

TheMPO: A Knowledge-Based System for Therapy Planning in Pediatric Oncology^{*}

(Appeared in: *Computers in Biology and Medicine*, vol. 27(3), 1997:177-200)

R. Müller

Institut für Informatik, Universität Leipzig, 04109 Leipzig, Germany

M. Sergl, U. Nauerth, D. Schoppe, K. Pommerening

*Institut für Medizinische Statistik und Dokumentation, Universitätskliniken Mainz,
55101 Mainz, Germany*

H.-M. Dittrich

Kinderklinik, Universitätskliniken Mainz, 55101 Mainz, Germany

Abstract

This article describes the knowledge-based system THEMPO (**T**herapy **M**anagement in **P**ediatric **O**ncology), which supports protocol-directed therapy planning and configuration in pediatric oncology. THEMPO provides a semantic network controlled by graph-grammars to cover the different types of knowledge relevant in the domain, and offers a suite of acquisition tools for knowledge base authoring. Medical problem solvers, operating on the oncological network, reason about adequate therapeutic and diagnostic time-tables for a patient. Furthermore, a corresponding patient record, also based on semantic networks and graph-grammars, has been implemented to represent the course of therapy of an oncological patient.

Keywords: Knowledge-Based Systems, Therapy Planning, Electronic Patient Record, Protocol-Directed Care

^{*}The THEMPO-project is partially funded by the STIFTUNG INNOVATION E.V. of the Ministerium für Wirtschaft und Verkehr of the state Rhineland-Palatinate.

1 Introduction

The treatment of childhood cancer is characterized by the application of standardized *protocols* containing diagnostic criteria, detailed instructions about chemo- and radiotherapy, and recommendations concerning the management of toxicity (e.g., [1]). Due to these protocols, the prognosis of malignant childhood diseases has improved significantly during the last two decades [2,3]. However, this has gone hand in hand with an increasing complexity of treatment, and has led to new challenges and burdens for the medical staff, especially with regard to medical decision making and documentation. As a former program, restricted to the subtask of chemotherapy calculation in this domain (CATIPO [4]), has shown encouraging results in the clinical routine, it has been assumed that a larger knowledge-based system, providing a formal representation of the protocols as its core and therefore being able to support protocol-directed therapy and documentation, could be helpful for the medical staff involved in therapy. Therefore, the Department for Medical Statistics and Documentation of the University Hospital Mainz, Germany, has developed the knowledge-based system THEMPO¹ (**T**herapy **M**anagement in **P**ediatric **O**ncology), providing a) graphical tools for the acquisition of protocol knowledge, and b) problem-solving modules which determine an appropriate (sub)protocol for a given patient, and which configure and modify this protocol according to the patient's initial condition and his individual response during long-term treatment.

First of all, in the remaining part of this section, the structure of pediatric cancer therapy is analyzed. In section 2, three acquisition tools are introduced, which enable the physician to specify protocol knowledge, including a) timetables for chemo- and radiotherapy, b) therapy branches and branching points, and c) rule knowledge for therapy modification. The internal representation approach, which is based on semantic networks realized through attributed, directed and labeled graphs and graph-grammars, is described in section 3. Section 4 outlines the medical problem solving process which uses the protocol knowledge and the patient's data to suggest patient-adapted therapy timetables. The corresponding time- and context-oriented patient database, which is based on attributed graphs and graph-grammars too, is described in section 5. After having addressed implementation details in section 6, an overview concerning related work is given in section 7. Strengths and weaknesses of the current approach together with future research topics in the context of the THEMPO-project are discussed in section 8. As an initial remark, it has to be stated, that - for better presentation - some of the domain examples originating from protocols have been slightly simplified or modified (especially in section

¹The THEMPO-page can be found at
http://www.Uni-Mainz.DE/FB/Medizin/IMSD/TheMPO/TheMPO_eng.html

2 and 4), to clarify the knowledge representation and processing done by the system. Therefore, only the original protocols should be used as knowledge source for the domain of pediatric oncology.

1.1 Protocol Structure in Pediatric Oncology

In pediatric oncology, protocols typically cover time periods from 6 months up to 2 years, including initial diagnostics and assignment to different risk groups and therapy branches, intensive inpatient chemo- and radiotherapy phases for tumor remission, consolidation therapy, and outpatient long-term therapy to avoid tumor recurrence. To analyze whether a new drug or drug combination is able to improve the prognosis, protocols often randomize patients of a particular risk group and assign them to different, competitive therapy branches. A typical chemotherapy may last from 4 days (chemotherapy N1 of the neuroblastoma protocol) up to 9 weeks (chemotherapy I of the acute lymphoblastic leukemia protocol ALL-BFM95), and may consist of 3 cytostatic drugs (N1) up to 10 drugs (HR-2 of ALL-BFM95). Depending on the initial prognosis, intensive chemotherapy units may be administered up to 4 or 8 times sequentially, alternating with phases of recovering. Figure 1 illustrates the structure of a typical protocol, and figure 2 describes the internal structure of a chemotherapy.

Furthermore, a knowledge-based system in this domain has to deal with the following types of temporal constraints and relationships:

- **Exact quantitative relations:** Folin acid rescue should be given exactly 42, 48 and 54 hours after the METHOTREXATE-infusion has started.
- **Inexact quantitative relations:** Chemotherapy of type III (of brain tumor protocol) should start not earlier than 2 weeks after the start of the preceding chemotherapy of type II, but not later than 3 weeks.
- **Qualitative relations:** Electrocardiography and echocardiography should be done before the first and third administration of ADRIAMYCIN.

In addition to this time-table knowledge valid for a patient collective, protocols also contain rule knowledge supplementing or overwriting the standard time-tables, such as:

IF *initial CNS-infiltration has been observed*
THEN *give MTX intrathecal additionally at day 18 and 27 (and not only at day 1,12,33,45,59).*

In summary, a knowledge-based system in the domain of pediatric oncology has to be able to represent time-tables of chemo- and radiotherapy, long-term sequences and stratifications of therapy phases, additional rule knowl-

ALL - BFM 95

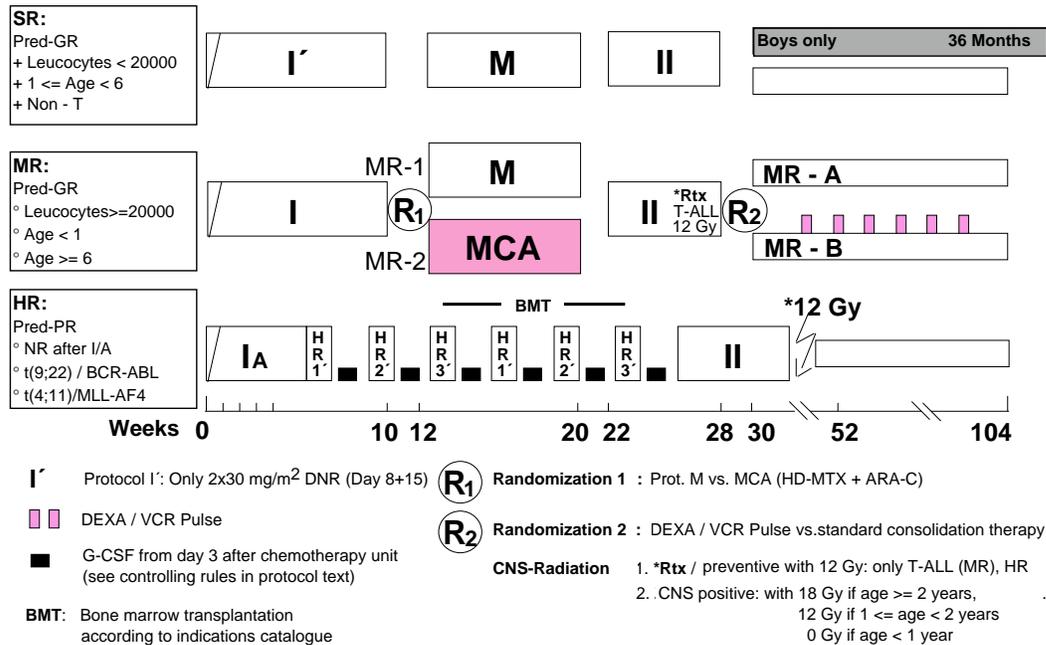


Fig. 1. Temporal structure of the protocol for acute lymphoblastic leukemia (ALL-BFM-95). After the initial diagnosis, the patient is assigned to one of the three risk groups (SR = *standard risk*, MR = *medium risk*, HR = *high risk*). In the assignment criteria boxes at the left side, a '+' indicates an AND-, a circle an OR-conjunction of the conditions (*Pred-GR* resp. *Pred-PR* means *Prednisone Good Response* resp. ...-*Poor Response*). *t(9;22)/BCR-ABL* ... in the HR-box are abbreviations for specific genetic findings indicating critical leukemia subtypes. I', I, M, MCA, I_A and HRx' are types of chemotherapy, the long bars at the right symbolize types of long-term chemotherapy for recurrence avoidance. Radiation therapy is indicated by 'n Gy'. G-CSF is the *Granulocyte-Colony-Stimulating-Factor*. Reproduction of graphic by kind permission of the central commission of the ALL-BFM-95 protocol (H. Riehm et al.).

