Graphical Explanations for Large Semantic Networks

R. Prechtel, H. Niemann, E. Nöth
Lehrstuhl für Informatik 5 (Mustererkennung)
Universität Erlangen-Nürnberg
Martensstr. 3
8520 Erlangen, F.R.G.

G. Sagerer
AG Angewandte Informatik
Universität Bielefeld
Postfach 8640
4800 Bielefeld 1, F.R.G.

Abstract

In this paper we present ELK (**E**xplanations for **L**arge **K**nowledge bases), the explanation facility of our system shell for knowledge based pattern analysis ERNEST (**ER**langen **NE**twork **S**ystem and **T**ools). In ERNEST semantic networks are used as knowledge representation formalism.
Our major goal in the development of ELK is to explain and visualize large knowledge bases with more than 1000 concepts in an easily understandable way. Therefore we first reduce the information of each graphic if necessary. For this compression we use seven principles which will be discussed in this paper.
All graphics in ELK are generated automatically. Fast layout algorithms are integrated for grids, trees, hierarchies, neighborhoods, diagrams, tables, and text. In order to facilitate understanding a clear layout was achieved by ordering and grouping the objects in a proper way.
The user interface gives the user an overview of all available explanations. The objects in each graphic are pickable to provide access to further details.

1. Introduction

One advantage of knowledge based systems is that powerful explanation facilities can be provided. The use of such an explanation tool is twofold:

- It helps the system designer to develop, inspect, and debug the knowledge base and the intermediate and final results of an analysis.
- The acceptance of the system is improved because the user can get explanations of what the system does, how it works, why its actions are appropriate, and when the limits of the system are reached.

Many explanation facilities use a natural language approach. The paradigm is a *dialog* between the user and the system. The user asks questions, typical are 'WHY'- and 'HOW'-questions which were first used in MYCIN [2]. To be able to give a specific answer the expert system needs a highly structured knowledge base [8], for example control knowledge ought to be stored explicitly [4]. The expert system also needs a deep model about the domain to provide justifications [3,8].

However, experts only use dialogs when explaining a limited problem. If the problem is more complex the approach becomes different. Examples of this different approach are lectures, books, and operating instructions. The approach has the following characteristics:

1. The explanation in a lecture is much more extensive than the (short) answer in a dialog. It is *refering to a theme* and not only to one specific question.
2. A *good structure* is the key to every good explanation. An overview must be given and the details must be arranged in an adequate sequence.
3. Similar to the paradigm of a dialog, the statements must be *tailored* to the people being addressed.
4. Some kind of information, e.g. about complex relationships, is provided by *graphics.*

According to the first three points of this *lecture-based* paradigm our first goal was to define single explanation units which can easily be understood. The second step is to layout the data in graphics or natural text. We use graphical explanations extensively because they are more efficient than natural language explanations with respect to showing the structure of the network, facilitating the comparison between different results, and displaying results in our application domain of image and speech understanding. Finally, a user interface must be provided. Therefore, we have three modules: the extraction of data, the layout, and the user interface.

Our explanation facility is part of the **ER**langen **NE**twork **S**ystem and **T**ools ERNEST. The knowledge representation formalism of ERNEST is based on semantic networks. We have three node types: *concepts* building up the knowledge base, *instances* for the results and *modified concepts* which are concepts with data driven constraints. There also exist five link types: *specialization* links defining the relation between general and specialized concepts, *part* links connecting a concept with its parts, *concrete* links separating different conceptual systems, *model* links which are used by the automatic knowledge acquistion module and *instance* links between concepts and instances.

The concepts are frame like structures. The most important slots are the slots for the five link types and slots for *attributes*, *relations* and *judgments*.

Besides of the explanation facility, problem independent modules are provided in ERNEST for the control and the automatic knowledge acquistion. The control module uses the A^* algorithm. More details about ERNEST are provided in [7].

Three parts of ERNEST have to be explained by the explanation facility: the declarative knowledge, the procedural knowledge, and the results. It is intended to implement with ERNEST semantic networks with more than 1000 concepts. The explanation of large knowledge bases is one main goal of our work. Therefore, we called the explanation facility ELK — **E**xplanations for **L**arge **K**nowledge bases. In this paper the three modules of ELK the extraction of data, the layout, and the user interface will be discussed.

