of the hypertext itself. Therefore, the creation of the hypertext is preceded by the phase of projecting its structure. Based on the contents of the material to be presented in the hypertext a decision is made about the appropriate types of nodes. For each node type a description (frame) is specified which points the necessary items in the contents of the node and the mandatory links with the other nodes.

While building the hypertext the compilers make the instance nodes of the previously specified types and fill them with the concrete information to fit the corresponding frames. Using the frames the system has the ability to control the process of filling the nodes. Listed for each hypertext node must be the related domain concepts represented in the knowledge base as well as the relations that associate the node with these concepts.

The set of the relations used is determined by the nature of the problems which solution the hypertext is intended to support. It is convenient to formulate the relations in terms of certain actions. In some cases appropriate are the relations "describe", "confirm", "contradict", "exhibit", other areas would require "cause", "result", "lead to", "tend". When a hypertext is created for some practical purpose the relations can be "find", "create", "change" and so on. Let us name these relations the P–relations (problem relations) stressing their problem–specific nature.

In a hypertext built in such a way the following activities of the HTMS become possible:

(1) The system identifies the user's problem. From the system's standpoint this means the search in the domain knowledge base for the relevant concepts and the selection of the P–relations corresponding to the user's problem. Different types of user interface suppose distinct specific realizations of such activity.

(2) Based on links between concepts in the knowledge base the system can bring into consideration some other concepts and P–relations which were not mentioned in the user's request. This activity is possible when some metaknowledge about links between concepts and P–relations is available.

(3) The system finds the hypertext nodes associated with the selected domain concepts by the selected P–relations. The nodes associated with the concepts or by the P–relations taken on the system's initiative are considered to be auxiliary.

(4) The system binds together the nodes thus picked introducing when it is necessary new links between them based on the P–relations and the links between the concepts. In addition, new hypertext nodes can be created. In particular, it can be the grouping nodes to bring together the pieces of information from the hypertext nodes associated with the same concept or to give the summary of the nodes with the uniform structure.

(5) The system provides the proper appearance of the picked hypertext material to mark the most significant information from the standpoint of the user's problem.

3 Example

As an illustration we refer to the hypertext on programming in the Windows environment. Corresponding to such hypertext is the knowledge base containing the basic concepts of Windows: windows and their constituents, messages, controls, resources, dialogs. Listed are the major properties of these concepts (the window class, the style of the brush, the width and the height of the window's client area and so on) and indicated are the links between the concepts (a window receives and sends messages, a control belongs to some window etc).

The range of problems solved with the aid of the hypertext requires to consider such P–relations as “find”, “determine”, “get”, “create”, “delete”, “change”, “use”. The hypertext should contain information about the data structures, messages and functions of the Windows. Correspondingly, the node types are introduced into the hypertext to describe the data structures, messages and functions.

The frame of a function node comprises the name of the function, its arguments, the returned value, the purpose, the peculiarities of its usage (relations with other functions, used resources, necessary settings), the data structures used, Windows messages processed or caused by the function and the example of the usage of the function. Evident are the necessary types of links to bind the node of a function with the nodes of other functions, as well as with the nodes describing messages and data structures. The P–relations are established with the concepts denoting elements of Windows which are used by the specific function, created, deleted or changed by it, which must exist or be specified to perform the function etc. For example, the node concerning with the function Polygon is associated with the concepts of the client area of the window, device, coordinate system, pen, brush, raster operation.

The examples of metaknowledge:

- to use or get access to any object its identifier must be available;
- when some object is created it must eventually be deleted.

Now suppose that the user needs to solve the following problem: when somebody attempts to trigger the second instance of an application it is necessary to activate instead the first instance of it. In response to the user's request the HTMS activates the semantic net vertices "Application", "Instance of an application", "Main window of an application", "The state of a window" and takes into consideration the P–relation "change". Based on the links between the concepts and using the metaknowledge the system activates also "Instance handle", "Window", "Window handle" and considers the P–relations "find", "determine", "use". This allows to pick the following nodes of the hypertext: "WinMain function", "IsIconic function", "OpenIcon function", "'SetActiveWindow function" and the nodes describing possible states of the window. The nodes "SetActiveWindow function" and "OpenIcon function" are marked as the basic (because they change the window state), the other ones are auxiliary.

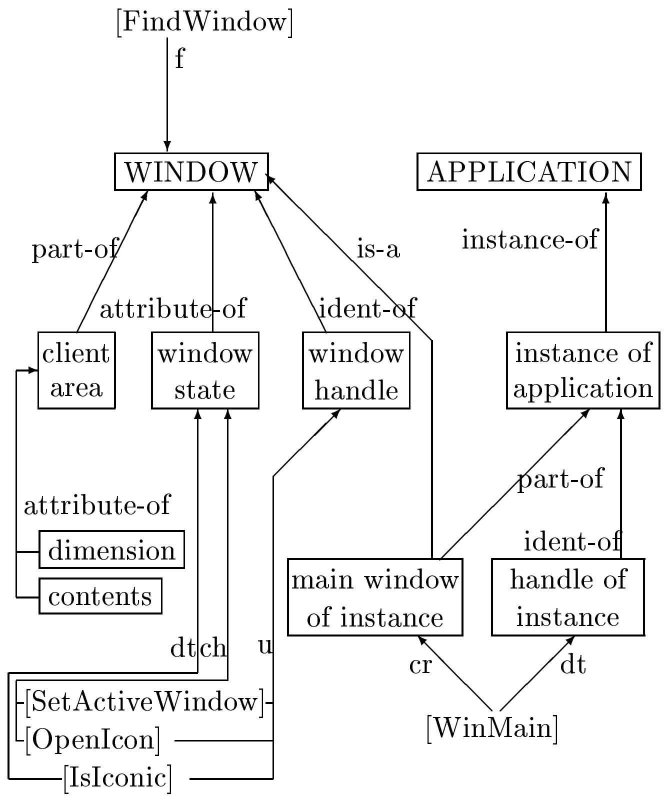

Fig.1. A fragment of the semantic net and the related hypertext nodes. The names of P–relations are abbreviated: f–find, ch–change, dt–determine, cr–create, u–use. The titles of the hypertext nodes are given in square brackets.