edge for patient-adapted therapy modification, and temporal constraints concerning the duration of therapy-intensive and therapy-free intervals. As especially modern chemotherapy is based not only on the drugs themselves and their dosages, but also on the complex and sophisticated temporal interaction structure of these drugs, special attention has to be paid to the adequate representation and processing of temporal knowledge.

2 Knowledge Acquisition

To define the structure of oncological protocols, THEMPO provides three graphical acquisition tools hiding the internal representation language from

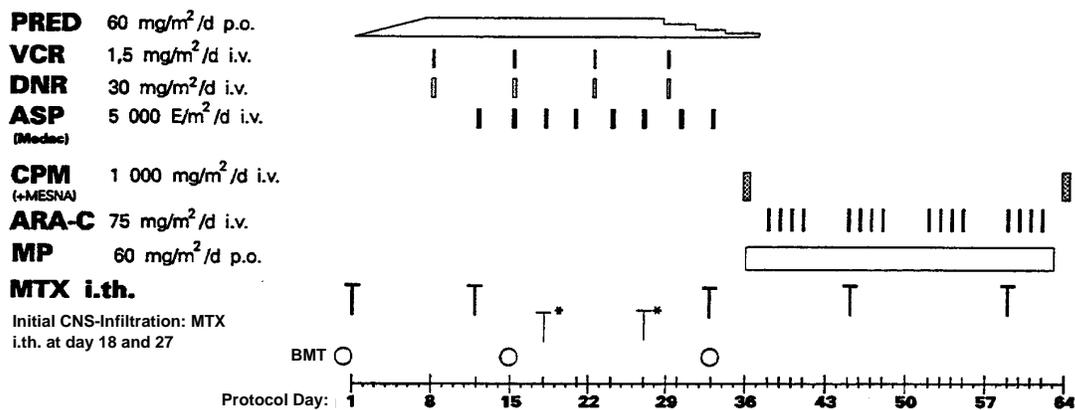


Fig. 2. Time-table of a chemotherapy (type I) of the ALL-BFM95 protocol (the labels PRED, ..., MTX are abbreviations for chemotherapy drugs). The bar representing the time-table of PREDNISON (=PRED) indicates that the dosage of this drug should be increased gradually up to the daily dosage of 60 mg/m² during the first 7 days, and should be reduced during the last 9 days. Reproduction of graphic by kind permission of the central commission of the ALL-BFM-95 protocol (H. Riehm et al.).

the physician. The first tool, called `THERAPYEDITOR`, enables the physician to define cytostatic infusions and supportive medications, and to group these therapy elements together to different types of chemotherapy. Furthermore, phases of radiotherapy can be specified with the `THERAPYEDITOR` as well. With the second tool, called `PROTOCOLEEDITOR`, the physician defines sequences of chemotherapy and radiotherapy phases which have been specified with the `THERAPYEDITOR`. This includes the definition of constraints concerning the intervals between the successive phases, and the specification of stratification (branching) points. The third tool, the `RULEEDITOR`, which can be invoked from both of the above tools, enables the physician to define rule knowledge needed for instance to specify modification or discontinuation criteria valid for a particular chemotherapy, or to define branching criteria for stratification points. All tools internally operate on the same semantic network, each tool having write access to a well-defined subnet. The following three sections briefly illustrate the functionality of these acquisition tools. The structure of the underlying semantic network is described in section 3.

2.1 *TherapyEditor*

This tool is used to specify the basic components of a protocol, including cytostatic medications and types of chemotherapy. Figure 3 illustrates the specification of the internal temporal structure of the chemotherapy N1 within the neuroblastoma protocol.

Protocol: Neuroblastoma Chemotherapy: N1 Select Image... **N1**

Modification Rules

Duration
 Forerun: 0 Day(s)
 Main: 4 Day(s)

Pre-Administration (h) ----->	DDP (Inf.)	VDS (Inf.)	VP16 (Inf.)
	0	1	0
Day 1	<input checked="" type="checkbox"/> 50 %	<input checked="" type="checkbox"/> 100 %	<input checked="" type="checkbox"/> 100 %
Day 2	<input checked="" type="checkbox"/> 100 %	<input type="checkbox"/>	<input checked="" type="checkbox"/> 100 %
Day 3	<input checked="" type="checkbox"/> 100 %	<input type="checkbox"/>	<input checked="" type="checkbox"/> 100 %
Day 4	<input checked="" type="checkbox"/> 100 %	<input type="checkbox"/>	<input checked="" type="checkbox"/> 100 %

Help Undo Cancel Save ↵

Fig. 3. Definition of the time-table of the chemotherapy N1 (neuroblastoma protocol).

With the window shown in this figure, the physician has specified that the duration of this chemotherapy should be exactly 4 days; the infusions DDP (=CISPLATIN) and VP16 (=ETOPOSIDE) are given every day. VDS (=VINDESINE) is given only at the first day. The integer '1' in the 'Pre-Administration'-field of the VDS-column indicates that this drug should be given exactly one hour before the actual beginning of the chemotherapy, as this drug is stopping the cell division at a stage sensitive for the other cytostatic drugs. The entry '100%' specifies that each drug is given with 100 percent of the daily dosage at every day of the administration (with exception of day 1 of the DDP-application, where only 50 percent of the daily dosage are given to accustom the child to this drug). At the upper right, an icon has been assigned to this chemotherapy; this icon will be used as an identifying symbol by the PROTOCOLEditor. Detailed information about the involved drugs (such as the daily dosage or the run-time, or whether a drug should be given intramuscular, oral or as an infusion) can be specified with separate forms.

