Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT

HyperCAKE: A Knowledge Acquisition Tool for Hypermedia-Based Expert Systems

Frank Maurer

SEKI Report SR-92-04 (SFB)

HyperCAKE: A Knowledge Acquisition Tool for Hypermedia-Based Expert Systems

Frank Maurer
University of Kaiserslautern
AG Prof. Dr. M. M. Richter
PO Box 3049
6750 Kaiserslautern
Germany
e-Mail: maurer@informatik.uni-kl.de

Abstract

We describe the architecture of the *multi-user* knowledge acquisition tool HyperCake which is designed to support the development of hypermedia-based expert systems. HyperCAKE contains a multimedia database system for the management of the knowledge base, mechanisms for the management of the hypertext, a consistency checker (based on graph grammars), modules which support the KADS methodology, and tools for the development of application-specific shells. Further, we describe three knowledge level shells which are developed with HyperCAKE. One supports the development of expert systems for technical diagnosis. The second is a knowledge based geographical information system for data analysis. The last realiyes an intelligent development plan.

Keywords

hypermedia, expert systems, knowledge acquisition, multi-user

1. Introduction and Overview

In this paper, we describe the architecture of a knowledge acquisition tool for the development of hypermedia-based expert systems. The underlying assumption of our work can be best described by the the following expectations for future software generations:

- The results of research in expert systems (or more general: AI) will be integrated into the next software generations. This will lead to more intelligent solutions and applications.
- The integration of multimedia will lead to a revolution in the man-machine-interface. The bandwidth of the communication between man and machine will be drastically increased by audio, video and computer animation.
- The complexity of software will increase. Therefore software will be developed by groups of programmers.

These assumptions lead to additional requirements on the next generation of expert systems:

- Expert systems have to integrate multimedia aspects. First, to improve the knowledge acquisition. Second, to extend the user interface. The integration has to overcome the mere coupling of an expert system to a hypermedia system as an intelligent on-line help (see for example [GainesLinster90], [KarbachLinster90] and [Angstmann91]). What is more, the expert system must be able to reason about the knowledge structure of a hypermedia system.

- In future knowledge-based systems will be developed by large teams of knowledge engineers and domain experts. This process has to be supported by development tools which must have multi-user capabilities. The development tool has to preserve the integrity of the knowledge base, support the communication between experts and knowledge engineers, to document and simplify the development process from the beginning up to the maintenance phase, and allow the development of problem-specific tools.

- Future expert system will not be stand-alone solutions, but must be integrated in conventional software.

- Because of economical reasons the development of expert systems must be divided into several steps, where every step must result in a useful solution for the customer. A useful result is a documentation of the domain so that an (educated) user will be able to solve problems which only experts were able to solve. This result may be a hypertext. The hypertext is used as a kind of conceptual model where the inference engine used for operationalization is a "educated layman". The replacement of the human by an interpreter needs much effort, but results in a reduction of the (required) skills of the end-user (cf. fig. 1).

The requirements described above show several methodological deficits of the today´s techniques:

A main problem of the present techniques is the sharp boundary between context-dependent knowledge, which only may be interpreted in a human context (i.e. with common sense), and the representation of this knowledge, which may be interpreted by a formal system of rules without reference to contextual information. Our Approach tries to support a "smooth" transition between context-dependent and context-independent knowledge by the integration of different knowledge representation schemes.

The sharp boundary between context-dependent and context-independent knowledge is also a reason of the well-known "knowledge acquisition bottleneck". KADS tries to overcome the resulting methodological gap by increasing the expressive power of the formal language (for conceptual models). We try to decrease the gap by a stronger structuring of the context-dependent level.

Beside the problem of context-dependency there arise difficulties because of the use of different media for storing the different types of knowledge. Context-dependent knowledge is stored in books, journals, drawings (e.g. architecture) and nowadays this kind of knowledge sometimes is stored in hypertexts. Formalized knowledge is stored in expert systems. The transition from the context-dependent level to the context-independent level becomes easier if the tools for storing and retrieving the knowledge are integrated.

Large knowledge bases may only be developed by groups of people because of the amount of data. This increases the problem of checking the logical consistency because one developer does not always know what his colleagues have stored. The methodology of consistency checking is not sufficiently developed for systems which integrate context-dependent and context-independent knowledge.