2. The extraction of data

The extraction module is the interface between ELK and ERNEST. Because large knowledge bases must be handled we cannot display all of the information in one single graphic of the size of a workstation screen. We must solve the problem of extracting the information in such a way that one single explanation unit is understandable.

We use seven principles to separate and compress the available information:

1. The definition of explanation steps in explanation sequences.
2. The separation of static knowledge and dynamically generated results.
3. The separation of different levels of detail.
4. The separation of different conceptual systems and topics.
5. The extraction of 'important' information.
6. The adaptation based on the user group and the user knowledge.
7. The adaptation according to the current interests of the user.

Every explanation in a book or a lecture needs a 'read thread'. Therefore, the definition of explanation sequences with several explanation steps is the motivation of all separation, extraction and adaptation ideas. If you want to explain something, you must know what the single explanation steps are and how they can be ordered. This must be done in reference to the people being adressed. The same problems must be solved in an explanation facility for large knowledge bases.

Different explanation sequences must be provided for the explanation of the static knowledge in the system and the dynamic use of the knowledge. The first sequence gives an introduction to the system, the second one helps to check the results. ERNEST supports the separation of static knowledge and dynamically generated results.

In an explanation one will first give an overview and then discuss details. Therefore, a separation of different levels of detail is needed. In ELK the user can get an overview of the network, then the graphic of one conceptual system, then details about one concept, e.g. about the attributes (see Figure 1), and finally details about specific procedures. The overview of a network will be overloaded if 1000 concepts are supposed to be displayed at once. Therefore, we defined *concept families* summarizing similar concepts. The results are structured in a similar way.

Within a level of detail different conceptual systems and topics can be distinguished. In ERNEST *specialization* and *part* links are only allowed within a conceptual system. For the connection between different conceptual systems exists the special link type *concrete*. In a such a way the conceptual systems are clearly separated. Also each concept has different slots defining explanation topics like attributes or relations (see Figure 1).

For each topic you can focus on the 'important' information. However, we need a definition which information is 'important'. One example of such a definition in ELK is concerned with the search tree of ERNEST which is generated during the analysis of a signal. There exist search trees with several thousands of nodes, which makes a compression of this information very desirable. We provide a fisheye-view [5], the pruning of the tree, and the deletion of special node types. Figure 2 shows one example. Here is the focus on paths with a high judgment, leaves, and nodes where the tree branches.

The explanation system is used by different users with different knowledge. A novice will need more information about the underlying ERNEST ideas and less information about confusing details of the knowledge base. On the other hand an experienced system developer needs quick and efficient access to every detail in the net without being distracted