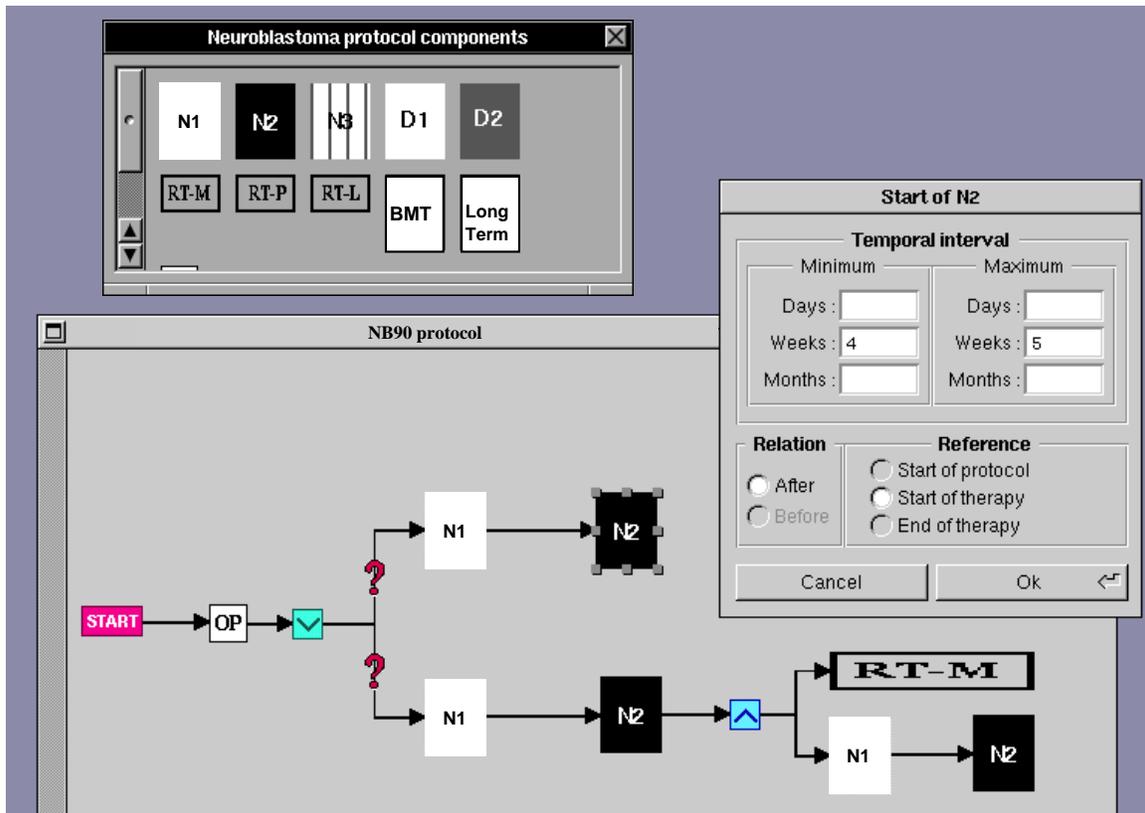


Fig. 4. Graphical definition of a part of the neuroblastoma protocol with the PROTOLEEDITOR (see text for details).

2.2 ProtocolEditor

The PROTOLEEDITOR enables the physician to define sequences of chemotherapy and radiotherapy phases which have been specified with the THERAPYEDITOR described above. This covers the specification of constraints concerning the intervals between the successive phases, and the definition of stratification points specifying that – depending on the patient’s individual data – one of several therapy branches should be selected. The figures 4 and 5 illustrate the interactive process of specifying the temporal structure of a typical protocol.

The palette at the top of figure 4 provides symbols representing the different types of chemo- and radiotherapy which have been defined with the THERAPYEDITOR. This symbols can be dragged to the worksheet beneath to specify the structure of the protocol. Here the physician has specified that, first of all, a surgical phase is applied (symbol OP). A stratification point (symbol ‘V’ for ‘OR’) indicates, that after this surgical unit, the patient is assigned to exactly one of several therapy branches, depending on his individual laboratory values and tumor findings. The question marks indicate that the rule knowl-

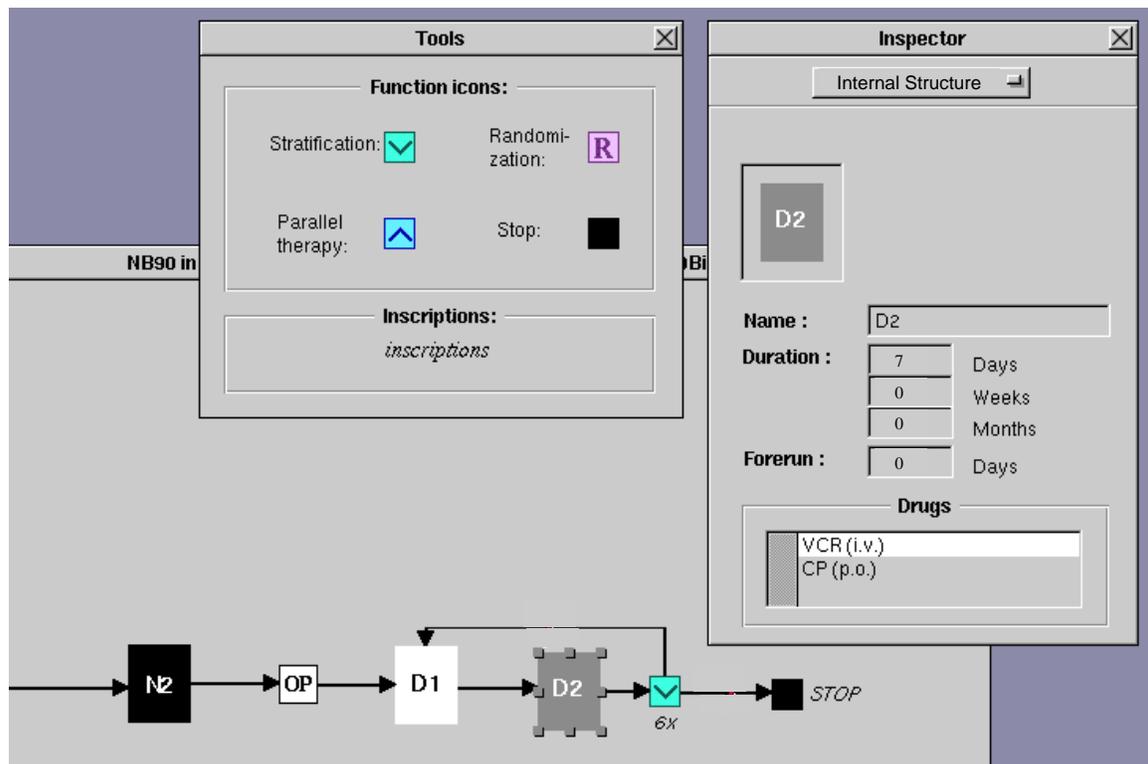


Fig. 5. Graphical definition of a cyclic structure.

edge specifying the stratification criteria has not yet been defined (this can be done with the RULEEDITOR in one of the next steps). Temporal relationships between protocol components are acquired by dragging an edge between two symbols. The user then is prompted to fill a form containing the quantitative information. In figure 4 the physician is specifying that the chemotherapy of type N2 (upper branch) should start not earlier than 4 weeks after the beginning of the preceding N1-therapy, but not later than 5 weeks. The symbol ‘^’ (for ‘AND’) in the bottom branch means that the patient should undergo the N1/N2-sequence and a *parallel* radiotherapy of the metastases (RT-M). Fig. 5 illustrates the definition of a cyclic protocol structure. In this figure, it is defined that the sequence D1/D2 should be given 6 times, and that the therapy then should terminate. Rule-based termination criteria, such as ‘**IF** *Complete remission of tumor* **THEN** *terminate cycle*’, can be defined with the rule editor described in section 2.3. Furthermore, the physician can inspect the components of a chemotherapy by selecting the associated symbol and calling an inspector. In Fig. 5, the inspector is informing the user that the D2 therapy lasts 7 days and consists of the drugs VCR (=VINCRISTINE) and CP (=CYCLOPHOSPHAMIDE).