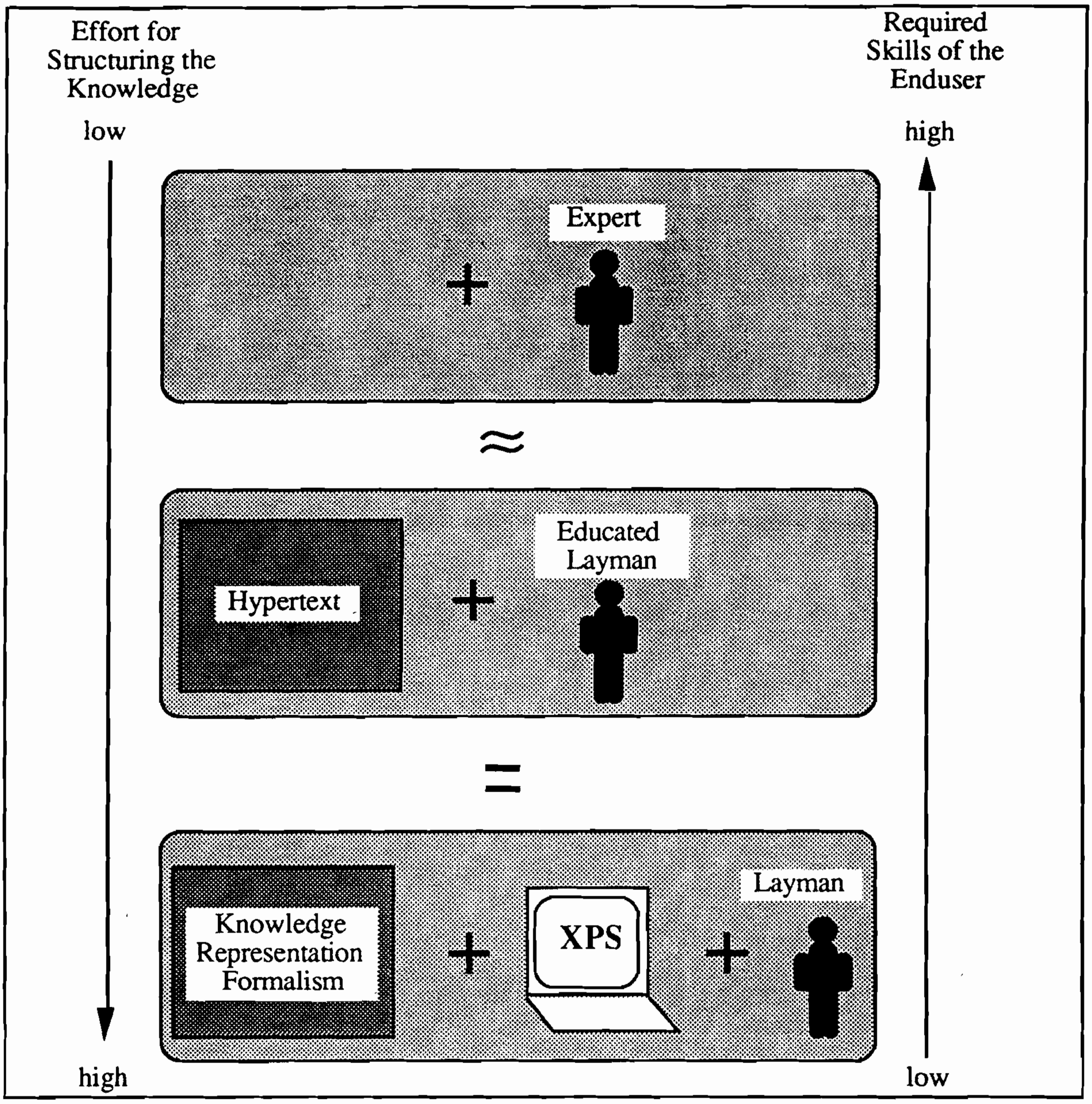

Fig. 1: Supposed relation between expert, hypertext, and knowledge base

Within this paper we describe several software systems. For clarification purposes we introduce now its names. HyperCAKE is our knowledge acquisition tool. The integrated inference engine is called HyperXPS. WisGIS, HyDi, and IntPlan are application-specific shells.

In chapter 2, the architecture of a tool is described which was developed according to the above stated requirements, especially the multi-media aspects and the multi-user capabilities. In chapter 3, we present three application which were built with the developed tool: WisGIS is a knowledge-based geographical information system for the prognostication of the quality of soil; HyDi is used for the heuristic, associative diagnosis of technical systems; the IntPlan-System realizes an intelligent development plan. An evaluation of our approach is given in chapter 4 whereas in chapter 5 we describe the state of realization.

1.1. Terminology

Within this paper we use the following terminology. *Knowledge acquisition* consists of (initial) data collection, model construction, and knowledge elicitation.

Data collection uses interviews, technical literature, drawings etc. and provides an overview of the domain which is documented in the data model (cf. [Maurer91]). The *data level* is located between the *level of verbal data (M1)* and the *conceptual model (M2)* in KADS.

For the model construction process we use the KADS terminology.

Knowledge elicitation fills the formal structures defined in the conceptual model with the concrete domain knowledge. The knowledge elicitation phase results in operational knowledge.

The described phases are not carried out in a sequential order (following the waterfall model) but contain loops and feedback. HyperCAKE is being developed to integrate rapid prototyping with model-based knowledge acquisition.

2. System Architecture

Here we give an overview on the overall architecture of the knowledge acquisition tool HyperCAKE. Detailed descriptions of the modules are given in the subsections of this chapter.