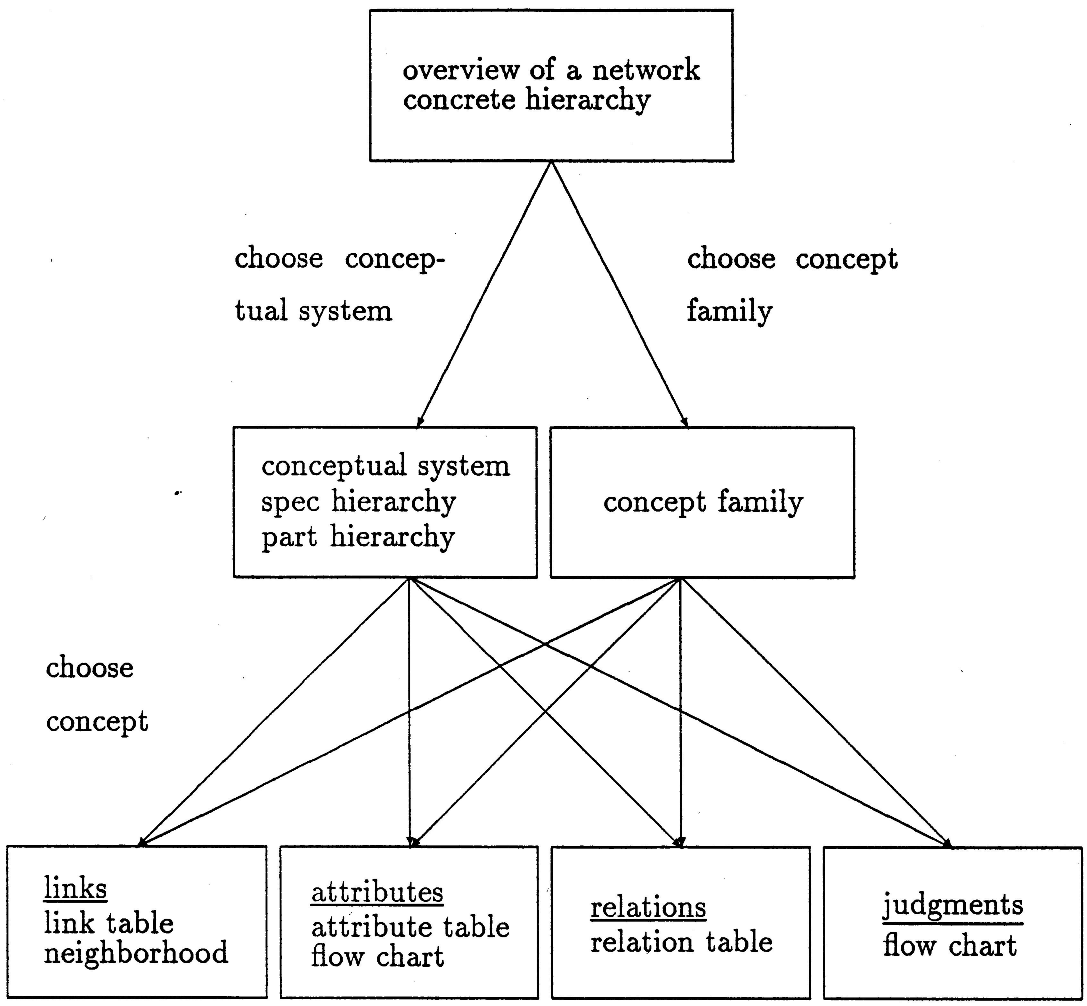

Figure 1: Graphical explanations for the declarative knowledge in ERNEST

by information he already knows. For the intermediate steps between novice and expert it is necessary to access to the knowledge which the user has gained up to this point. To adapt the explanation sequence automatically to the user a user model [6] is needed.

Finally, the explanation facility should provide an adaptation on the current interests of the user. Here the goals of the user play the decisive role. If he wants to build a new net different explanations are needed depending on the knowledge acquisition module he uses (interactive or automatic). If he wants to get an overview of an existing network he can also be interested in some details about attributes or judgments. Displaying only the desired information will lead to an easier acceptance of the system.

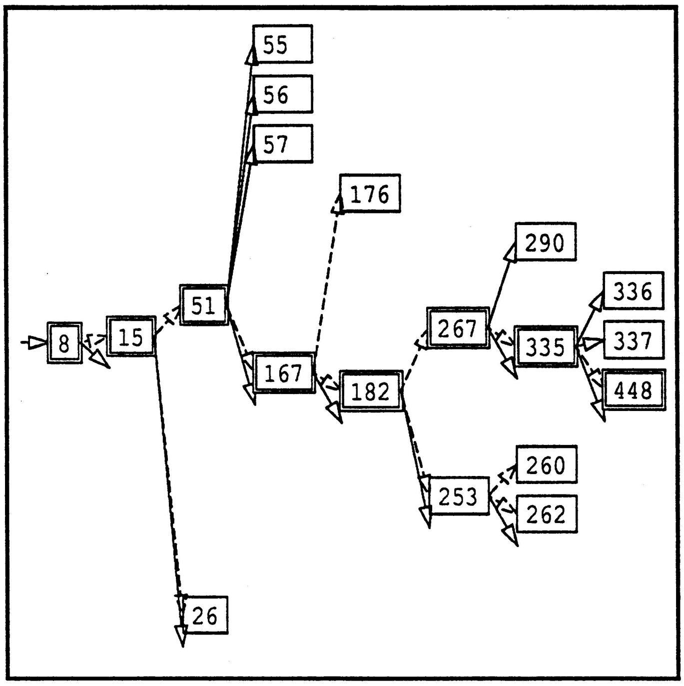

Figure 2: A part of a search tree (dashed arrows indicate a sequence of nodes without branches, arrows with no goal node mark the deletion of paths with a low judgment)