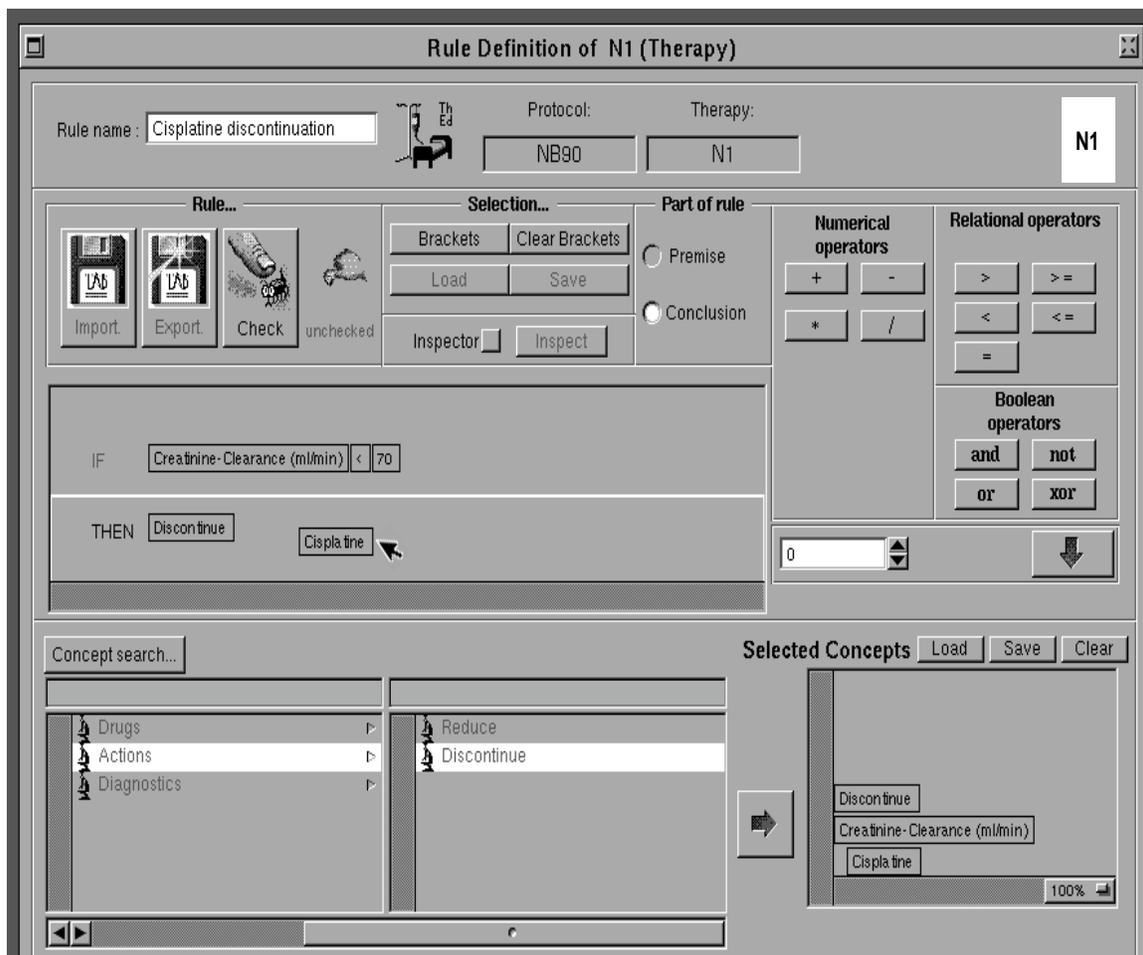


Fig. 6. Graphical definition of a rule with the RULEEDITOR.

2.3 RuleEditor

The RULEEDITOR is used to define rule knowledge. It covers the definition of local modification or discontinuation criteria valid in the context of a particular chemotherapy, and the specification of initial stratification points and termination criteria in therapy cycles. The RULEEDITOR can be called both from the THERAPYEDITOR and the PROTOCOLEEDITOR, and exports the specified rule to the calling module, which then stores the rule in the appropriate part of the internal semantic network (see section 3 for more details). Figure 6 shows the specification of a rule valid in the context of the chemotherapy N1.

In this figure, the rule is defined that whenever the creatinine-clearance value is less than 70 ml/min during a chemotherapy N1, the drug CISPLATIN should be discontinued (as nephrotoxicity – indicated by abnormal creatinine-clearance values – is one of the serious side effects of CISPLATIN). This rule is specified by a) selecting the appropriate concepts (such as drug names, or laboratory parameters) in a concept browser at the bottom of the window, and b) by

dragging the selected concepts to a worksheet above where they are used to assemble the rule. The rule then is subjected to a syntactic check and transformed to the internal rule notation. The calling editor, which in this case has been the THERAPYEDITOR, imports the rule, and stores it in the subnet representing the N1-chemotherapy.

3 Protocol Representation

For protocol representation, a semantic network approach has been chosen. In THEMPO, this approach is based on labeled, directed, and attributed graphs². A protocol component such as a cytostatic infusion, a chemotherapy, or a surgical unit is represented through a labeled node (e.g., with the label *Infusion*); information such as the names of the cytostatic and supportive ingredients, or the daily dosage of a cytostatic component, is stored in attributes associated with the particular label. From the object-oriented point of view, the node labels and their associated attributes form a class hierarchy, the instances of the particular classes being the nodes of the semantic network. Relationships between protocol components, such as temporal or *component-of* relations, are expressed via labeled and directed edges connecting the nodes in the network. Figure 7 shows the inheritance hierarchy of the attributed node labels, and the figures 8 and 9 illustrate the internal representation of a neuroblastoma protocol.

One basic problem arising in connection with complex network-oriented data structures is the formal description of the class of legal networks. A *next-edge* may point from a *TherapyPhase*-node to another, but it is inadequate to connect a *Section*-node with a *RuleSet*-node. To formally define the class of legal protocol networks, a *graph-grammar* approach has been used. Graph-grammars are a calculus to formally describe the legal manipulations on graphs by defining a finite set of so-called *graph productions*. This technique, which has its roots in the graph-theory and software-engineering community [5–7], has already been used to specify the semantics of programming languages [8], to build tools for computer-aided software engineering [7], and, in the domain of medicine, for the modeling of medical dilemmas and decisions [9,10]. In the following the theoretical background of the graph-grammar approach is briefly described; a detailed description of this calculus can be found in [11,12].

²In the following, the terms *semantic network* and *graph* are used, in a way, as synonyms. However, the term *semantic network* emphasizes the representational point of view, the term *graph* focuses on software-engineering aspects.

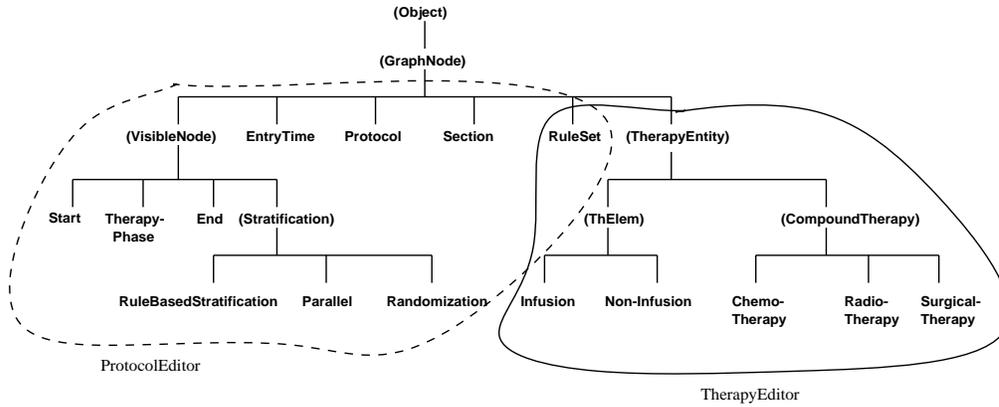


Fig. 7. Inheritance hierarchy of attributed node labels. Every node in the protocol network is an instance of one of the classes in this hierarchy. Abstract classes which can't be instantiated directly are denoted by brackets. The dashed line indicates the classes used by the PROTOLEEDITOR, the other line shows the scope of the THERAPYEDITOR. All classes derived from the abstract class *VisibleNode* denote objects which are visualized as graphical symbols on the worksheet of the PROTOLEEDITOR (see figures 4 and 5).

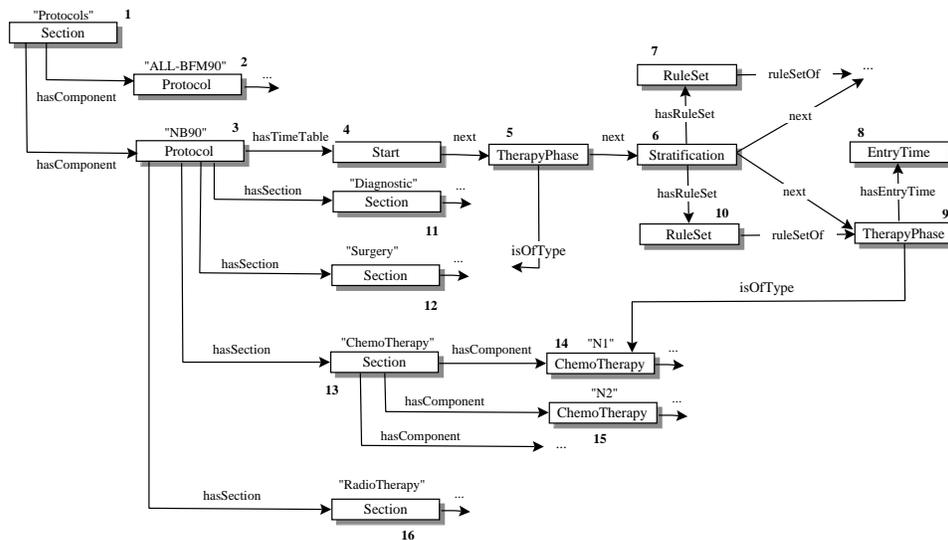


Fig. 8. Part of the semantic network representing the neuroblastoma protocol. The global temporal structure is represented by a subnet starting at node 4. The first therapy phase is a surgical intervention (indicated by the *isOfType*-edge leading to the surgical subnet which contains the surgical components; this subnet has been omitted here to save space). After that the patient is assigned to one of several therapy branches depending on his individual data. The stratification criteria are stored in RULESET-nodes; one node for each branch.