HyperCAKE uses the object-oriented database system GemStone[1] for the management of the objects of the knowledge base. With its transaction concept GemStone guarantees the integrity of the knowledge base in a multi-user-environment.

Based on the object management system we implemented a hypertext abstract machine (HAM) following the ideas of [CampbellGoodmann88]. The HAM allows to structure a knowledge base.

The implemented HAM manages the hypermedia network. It allows to create and delete nodes (= objects) and links (= relations between objects). The HAM uses *Contexts* to structure the knowledge base statically. Dynamically different views on the network may be created by means of a filtering mechanism which supports a net-oriented query language.

Based on the HAM we developed the tool HyperXPS which allows to define application-specific shells. HyperXPS enables the knowledge engineer to define typed nodes and links with a specific semantic. HyperXPS includes a rule language and the corresponding interpreter.

---

1 GemStone is a registered trademark of Servio Corporation.

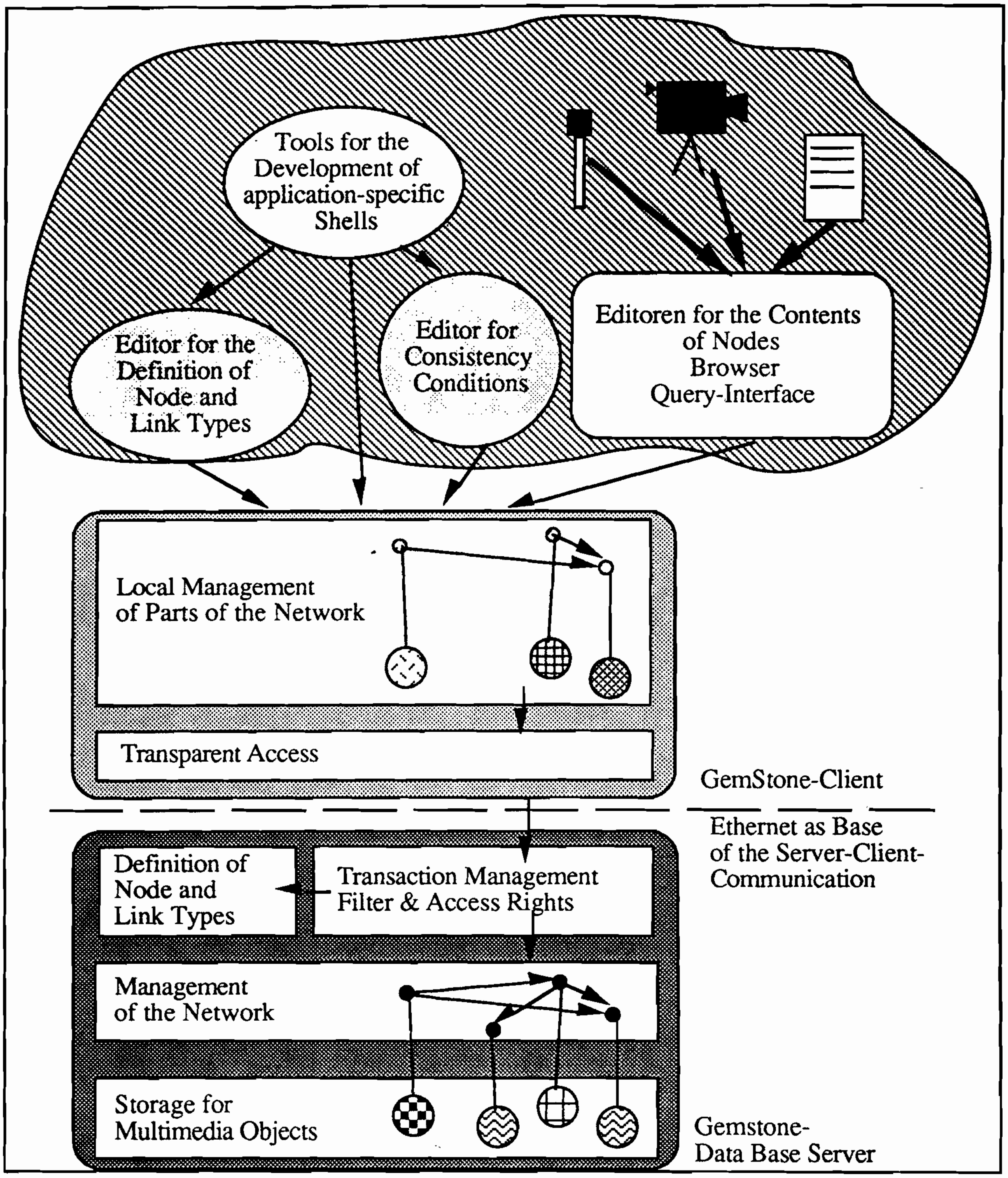

Fig.. 2: The overall architecture of HyperCAKE

2.1. The Database for Object Management

A knowledge acquisition tool has to support teams of knowledge engineers and domain experts in the development of knowledge-based applications. A developer must be able to work on a common knowledge base from its own workstation.