3. The layout

It was our goal to achieve graphics which are easy to understand. Our approach follows the ideas and results presented by Bertin [1]. He describes which characteristics a graphic must have to provide answers to different questions in one single moment of inspection.

To achieve a graphic which is understandable in one single moment one must first analyse the given information. The *invariants* of the information and the *components* which are changing must be found. The components must be examined to derive their number and to obtain the length and the structure of each component. There exist four structure types: *associative* (e.g. all red objects), *selective* (e.g. distinguishing between red and blue objects), *ordered* (e.g. small, medium, high), and *quantitative* (e.g. 1, 2, 3, ...).

Now the given information must be displayed with the media of a graphical system: the *two dimensions of the plane* and the six so called *color pattern variables*: *size*, *brightness*, *pattern*, *color*, *direction*, and *form*. With the two dimensions of the plane every structure type can be shown. In most cases this is also the best representation. However, not every color pattern variable can represent every structure type in such a way that it is understandable in one single moment, e.g. a quantitative structure can only be shown by size (in other representations one must carefully study the legend before one can understand the quantitative structure).

In ERNEST we have several criteria to structure the information:

1. We defined for every link a *degree*, for example a concept with no part links gets the part degree 0 and concepts with parts get a part degree equal to the distance along part links to a concept with no parts. Therefore, the concepts can be ordered along their link degrees.

2. Attributes depend on the results of other attributes. Therefore, the analysis procedures carry out an adequate execution sequence. These dependencies can be used for ordering the attributes.

3. The nodes of the search tree should be ordered according to their depth.

We have six layout types based on how much and which structuring information is present:

- *grids.*
  Here we have two discrete axes defining a grid for the plane. For example, the overview of a network has the two axes concrete and spec degree (see Figure 3).

- *hierarchies.*
  Here we have only one discrete axis. The second dimension of the plane can be used to minimize the number of crossings or the length of the connection lines.

- *trees.*
  Here we also have only one discrete axis, the depth of the tree, but it is easy to construct a graphic without crossings. To minimize the length of the connection lines every node is placed in the vicinity of its father (see Figure 2).

- *neighborhoods.*
  Here only the direct predecessors and successors are displayed, i.e. the length of the components is only 3. Therefore, it is possible to use the two dimensions of the plane and also the two diagonals and so we can represent four quantitative structures (see Figure 4).