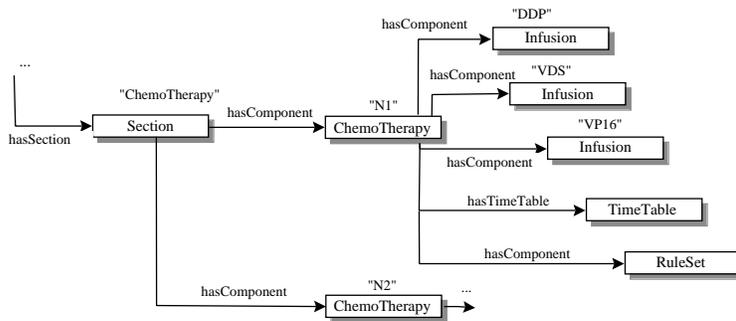


Fig. 9. Part of the semantic network representing the structure of the chemotherapy N1 of the neuroblastoma protocol. The *TimeTable*-node represents the time-matrix specified in figure 3 in section 2, and the *RuleSet*-node stores rules – such as the CISPLATIN-rule in figure 6 – dealing with the modification of a therapy when side effects and toxicity events are detected. An *Infusion*-node stores all information about a particular infusion, such as the daily dosage and the age dependencies, the solvent configuration and the run-time.

3.1 Graph-Grammars

The basic construct of a graph-grammar is the so-called *graph production*, describing a valid unit of graph manipulation. A production P again is a graph providing substructures for the *deletion*, *generation*, and *embedding* of nodes and edges. Figure 10 shows a simple graph operation of the protocol grammar.

This production would be interpreted as follows:

- The parts *Del* (for the *Deletion* of nodes and edges) and *Det* (for *Determined Embedding*) specify where the production should take place in the current host graph (i.e. the protocol graph). In this example, the existence of a *VisibleNode*-node (or a node with a sublabel of *VisibleNode* – such as *TherapyPhase*), and an incoming *marks*-edge with a *Cursor*-node as its source node, has to be determined. The *Cursor*-node is used to achieve the unambiguity of a matching region in a host graph, and is usually set by the preceding graph production to indicate where the next production should take place. For instance, a *Cursor*-node may point to the node representing the chemotherapy currently being edited by the THERAPYEDITOR. Furthermore, to avoid the problem of several matching regions in the host graph, productions can be provided with additional constrains for the starting region. In so-called *preambles* it is possible to specify that some node attributes of the matching region must fulfill specified conditions, or P will not be processed.
- If the structure specified by *Del* and *Det* does not exist, the production is aborted. Otherwise, the following steps have to be done:
 - Delete the structure specified in *Del*. In the case of Figure 10, remove the *Cursor*-node and the outgoing *marks*-edge.

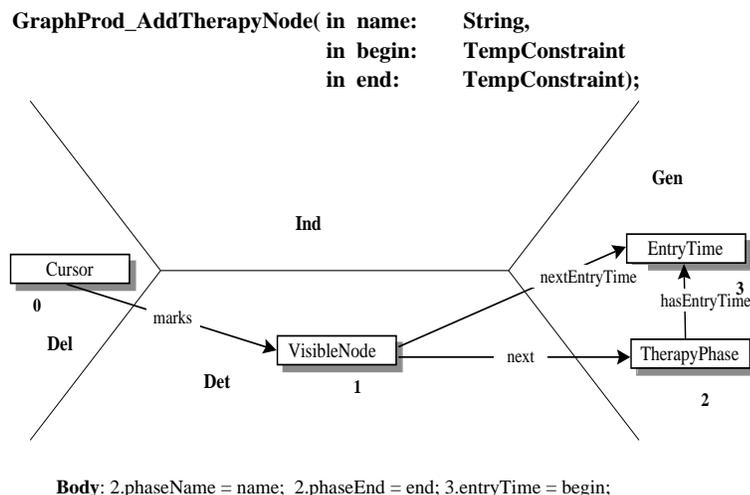


Fig. 10. Sample graph production of the protocol graph grammar (with *indeterminate embedding* = \emptyset), which is launched by the THERAPYEDITOR and adds a new therapy phase node to the oncological network. The string at the top (**GraphProd...**) is the head of the production with the input parameters *name*, *begin* and *end*; **TempConstraint** is a structure to represent the temporal constraints affecting the begin or end of a phase (the little window ‘Start of N2’ in figure 4 contains an example for temporal constraints). The values of the input parameters then are assigned to the attributes of the nodes in the **Body**-statement (see text for more details). For a more complex graph production with indeterminate embedding $\neq \emptyset$, see figure 13 in section 5.

- Process the *Gen*-part of the production (for *Generation* of nodes and edges). In this case, create a *TherapyPhase*-node and an *EntryTime*-node, and connect them with a *hasEntryTime*-edge (in the specified direction).
- After that, embed the new nodes into the existing host graph. This embedding is processed in two steps. The first step is described by the substructure called *Det* (for *Determined Embedding*), and denotes an embedding which is carried out anyway (therefore *determined*), as the substructure *Det* does exist in the graph (otherwise, the production would not have been launched). In this example, the determined embedding consists of simply connecting the *TherapyPhase*-node with the *VisibleNode*-node through a *next*-edge, and of linking the *EntryTime*-node to the *VisibleNode*-node via a *nextEntryTime*-edge.
- The second part (*Ind* for *Indeterminate Embedding*) describes an embedding which depends on the current structure of the surrounding host graph, and therefore is not known in advance. In this simple example, the *Indeterminate Embedding* is empty. Figure 13 in section 5 provides a more

- complex production with indeterminate embedding $\neq \emptyset$.
- The last step of this production processes the attribution commands in the **Body-Statement** (e.g., the assignment of the value of the parameter *begin* to the *entryTime*-attribute of node 3 (the new *EntryTime*-node)).

More generally and formally, and given a set of edge labels E and attributed node labels N , a graph-grammar is a triple (S, P, T) , where

- S is a *start graph*,
- P a finite set of *graph productions*, and
- T a finite set of *graph transactions*.

A start graph S is a graph built from elements of N and E , the attributes of all nodes of S having defined values. The start graph is generated when a new protocol is defined with the PROTOLEEDITOR, and provides a tree to organize and group the protocol components and time-tables. Graph *productions* define the valid basic manipulations on a graph and have the structure shown in figure 10. Furthermore, graph productions can be grouped together to *graph transactions*. A graph transaction is an indivisible sequence of graph productions which has to be carried out entirely or – if an error occurs during the processing of the production rules – has to reset the graph to the state before the transaction was launched. By using transactions inconsistent states of the protocol graph can be avoided.

Formally, a protocol graph is valid if it can be derived from the start graph S by using transactions only of the protocol graph-grammar. In practice, the validness of the protocol network and especially the referential integrity is enforced by allowing external modules only to call public transactions of the grammar. The graph-grammar approach introduced and illustrated above has been used to model and develop the oncological knowledge base of THEMPO. All together, the recent protocol graph-grammar consists of 36 generic graph productions and 30 graph transactions which can be parameterized with node and edge labels and attribute-values.

In conclusion, it can be stated that graph-grammars are an appropriate calculus to formally describe complex graph manipulations which have to be processed during protocol acquisition. Entity-Relationship models, for instance, which also specify the valid structure of graphs, don't provide a formalism to describe the *dynamics* of network transformation processes. The graph-grammar approach introduced has been used to describe the structure and the dynamical behavior of the protocol graph formally on a high level of abstraction without having to take care of implementation details. With this approach conceptual errors can be avoided or detected at an early stage. The graph-grammar model for instance only claims that edges between nodes are represented *explicitly*; it does not make any assumptions concerning the ques-

tion how these edges are realized in detail (e.g., through *adjacency-matrices* or *adjacency-lists*). This is an implementation issue (see section 6) as well as the realization of graph transactions with the low-level transaction mechanism of a database.