Multimedia Objects

HyperCAKE permits the management of multimedia information: text, picture, audio, graphic and video. Additionally, the storing of structured information like tables and concept descriptions is possible. Thus, HyperCAKE integrates the management of context-dependent knowledge with possibilities for the representation of context-independent. Only this tied coupling allows to document the knowledge acquisition process because all gathered information, from the data level via the conceptual model to the implementation level, is managed in one knowledge base.

Access Rights and Transaction Management

Every object in the knowledge base is related with permisions for read and write access. We distinguish between the owner of the object, groups of users and the world. Using the access mechanism, an user is able to create private data. Thus, he may use the system to document "immature" ideas without fearing disadvantages.

GemStone supports transaction management to guarantee the integrity of the knowledge base. Based on this, HyperCAKE ensures that information is only integrated into the knowledge base if it is not locked by another user.

Additionally, the consistency checker described in section 2.3 helps to maintain the logical consistency of the network extending the possibilities of the transaction management.

2.2. The Typed HAM

The question, how semantic is linked to information, leads us to an extension of the HAM with an (object-oriented) type concept[2]. Based on the polymorphism of the used implementation language we can replace generic methods (e.g. show, edit, followLink, getSuccessors, getPredecessors, set:, etc.) by specific ones. This allows to define new semantics for special information classes.

Nodes

Nodes store atomic information units as texts, graphics, tables, concept descriptions, knowledge sources etc. Nodes are the general means for the representation and management of knowledge. Fig. 3 shows a part of the class hierarchy for nodes.

HAMNode		
	TextNode	# Texts
	GraphicNode	# Graphics
	SoundNode	# Audiodata
	VideoNode	# Videodata
	Context	# Sets of objects
	Keyword	# Keywords

Fig. 3: Node classes of the HAM

Links

To represent relations and associations between information nodes we use links. The present version of the HAM supports directed, binary links. The starting point of a link is a (part of)

---

2 Types in the meaning of abstract data types, where data is linked with operations.

a node. Target of a link is always an entire node. To implement semantics for different kind of links we support a set of link classes which is (partially) shown in fig. 4

```
HAMLink
        ContextLink
        IsALink
        PartOfLink
```

Fig. 4: Link classes of the HAM

Links describe static relations between information nodes and are explicitly stored. Additionally, the HAM allows to define dynamic links which are computed if required based on the actual structure of the network. Dynamic links represent associations between nodes which are not direct neighbors. We use them within the WisGIS-System (see chapter 3) to compute the relations between input- and output-attributes. Then, the user can inspect the computed links.

Modularization of the Knowledge Base

A basic assumption for the development of HyperCAKE is, that there is a huge amount of information to be managed within a expert system project. Therefore, we use a database. On the other hand there was a strong need for means of modularization of the knowledge base. [CampbellGoodmann88] used a special concept called *Context* for structuring the network. Another possibility are so-called *Composites* ([Halasz88]) together with a specialized link class. We use the second possibility: we define a node class *Context* and a link class *ContextLink*.

Semantic Network

The implemented HAM allows the representation of knowledge by a semantic network. The nodes of the semantic network may contain multimedia information. The "semantic" is defined by the implementation of the functionality for the node and link classes.

2.3. Consistency Checking

While developing large knowledge bases consistency checking is a problem. This comes to the fore especially in a multi-user environment because each developer only has insufficient knowledge about the work of his colleagues. From this finding results the demand for a support in consistency checking by the used tool, in our case HyperCAKE. It is clear that a system only is able to check structures which it *knows* and *understands*, i.e. a consistency checker always works with a kind of closed-world-assumption. Looking from the hypermedia point of view it is not possible to use the contents of texts and pictures within the checking process (to use them we would need computer vision and text recognition systems which are far beyond the current state of the art). Therefore, in HyperCAKE we only check

1) the consistency of the graph structure

2) the formalized knowledge.

For this we extend the ideas of [Maurer91b] for consistency checking in the MOLTKE-System.

Definition of Consistency Conditions

In HyperCAKE consistency conditions are specified by means of graph grammars. A condition describes a subnet. It consists of a set of node and path specifications together with

an action. The condition is matched with the hypermedia network. If it is fulfilled the action is carried out. Possible actions are:

- output a warning to the user,
- delete the subnet,
- reject the newly entered node or link,
- create an entry to the agenda (see below) for the missing nodes or links.

To increase the efficiency of the graph matching we use a kind of rete network (cf. [Forgy82]).

Agenda Management for Missing Objects

A user-friendly knowledge editor has to allow the input of knowledge in arbitrary order. Especially in network-based representation the sequence of insert operations has a heuristical nature (i.e. a specific order is not compelling). HyperCAKE permits to enter parts of graphs in arbitrary order. Inconsistent subgraphs are stored on an agenda. Nodes and links which are stored in the agenda are not used in the inference process. After the insertion of a new object HyperCAKE checks if it is stored in the agenda. If true, it checks if the subgraph now is consistent and deletes it from the agenda if appropriate.