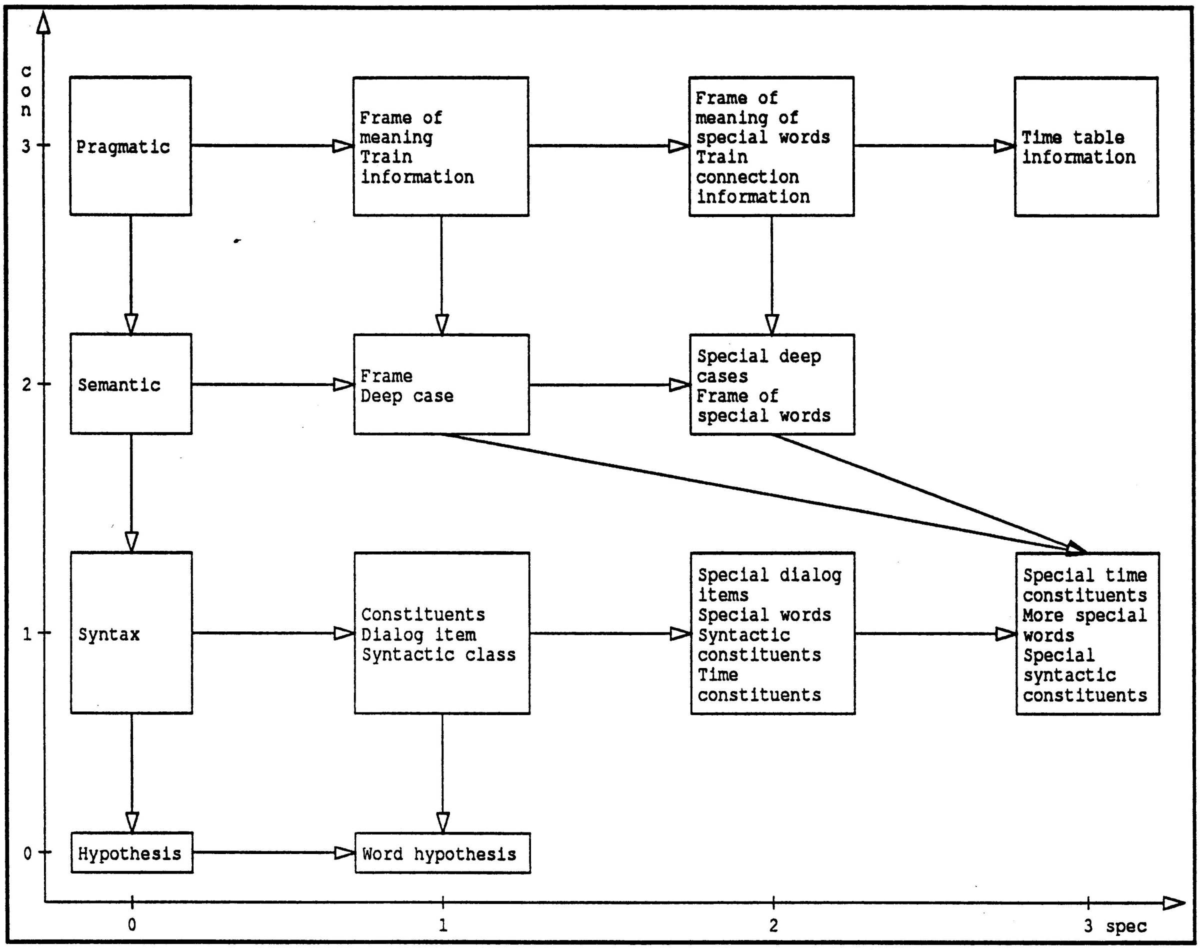

Figure 3: An automatically generated overview of a network in ERNEST

- *diagrams.*

  Here the two dimensions of the plane are used to show a quantitative structure in the horizontal direction and an associative structure in the vertical direction (see Figure 5).

- *tables.*

  In a table both directions of the plane represent an associative structure.

Besides these layout types, text and pictorial images can be created in ERNEST. When the graphic is larger than the display, efficient navigation techniques are provided, for example, to travel along a path in the search tree.

So far we have not used the color pattern variables. Bertin proved that *form* can only show an associative structure in one single moment and *direction* only an associative and a