3.2 Rule Representation

Rules which have been specified with the RULEEDITOR described in section 2 are internally stored in CLIPS, which is a rule-based language with forward-chaining inference [13]. For each *ChemoTherapy*-node in the protocol network, there exists a *RuleSet*-node which is connected with the *ChemoTherapy*-node via a *hasComponent*-edge, and which stores rules valid only in the context of this particular chemotherapy (see figure 9 as well). If a patient passes through the chemotherapy represented by this *ChemoTherapy*-node, the relevant rules can easily be retrieved by accessing this associated *RuleSet*-node.

4 Medical Problem Solving

To model the oncological problem solving process, a modification of the method *Episodic-Skeletal-Plan-Refinement*, based on the works of [14], has been used. This problem-solving method consists of the three subtasks *Propose-and-Calculate-Plan*, *Identify-Problem* and *Revise-Plan*. These subtasks will now be described briefly.

4.1 Proposing and Calculating a Plan

When a pediatric patient is admitted to hospital after a cancer disease has been diagnosed or a therapy-free interval has finished, the *propose-plan* module reasons about the appropriate therapy to administer next. This is done by exploring the semantic network described in section 3. If, for example, the patient has recently finished a chemotherapy of type N1 and is then recovering in a therapy-free interval, the problem-solver explores outgoing *next*-edges to identify the next therapy phase, processes the temporal constraints in *Entry-Time*-nodes to suggest the time for starting the next phase, and loads and processes rules stored in *RuleSet*-nodes to determine whether the patient fulfills all criteria for starting the therapy phase. This problem solving process can explicitly be launched by the physician for a given patient, or can be done through a demon process scanning the patient database (preferably at night) to determine which patients have to be prepared for which type of therapy.

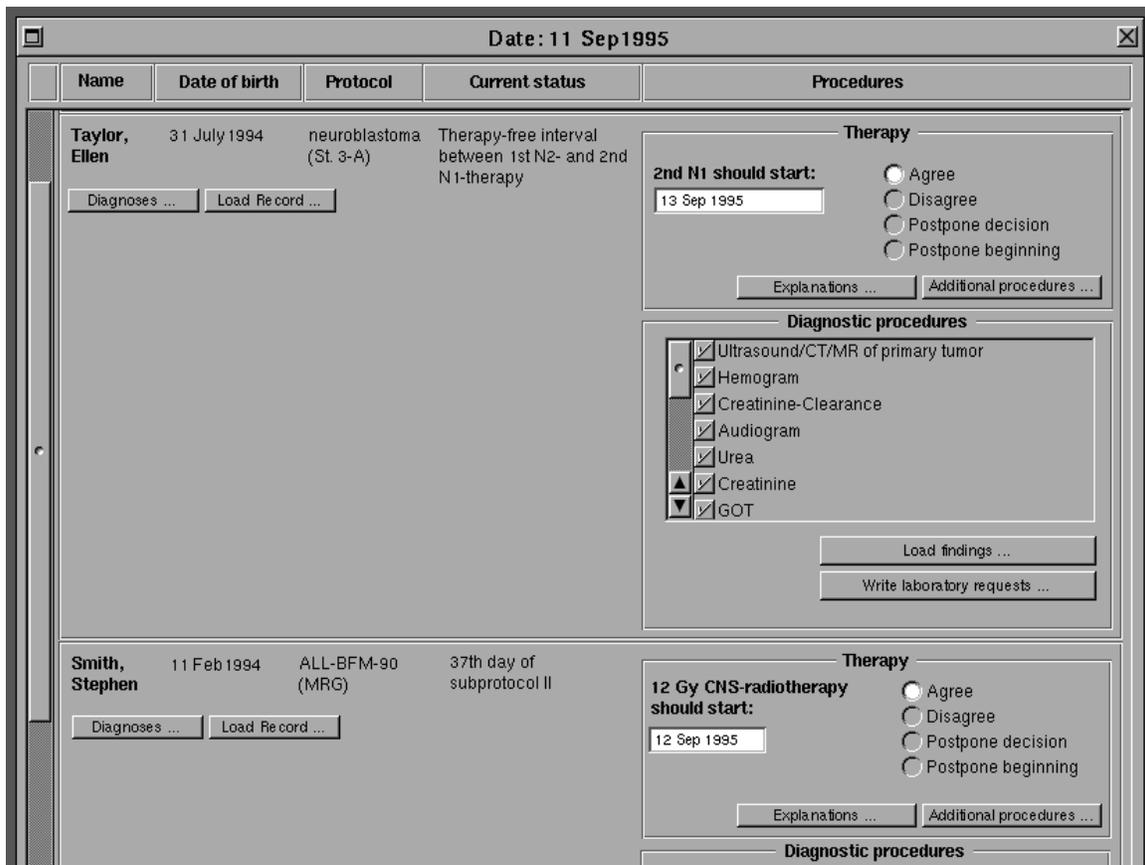


Fig. 11. Suggestions of the medical problem solver for two (fictitious) patients.

Figure 11 shows the suggestions of the problem solver for two patients in the database.

For the first patient in this figure, who is currently recovering from the 1st chemotherapy of type N2, the problem solver for example has suggested to start the next chemotherapy (type N1) on 13 Sep 1995. The physician can agree, disagree, or postpone the decision. If the button entitled with *Explanations ...* is clicked, the problem solver a) invokes the PROTOCOLEditor to load and view the relevant part of the protocol, and b) loads the individual patient data which have led to the suggestion. In the box underneath, the physician can inspect the diagnostic procedures which have to be done before the patient enters the chemotherapy, and can add or remove procedures. If the physician agrees to the suggestion of the problem solver (e.g., agrees that the next chemotherapy should be of type N1 and should start at the suggested date), a submodule is launched to calculate the patient-adapted daily dosages of the drugs of the chemotherapy. The knowledge about the components of a chemotherapy, its time-table and the daily dosages is retrieved from the subnet shown in figure 9. Furthermore, portioning and event plans can be calculated for the nursing staff, giving the information at which time the infusions have to be administered, and which infusion rates have to be tuned in.

To achieve transparency of the underlying medical reasoning, the problem-solving process has been explicitly described with a CML-oriented language. CML (= Conceptual Modeling Language) [15] is a semi-formal specification language to describe the expertise model of knowledge-based systems, and has been developed within the COMMONKADS-research project (e.g., [16]). CML provides constructs for the specification of ontologies, basic inference steps, and complex tasks. The following example shows the specification of the task *T_ProcessRuleBasedAssignment*, which is a subtask of the *propose-plan*-module, and which determines an appropriate risk group for a particular patient (the prefix *T* denotes a *task*, the prefix *I* an *inference*):

task T_ProcessRuleBasedAssignment;

task-definition:

goal: *derive appropriate (risk-group) assignment*

input:

patient-id: *a variable storing the patient identifier*

protocol-id: *identifier of protocol the patient is treated with*

assign-id:

assignment identifier within the protocol

(a protocol may have more than one assignment)

output:

assignment-derived:

the assignment which has been derived (e.g., MR=medium risk group)

(is NIL if assignment has not been possible)

record-access-results:

a list containing information for example about

missing or uncertain patient data in the record

spec:

This task tries to derive an appropriate assignment for the patient identified by *patient-id* and currently undergoing the protocol *protocol-id*. Therefore, the assignment rules stored in the protocol subnet identified by *assign-id* are processed, and relevant information about missing or uncertain data, or data which do not exactly match the temporal requirements of the rules, is stored in the list *record-access-results*.

task-body:

sub-tasks: ... *(list of the subtasks used)*

additional-roles:

assign-rules: *list of assignment rules to be processed*

tmp-rule-id: *identifier of current rule*

tmp-result: *high-level result of current rule processing (e.g., PREMISE_FALSE)*

tmp-record-access-results:

detailed information about missing or uncertain

data concerning the current rule evaluation

control-structure:

T_ProcessRuleBasedAssignment(patient-id, protocol-id, assign-id

– > assignment-derived, record-access-results) :=

assignment-derived := NIL;

```

L_GetAssignRuleIds(protocol-id, assign-id - > assign-rules)
  // get assignment rules from knowledge base
FOR tmp-rule-id IN assign-rules DO // for each rule ...
  T_ProcessRule(patient-id, protocol-id, assign-id, tmp-rule-id
  - > tmp-result, assignment-derived, tmp-record-access-results)
  IF (tmp-result == PREMISE_TRUE) THEN // rule fired ...
    // overwrite former, irrelevant record access results,
    // as assignment has been successful, and exit task ...
    record-access-results := tmp-record-access-results;
    EXIT-FOR-LOOP
  ELSE // PREMISE_FALSE or UNDECIDABLE
    // add information about data responsible for PREMISE_FALSE,
    // or missing data (in case of UNDECIDABLE) to list record-access-results
    L_AddTmpResults(record-access-results, tmp-record-access-results
    - > record-access-results)
  ENDIF
NEXT tmp-rule-id; // next rule ...