4 Knowledge acquisition

We have some experience in designing methods and scenarios of interviewing an expert to acquire knowledge for expert systems intended to solve different types of problems. Some of them (they are implemented in the knowledge acquisition system AFORIZM) seem applicable for the purpose of acquisition of the domain theory lying behind the hypertext material.

First, the method of grouping (or bottom-up structuring) was applied to obtain the domain concepts of the higher level of abstraction and to arrange them into the hierarchical structures. The idea is to induce the expert to pick groups of related items from some collection of items and to give names to the groups. Then the groups, in their turn, can be united into the higher–order concepts. In this process various associations between items can be exploited depending on the type of the problem under analysis. Here are few examples:

(a) the items are united by similarity; this allows to obtain the ISA-hierarchy;

(b) the structural relations are considered, e.g. the expert unites into a group the concepts denoting the units that can be the parts of the same aggregate;

(c) the situational grouping: related are the items which can be present together in one and the same situation.

It seems possible to apply this method to the hypertext nodes viewing them as the items to group.

Second, the reverse procedure of the top–down structuring is useful when we have not the initially available collection of domain concepts of a low abstraction level. The expert is proposed to divide mentally the whole area of his professional conceptions and ideas into several subareas and to give substantial names to these subareas. Then this expedient is applied recursively to the revealed subareas etc.

From the other methods developed to use in building expert systems the relation analysis method seems promising. This method was applied to obtain the preconditions and postconditions of actions to be used in solving planning tasks. It is based on a set of specially chosen relations which can be observed between actions, the role of relations being the following:

(1) the expert is suggested to decide which of them associate the given pair of action;

(2) when explaining his opinion the expert mentions the domain concepts significant for the problem under analysis; moreover, he formulates in their terms the pre– and postconditions of the actions — just the knowledge we need.

For example, when the expert thinks that doing the action A can make possible to do the action B his explanation of this phenomenon allows to acquire the postconditions of A and the preconditions of B (and, evidently, the domain concepts in terms of which these conditions are formulated).

It should be mentioned that this method cannot be applied directly to the arbitrary hypertext irrespective of its contents. The set of relations exploited in it may be useful in hypertexts supporting some practical activities (e.g. programming in the Windows environment: the functions of Windows are the actions which can be associated by these relations). On the other hand, it is possible to find relations with similar opportunities for knowledge elicitation to apply in other types of problem domains. It is worth considering the groups of spatial, temporal, cause–and–effect relations, the relations between actions and their participants (agent, object, instrument and so on).

5 Discussion

The approach suggested requires a considerable effort to acquire the domain model and to establish the P–relations of the hypertext nodes with the domain concepts. However the result is the increased friendliness of the system towards the user's needs. The most significant in this context is the ability of the system to select and arrange the relevant information that allows the user to carry on investigation, problem solving or education more effectively than using the ordinary hypertext.

Let us list the useful opportunities created at the sacrifice of the formalization of hypertext and incorporating knowledge base in it:

- an automated selection of information relevant to the user's problems and needs;

- automated analysis of the contents of a node;
- generation of the user's subsets (views) of hypertext nodes and links based on the knowledge of node contents and the meaning of links;
- generation of new nodes and links on the user's request;
- the possibility to change the appearance of a node to mark the most significant information.

6 Related works

The approach suggested can be seen as the outgrowth of the idea of virtual hypertext structures, or "views", realized in the gIBIS system (Conclin and Begeman, 1988). The specific feature of our approach is the automated generation of a view due to the presence of the domain knowledge base that makes possible the expeditious response to the information needs of the user.

On the other hand, the given approach is based on the previous work of the authors on the automated creation of the hypertext to reflect the structure and contents of a knowledge base in a readable form (Andrienko and Andrienko, 1992).

The close ideas were found in (Baudin et al, 1993) where the domain knowledge base is proposed for the purpose of indexing in multimedia systems.

7 Conclusion

The paper offers the approach to the further development of the hypertext from the standpoint of enhancing its intelligent and partner's properties rather than amplifying its technical power. We see the main direction of our further work in the automation of building knowledge base and indexing the hypertext nodes.

References

[1] Conclin J., Begeman M.L. gIBIS: a hypertext tool for exploratory policy discussion, ACM Trans. Office Inform. Syst. N 10, p.140-152, 1988.

[2] Andrienko N.V., Andrienko G.L. The hypertext in the knowledge acquisition technology AFORIZM. Applied knowledge based systems, Kishinev, Shtiintsa, p.3-14, 1992 (Russian).

[3] Baudin C., Underwood J.G., Baya V. Using device models to facilitate the retrieval of multimedia design information. 13th IJCAI–93 proceedings, Chambery, France, v.2, p.1237-1243, 1993.

Gennady L.Andrienko, Nathalia V.Andrienko,
Pushchino State University,
B-35-38, 142292 Pushchino, Moscow reg, Russia
phone: (095) 923-80-03
e-mail and@adm.pgu.serpukhov.su

Received July 29, 1994