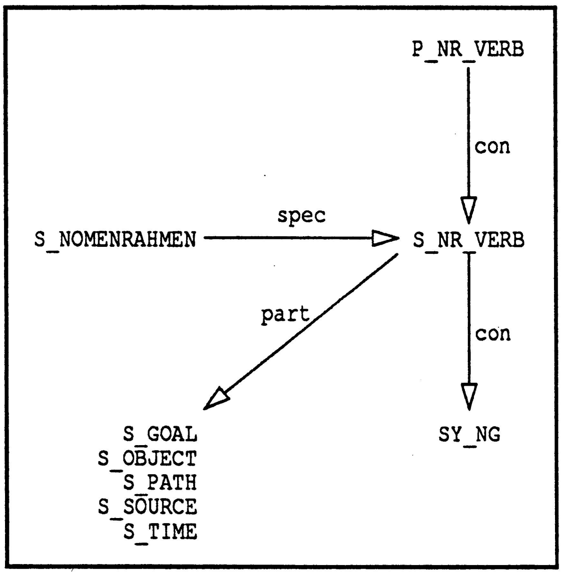

Figure 4: A neighborhood

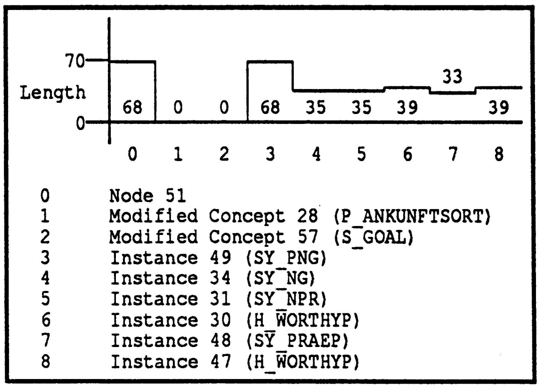

Figure 5: A diagram. The y-axis shows the number of speech frames interpreted by the instances and modified concepts in the node 51.

selective structure. Therefore, both are not very important. *Color* and *brightness* cannot be used on a monochrome display or hardcopy and by using *size* precious display size would be wasted. We want to use *pattern* to distinguish different node types. In ERNEST some intermediate node types exist between concept and instance. This ordered structure can be shown by pattern, e.g. with a striped pattern using different distances between the stripes.

We think that the figures in this section are good examples of the easily understandable layout strategy we use. Only in the overview of the network (Figure 3) the whole net must be presented. Using our compression principles only the concept families (20 instead of over 100 concepts) and the structure in relation to the concrete and spec link is shown.

4. The user interface

The user can access the graphic with a command and a menu interface. For each command and menu and for each ERNEST term a passive, context sensitive help is provided. The objects in the graphics are pickable to allow a quick access to further details. All choices listed in Figure 1 can be done by selecting the objects in the graphics.

5. Summary

One major goal at the development of ELK was to attain automatically generated graphics which are easy to understand. Therefore, we tried to separate the knowledge in single explanation units, we extract 'important' knowledge, and the displayed information can be adapted to a specific user. The layout of the graphics is clearly structured.

For the expert user we provide a command interface and a menu interface. By picking the objects in a graphic detail information can be reached in an efficient way.

The extraction of the information, the layout programs, and the interfaces are implemented in C under UNIX using X11 and MOTIF.

The next step is to implement an interface for novice and intermediate users exploiting a user model and providing automatically created explanation sequences. We want to use ERNEST to represent the user model and the knowledge about the explanations.

6. Acknowledgements

This research is supported by the DFG (Ni-191/7).

References

1. Bertin, J.: Graphische Semiologie. Berlin, New York: de Gruyter 1974.
2. Buchanan, B.G.; Shortliffe, E.H. (eds.): Rule-Based Expert Systems. Reading: Addison-Wesley 1984.
3. Chandrasekaran, B.; Tanner, M.C.; Josephson, J.R.: Explanation: The Role of Control Strategies and Deep Models. In Hendler, J.A. (ed.): Expert Systems: The User Interface. Norwood: Ablex Publishing Corporation, p. 219-247, 1988.
4. Clancey, W.J.; Letsinger, R.: NEOMYCIN: Reconfiguring a Rule-based Expert System for Application of Teaching. In Proc. 7th IJCAI, Vancouver, p. 829-836, 1981.
5. Furnas, G.W.: Generalized Fisheye Views. In Mantei, M.; Orbeton, P. (eds.): Proc. CHI 86 Human Factors in Computing Systems. New York, p. 9-15, 1986.
6. Kobsa, A.; Wahlster, W. (eds.): User Models in Dialog Systems. Berlin, Heidelberg, New York, Tokyo: Springer-Verlag 1989.
7. Niemann, H.; Sagerer, G.; Schröder, S.; Kummert, F.: ERNEST: A Semantic Network System for Pattern Understanding. IEEE Trans. on Pattern Analysis and Machine Intelligence, 12:883-905, 1990.
8. Swartout, W.R.: Knowledge Needed for Expert System Explanation. Future Computing Systems 1(2):99-114, 1986.