```

The task *T_ProcessRule* is the central subtask invoked from *T_ProcessRuleBasedAssignment*. This subtask does not only process the assignment rule itself by communicating with the internal CLIPS-interpreter, but collects and maintains all relevant information concerning the patient data accessed during rule processing. When *T_ProcessRule* has finished, this information is represented in the output-list *tmp-record-access-results*, which provides a structured entry for every patient parameter accessed during rule processing. To illustrate this, the following typical assignment rule is used as an example:

```

IF risk factor3  $\geq 0.8$  at day 1 (of initial chemotherapy) AND peripheral
leukemia cells  $< 1000$  at day 8 AND blast cells in bone marrow  $< 5$  percent
at day 33 AND ...
THEN assign patient to risk group MRG

```

Depending on the data situation in the patient record, the list *tmp-record-access-results* would provide information about the following rule processing ‘events’:

(i) **Unavailable data:**

For example, *tmp-record-access-results* may store that there is NO information at all about the number of peripheral leukemia cells in the record, and that therefore the rule could not have been applied. Information about missing data may be irrelevant as other assignment rules may have been able to fire, but this information becomes important for the physician, if NONE of the rules has been able to conclude a risk group for the patient. In the latter case, the task invoking *T_ProcessRuleBased-*

³ The risk factor is a parameter measuring the initial number of peripheral blast cells, and the degree of initial hepato- and splenomegaly.

Assignment would process the output list *record-access-results* (which is a collection of the *tmp-record-access-results-lists*), and would, for example, detect that the assignment has failed because of a missing leukemia cells value required by each of the rules. Therefore, an alert would be generated informing the physician that this parameter should be inspected as soon as possible.

(ii) **Temporal mismatch:**

Because of the organizational and medical complexity of the diagnostic and therapeutic procedures, and because of the patient’s medical individuality, a patient parameter – although available in the record – may not exactly match the temporal constraints of the rule, e.g., the blast cells of the bone marrow may have been inspected on day 31 instead of day 33. Therefore, to avoid decisions based on data being too old in the current clinical context, the temporal match is measured in the following way during rule processing:

- Let F_i denote the finding parameter i appearing in the premise of a rule R with n findings (e.g., $F_i =$ ‘blast cells in bone marrow (in percent)’).
- Let t_{F_i} denote the time point for which F_i is requested (e.g., $t_{F_i} =$ day 33). The granularity of time depends on the particular parameter and the specific context of the rule; in the domain of pediatric oncology, days can be viewed as the central time units. However, in particular situations such as the application of the drug METHOTREXATE and the corresponding folin acid rescue, time has to be modeled on the level of hours.
- Let P_i denote the corresponding patient parameter (and its value) in the record, and let t_{P_i} denote the time point of inspection of P_i (e.g., $P_i =$ ‘blast cells in bone marrow = 4 percent’ have been examined on $t_{P_i} =$ ‘day 32’). If there are “competitors” P_i^1, P_i^2 with $|t_{F_i} - t_{P_i^1}| = |t_{F_i} - t_{P_i^2}|$, then the systems usually selects the most recent value P_i^j ($j \in 1, 2$) available in the patient database⁴.
- Let $tol_{F_i, t_{F_i}, C}$ denote the temporal tolerance limit of F_i at t_{F_i} in the current protocol context C . For example, if $t_{F_i} =$ ‘day 33’ and $tol_{F_i, t_{F_i}, C} = 2$ (days), then $t_{P_i} = 30$ would be interpreted in the way that the parameter P_i is too old to be considered valid for rule R . The value of $tol_{F_i, t_{F_i}, C}$ strongly depends on the particular context C ; if, for exam-

⁴ However, there are situations where the older value has to be used. For instance, if a leukemia assignment rule requires a hematological parameter F at day 30, which is the *last* day of an initial chemotherapy, and if there are values of F only available for day 28 and day 32, it may be necessary to consider the older value, because this value does reflect the patient’s hematological situation *during* chemotherapy much better than the one after the chemotherapy, and because the assignment rule uses the patient’s initial chemotherapy response – measured by F and being an important prognostic factor – for risk group assignment. Exceptions like this are explicitly stated in the knowledge base.

ple, F_i is a hematological parameter, $tol_{F_i, t_{F_i}, C}$ usually is significantly smaller during a chemotherapy than during a recovering phase, because of the massive hematological side effects of the chemotherapy. The particular value of $tol_{F_i, t_{F_i}, C}$ is retrieved from the knowledge base during rule processing.

- Then, the degree of temporal matching between F_i and P_i in a context C is measured through:

$$TM(F_i, P_i, C) = \begin{cases} 0 & , \text{ if } |t_{F_i} - t_{P_i}| > tol_{F_i, t_{F_i}, C}, \\ \frac{tol_{F_i, t_{F_i}, C}}{tol_{F_i, t_{F_i}, C} + |t_{F_i} - t_{P_i}|} & \text{ else} \end{cases}$$

This term is 1, if $t_{F_i} = t_{P_i}$, and decreases if the difference $d_i = |t_{F_i} - t_{P_i}|$ between t_{F_i} and t_{P_i} increases. If a parameter has a relatively small temporal tolerance limit, d_i is weighted stronger than if there is a larger tolerance limit.

- If F_{i_1}, \dots, F_{i_j} is the subset of finding parameters F_1, \dots, F_n of rule R for which the corresponding P_{i_1}, \dots, P_{i_j} would fulfill the premise of R , then the temporal match of P_{i_1}, \dots, P_{i_j} is measured through:

$$TM_{R, P_{i_1}, \dots, P_{i_j}} = \prod_{k=i_1}^{i_j} TM(F_k, P_k, C)$$

(iii) **Certainty mismatch:**

Furthermore, if $c_{P_i} \in [0, 1]$ denotes the certainty of the patient parameter P_i , then the temporal/certainty match of P_{i_1}, \dots, P_{i_j} is measured through:

$$TCM_{R, P_{i_1}, \dots, P_{i_j}} = \prod_{k=i_1}^{i_j} TM(F_k, P_k, C) \cdot c_{P_k}$$

Certainty factors occur relatively seldom in the domain of protocol-directed oncology therapy, but there are several documentation slides where the physician can label a finding for example with *certain*, *nearly sure*, *unsure* and *completely uncertain*. Internally, this labels would be represented, for instance, as 1, 0.75, 0.5, 0.25.

If $T_ProcessRule$ processes a rule R with finding parameters F_1, \dots, F_n , it first inspects whether there is a patient data combination P_{i_1}, \dots, P_{i_j} fulfilling the premise of R , and checks whether the match $TCM_{R, P_{i_1}, \dots, P_{i_j}}$ is larger than a given threshold t . If yes, the match is considered to be sufficient, and the rule fires, and the patient is assigned to the risk group in the conclusion of R . If not, $T_ProcessRule$ selects another combination P_{j_1}, \dots, P_{j_m} fulfilling the premise of R from the record, and again calculates TCM . This is done until a patient parameter combination is found which fulfills the premise of R , and where the match TCM is larger than the threshold, or until no subset P_{j_1}, \dots, P_{j_m} fulfilling the premise of R is left. This algorithm is fairly simple, if the F_1, \dots, F_n

are connected via **AND**-operators (as then there is – at best – only one combination fulfilling the premise, i.e. P_1, \dots, P_n), but it becomes more complex when there are **OR**-conjunctions in R . During rule processing, $T_ProcessRule$ maps all information about temporal/certainty mismatches (and unavailable data) to the output list $tmp_record_access_results$ for further inspection.