2.4. Specification of Conceptual Models

The specification of a conceptual model is represented easily in the structures of the HAM. Typically, conceptual models are described as nets of nodes and links[3].

HyperCAKE offers the additional advantages of a multi-user environment and the possibilities for the definition of consistency conditions (which e.g. allow to describe that Knowledge-Source-1 must not be used with Knowledge-Source-2).

2.5. The Integrated Inference Engine (HyperXPS)

The HyperCAKE system is built to support the knowledge acquisition for hypermedia-based expert systems (in the following: HyperXPS). The difference to conventional expert systems is the integration of multimedia data, the difference to conventional hypermedia systems is the integration of an inference engine. HyperXPS is not built to extend the capabilities of knowledge based systems by expert system techniques but by the integration of hypermedia.

In addition to the conventional textual descriptions, symbolic representations (e.g. concept descriptions or attribute values) can be coupled with graphics or audio signals. These can be referenced in the communication with the enduser of the expert system. The integration of additional media is absolutely needed if the normal representation of knowledge in a domain uses other media then text (e.g. architecture, town or area planning, mechanical engineering).

---

3 The integration of operational languages for conceptual models into HyperCAKE is more difficult because we need to integrate the interpreter too.

Concept Hierarchies

HyperCAKE makes an object system available for the development of application-specific shells which

- allows the definition of classes and slots
- supports slot typing and defaults,
- supports inheritance,
- allows to create instances,
- permits to define methods.

The objects are represented as HAM nodes. Inheritance is implemented by HAM links.

Rule Interpreter

We developed a (standard) rule interpreter for the processing of the knowledge. The present version includes forward propagation and read/write access to slot values. Rules are interpreted as logical implications: If the logical value of the precondition becomes false, the (formerly set) slot is reset to unknown.

The restriction on this simple rule interpreter comes from the requirements of our present application domains.

2.6. The Interface to the Knowledge Base

On top of the knowledge management system we developed a user interface for HyperCake which in configurable according to the needs of each user.

Editors for Multimedia Objects

HyperCAKE includes an editor for texts with normal functionality for the creation and formatting. Graphics are drawn with standard software and imported into a hypertext node. We developed facilities for recording, playing and cutting of audio-data.

In addition to the editors for the data level, HyperCAKE includes mask-oriented interfaces for the input of context-independent knowledge structures (e.g. objects, slots, attributes).

The editor for conceptual models allows the creation and modification of the structure of a knowledge-based system. It can be easily extended to produce a description of the conceptual model in an operational language (cf. [Wetter90], [AngeleEtAl91] or [KarbachEtAl91])

Graphical Editor for Consistency Conditions

Consistency conditions are defined with a graphical editor which allows to create nodes and links. For every node its class can be specified together with a condition which must be fulfilled by its contents. Links in the consistency network represent link classes in the hyper-media network. The user has to specify how many links may be between two nodes to match the condition. Finetly, an action is specified.

Fig. 5 shows an example with three nodes and two links. The left link demands that there is a connection between a symptom instance and its class. The connection must consist of one or

more is-a-links. The right connection demands that there is a direct connection between a class and its type. If a subnet does not match the condition it is put into the agenda.

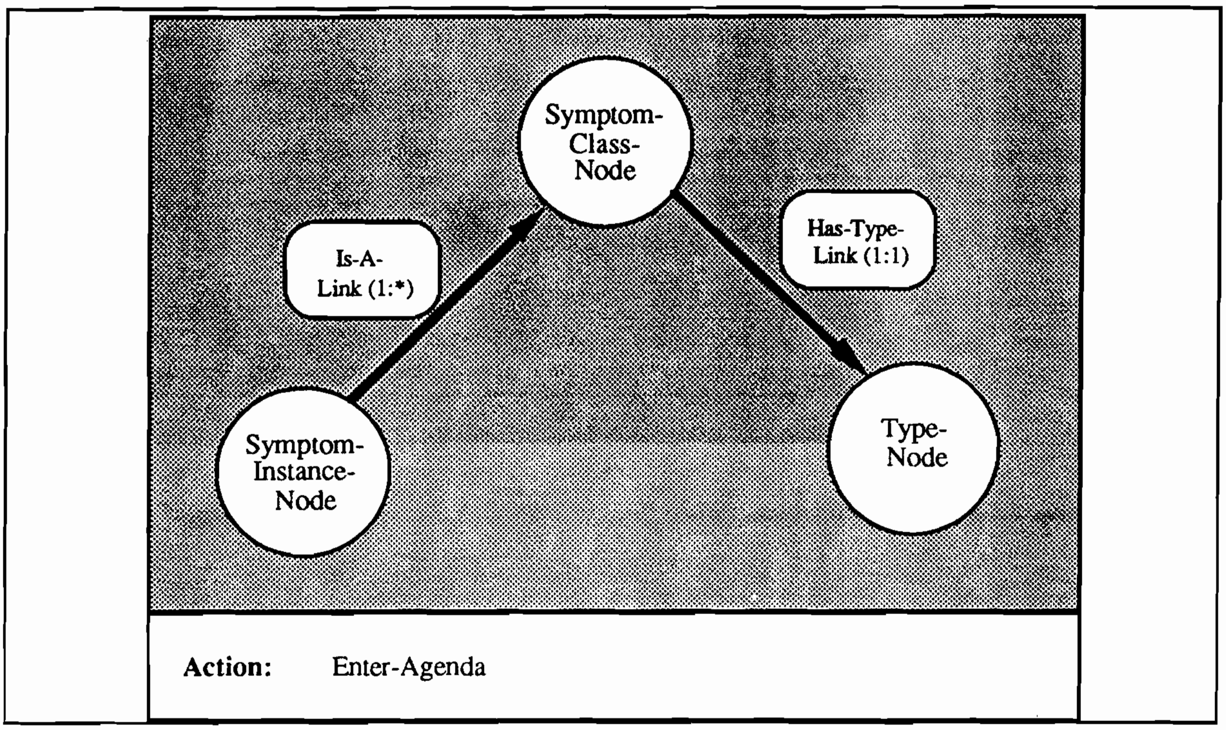

Fig. 5: A consistency condition

Definition of User-Dependent Views

Normally, the information stored in a network is to voluminous to show it entirely to the user. Therefore, HyperCAKE allows to define views based on an simple user model. The user model consists of the name of the user, his access rights, his abilities (a knowledge engineer needs another access to a knowledge base than the domain expert or end user), a filter, and a layout. The filter describes which information is shown. The access rights define the permitted operations. The layout describes the presentation of the information.

Generic Interface for the Domain Layer for Knowledge Elicitation

A conceptual model describes for the domain layer the concepts and relations of the application. Knowledge elicitation fills the defined structures with the domain knowledge. This task shall be carried out by the domain experts because they are able to check the correctness of the decisions of the expert system. Normally, domain experts are not used to computers. Therefore, a knowledge acquisition interface must support the easy development of domain specific interfaces. The effort for the building must be small to support rapid prototyping.

HyperCAKE contains a generic interface for the acquisition of domain instances. If this is not sufficient, the interface can be enhanced by aid of

- Masc generators,
- Networkbrowser,
- Texteditors,
- Views for graphics and (in a later step) video,

- possibilities to define domain dependent menus.

Hypermedia-Based Enduser Interface

The interface tool described above allows also the realization of enduser interfaces which can be used for knowledge communication, which means the presentation of knowledge to the user. For example, the main requirement in the development of the IntPlan-System was to use a graphical interface for the presentation of city maps and the knowledge.

3. Applications of HyperXPS

To check the worth of our approach we develop three application-specific shells. Each shell implements the domain specific vocabulary and interface. The first shell was developed to show that the implementation of application-specific shells based on HyperXPS may be carried out with minor efforts. The second shell demonstrates that the integration of hypermedia systems extends the possibilities of conventional expert systems. The third will evaluate our hypotheses about the connection between context-independent and context-dependent knowledge.

3.1. WisGIS

WisGIS (Wissensbasiertes Geographisches Informations System = Knowledge-based geographical information system) is a domain specific shell for the simulation of ecological systems. It was developed with the working group for water economy of Prof. Jacobitz, University of Kaiserslautern. The goal was to develop a simulation system for forecasting the quality of soil after pollution.

The input is a topological map which stores attribute-value-pairs (e.g. acid buffer capacity, amount of rain, etc.) for each area. The system abstracts quantitative data to a qualitative level. The resulting attributes are combined to predict the soil quality. Output are maps which show the combined values. The combination rules are specified by a set of rules.

The rule editor must be user friendly because the domain experts want to simulate more than one ruleset to find out how the ecological system works.

The input and output maps are stored in the commercial GIS[4] ArcINFO. So, we are able to access the visualization capabilities of the GIS.

WisGIS is fully implemented. [Hemker91] and [Jäckel91] describe the system in detail. WisGIS showed that the implementation of application-specific shells based on HyperXPS may be carried out with minor efforts.

3.2. HyDi

HyDi is a shell for the diagnosis of technical systems which demonstrates that the integration of a hypermedia system extends the possibilities of conventional expert systems. Fig. 4 shows the architecture of HyDi.

---

4 GIS = geographical information system.

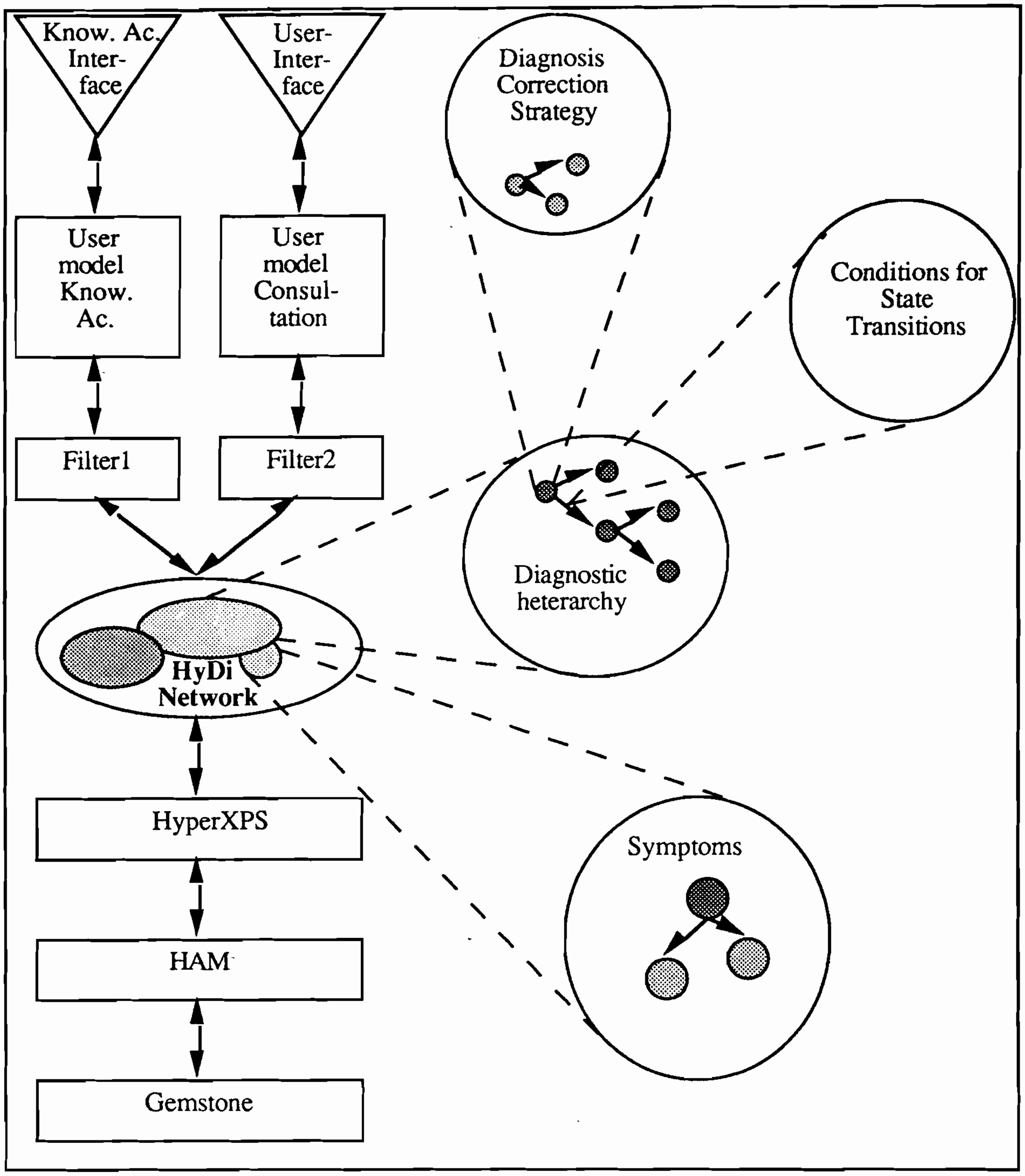

Fig. 6: The architecture of HyDi

The inference engine of HyDi follows the ideas of the MOLTKE-System ([Richter91]). HyDi includes the following representation primitives:

- Symptom: A symptom connects a name with a type, comments, and tests. Every value can have different (multimedia) presentations which are used in the interaction with the enduser.

- Diagnostic graph: A directed, acyclic graph describes the hierarchy of faults. Nodes represent faults, links describe the transition from one diagnosis to its refinements. Links are coupled with a logical condition. Each diagnosis is connected to a description of the appropriate correction.

- Strategy: The diagnostic strategy is implemented by a set of rules which describe the order of testing.

We developed different user views for HyDi based on the filter mechanism of the HAM.

3.3. IntPlan

Within the IntPlan-Project we develop an "Intelligent Development Plan" in cooperation with the working group of Prof. Streich (Department of Architecture, Civil Engineering and City Planning, University of Kaiserslautern). Within this project we evaluate the hypotheses of fig. 1 which states the correspondence of hypertext and expert systems. In the first step, we implement a hypermedia system which helps its user by answering the question if it is allowed to build a special house on a piece of land. The system shows its user the development plan and enables him to access the relevant laws. Additionally, each part of the drawing may be explained.

The second step extends the hypermedia system from step one by expert system techniques. This results in an active support from the system: the system leads its user to the relevant information, ask questions and responds depending on the answers. Step two is an important improvement of step one where only the user has an active role.

4. Discussion and Open Questions

The integration of expert systems with hypermedia techniques (HyperXPS) enables the creation of systems, which were not possible with only one of the basic techniques. Our integrated system allows to reason about the hypermedia structure. The expert system "knows" the hypertext. This is more than a mere linking of an expert system to an hypertext.

Our linking of a hypertext to a semantic network wipes out the sharp boundary between context-dependent and context-independent knowledge. This is a first step to overcome the methodological deficit described in chapter 1.

An integrated HyperXPS requires additional tools for knowledge acquisition: We need editors for multimedia information, which are available within HyperCAKE.

The integration leads to an extension of the language for the model of cooperation: The parallel playing of sound and video needs to be expressed. We just started thinking about the neccessary extension. All together we think that the model of cooperation will come to the foreground.

HyperCAKE is a step towards a multi-user knowledge acquisition environment. The used database system supports the work of different users on one knowledge base by transaction management and access rights.

An important point for further development of a knowledge acquisition methodology appears to be, at least for us, to reach a stronger integration of rapid prototyping and model-based approaches. The development of operational languages for conceptual models (cf. [Wetter90], [AngeleEtAl91], [KarbachEtAl91]) works in that direction. An open question remains: Can we define object-oriented languages for conceptual models as an alternative to the logic-oriented mechanisms and is this attempt worth the work? If we use object-oriented languages, knowledge engineering can more easily be linked with the mainstream of conventional software engineering.

5. State of Realization

The typed HAM and the rule interpreter are fully implemented as a stand-alone system. The mapping of the HAM onto the object-oriented database system GemStone will be finished in July 1992. Editors for text, graphics, and audio nodes are already implemented. First application-specific browser and query mechanisms are available. The development of an editor for KADS-Models just started.

We will finish the development of HyDi in spring 1992. The WisGIS system is fully implemented and linked to ArcINFO. The IntPlan-Project is in the conception phase.

The implementation language is Smalltalk-80 Release 4.0. Therefore the system, with the exception of the platform-dependent audio/video interface, is available on Unix-Workstations (Sun, HP/Apollo, IBM RS 6000, etc.), Apple MacIntosh and 80386-PC with Windows.

Acknowledgements

I thank Prof. M. M. Richter who supported me by this work with important suggestions and lots of advices. I am indebted to my colleagues K. D. Althoff, J. Paulokat and S. Weß, who spent their time for the correction of this work.

References

[AngeleEtAl91]
J. Angele, D. Fensel, D. Landes, R.Studer: KARL: An Executable Language for the Conceptual Model, Proc. KAW 1991

[Angstmann91]
R. Angstmann: Multi-Media als Benutzerschnittstelle zu wissensbasierten Anwendungen, Proc. Hypertext/Hypermedia 91, Springer-Verlag

[CampbellGoodmann88]
Brad Campbell, Joseph M. Goodman: HAM: A General Purpose Hypertext Abstract Machine, Communications of the ACM, July 1988, Vol. 31, No. 7

[Forgy82]
C.L.Forgy: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem (Artificial Intelligence 19 (1982) 17-37)

[GainesLinster90]
B. Gaines, M. Linster: Development of Second Generation Knowledge Acquisition Systems, Proc. EKAW 90

[Halasz88]
F. G. Halasz: Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems, Communications of the ACM, July 1988, Vol. 31, No. 7

[Hemker91]
H. Hemker: Erweiterung eines geographischen Informationssystems durch eine

wissensbasierte Inferenzmaschine, Diplomarbeit Universität Kaiserslautern 1992 (in preparation)

[Jäckel91]
Th. Jäckel: Entwicklung einer Shell zur Erstellung eines Expertensystems zur Vorhersage der Bodenqualität, Diplomarbeit Universität Kaiserslautern 1991 (in preparation)

[KarbachEtAl91]
W. Karbach, A. Voß, R. Schuckey, U.Drouven: MODEL-K: Prototyping at the Knowledge Level, Arbeitspapiere der GMD 566, August 1991

[KarbachLinster90]
W. Karbach, M. Linster: Wissensakquisition für Expertensysteme, Hanser-Verlag

[Maurer91]
F. Maurer: CAKE: Computer Aided Knowledge Engineering, Proc. Workshop "Software Engineering for Knowledge Based Systems" IJCAI 91, Sydney

[Maurer91b]
F. Maurer: Knowledge Base Maintenance and Consistency Checking in MOLTKE, Submitted to EKAW 92

[Richter91]
MOLTKE - Methoden zur Fehlerdiagnsoe technischer Systeme, Springer Verlag, (in preparation)

[Wetter90]
Th. Wetter: First Order Logic Foundations of the KADS Conceptual Model, in: B. Wielinga, J. Boose, B. Gaines, G. Schreiber, M. Van Someren: Current Trends in Knowledge Acquisition, IOS Press 1990, Amsterdam

[WielingaEtAl91]
B.Wielinga, A Th. Schreiber, J.Breuker: KADS: A Modelling Approach to Knowledge Engineering, KADS-II/T1.1/PP/UvA/008/1.0, Esprit Project P 5248 KADS-II, 1991