4.2 Identify-Problem and Revise-Plan

As only a small number of patients passes long-term treatment without significant side-effects, the internal problem-solver also has to identify problems caused by the aggressive therapy, and has to decide whether to reduce or discontinue a drug or drug combination, or whether to administer additional procedures for toxicity management. Therefore, when a chemotherapy is given to a patient, all rules stored in the $RuleSet$ -node associated with this chemotherapy are loaded by the internal CLIPS-interpreter. As different types of chemotherapy are characterized by very different types of side-effects and toxicity, and therefore require very specific therapy modification, this $RuleSet$ -node represents rules which are valid only in the context of this particular chemotherapy, and which have been specified with the graphical $RuleEditor$ of section 2.3. If patient data are entered into the record, the CLIPS-forward chainer then processes the rules to detect abnormal and dangerous situations (e.g., $Creatinine-Clearance (ml/min) < 70$), and suggests therapy modifications such as *Discontinue Cisplatin*.

5 Patient Record

To store a patient’s diagnostic data and course of treatment, THEMPO provides an electronic patient record as a source for retrospective inspection and medical decision making. The main component of the underlying patient model, which is based on semantic networks too, is a temporal subnet for the representation of the course of treatment. This temporal net represents medical events such as the administration of a cytostatic drug through labeled, attributed nodes, while temporal relationships are modeled with labeled edges. Figure 12 for example shows the internal representation of a chemotherapy administered to a particular patient.

The graph in this figure represents the administration of the chemotherapy ‘I’ within the leukemia protocol ‘ALL-BFM-95’ (node labels are given inside the boxes, attribute-values are placed beside the box frames). Days of treatment and phases of chemotherapy are modeled explicitly, and not implicitly through a set of time-stamped data. Medical events such as the administra-

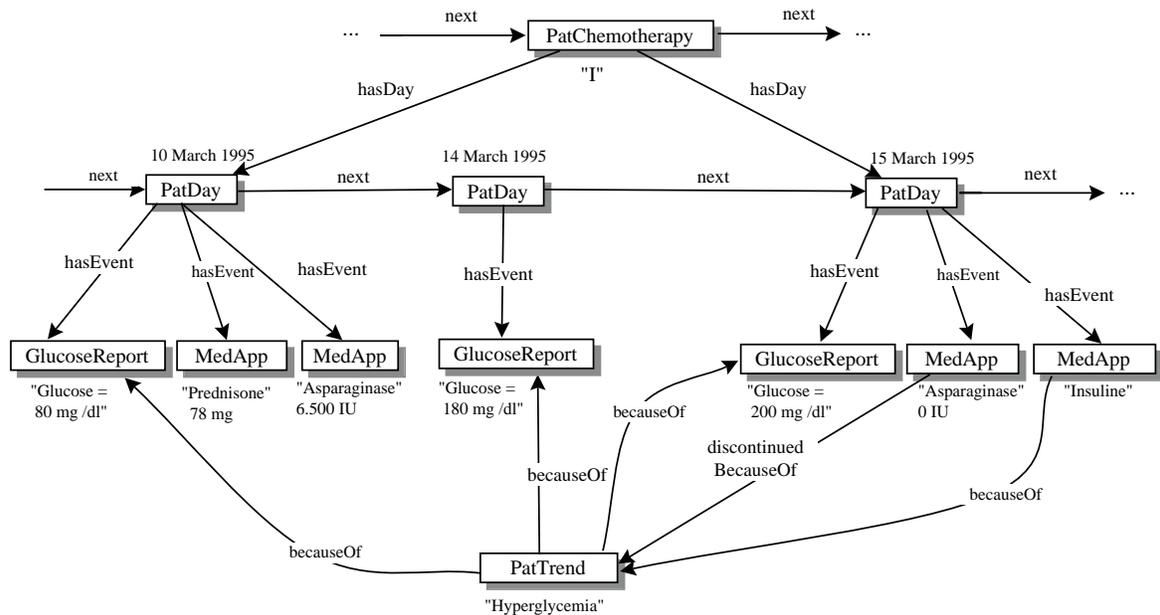


Fig. 12. Part of a patient graph.

tion of a cytostatic drug like PREDNISONE are related to *PatDay*-nodes via *hasEvent*-edges. Furthermore, the subgraph shown in this figure represents the causal context that ASPARAGINASE has been discontinued on 15 March 1995 because of a serious hyperglycemia trend – described by the node labeled with *PatTrend* – which has developed since 14 March 1995. To control the hyperglycemia, Insuline has been administered. The explicit representation of *becauseOf*- or *discontinuedBecauseOf*-relationships enables the record to directly answer questions such as *Why has the obligate drug ASPARAGINASE been discontinued on 15 March 1995?*

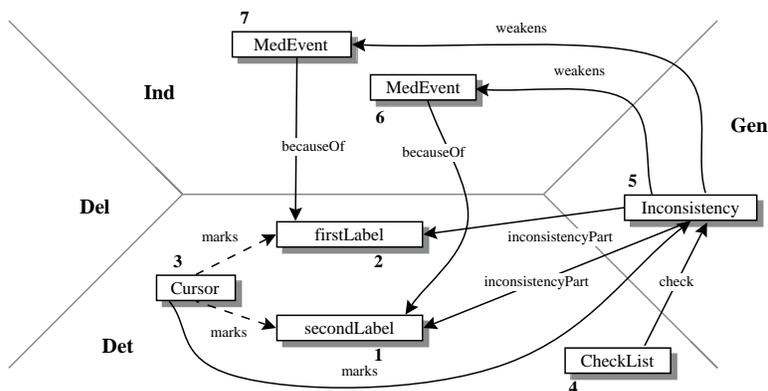
Similar to section 3, a graph-grammar approach has been used to manage the complex graph transformations driven by information entry and medical problem solving. Figure 13 shows a typical graph production of the patient graph-grammar.

This production maps an inconsistency (detected by the physician or the problem solver) between two (diagnostic) data items – such as a new laboratory value being not consistent to the current diagnosis – to the patient graph. The production in this figure is parameterized with: **(1)** the labels of the conflicting nodes (*firstPart* and *secondPart*), **(2)** a string for additional comments (*description*), and **(3)** the (day) time of detecting the inconsistency (*time*). After having found the matching region in the patient graph, the production generates an *Inconsistency*-node (node 5) and connects it with the two conflicting nodes (1 and 2) via *inconsistencyPart*-edges and the global *Check-List*-node (determined embedding). After that, the indeterminate embedding is processed: First of all, it is explored whether node 1 or node 2 is connected

```

GraphProd_EstablishInconsistencyContext( in firstLabel: NodeLabel,
in secondLabel: NodeLabel,
in description: String,
in time: Time);

```



Body: 5.inconsistencyDscript = description; 5.contextKey = sys_contextKey();
5.eventTime = time;

Fig. 13. Sample graph production *GraphProd_EstablishInconsistencyContext* of the patient graph-grammar (see text for more details).

with *MedEvent*-nodes through *becauseOf*-edges. If this is not the case, the production terminates successfully. Otherwise, the production connects every detected *MedEvent*-node with the new *Inconsistency*-node via a *weakens*-edge (if node 6 and 7 in the Ind-field are identical, the *weakens*-edge is drawn only once). The rationale behind this behaviour of the production is the following: Whenever a *MedEvent*-node n (which may, for example, represent a medical conclusion or a therapeutic procedure) is based on one of the nodes 1 or 2, which are now integrated into an *Inconsistency*-context, then n should be marked as a node whose medical validity is weakened by the detected inconsistency. Whenever the physician loads a window containing the information represented by n , then the system will generate a message that there exists a related data context which has been viewed as an inconsistency, and which therefore (possibly) weakens the validity of n .

For the acquisition of semantic relationships such as the *discontinuedBecauseOf*-edge between the ASPARAGINASE-node and the hyperglycemia-node in figure 12, or the inconsistency context in figure 13, the THEMPO-record provides graphical mouse-based tools which enable the physician to specify relationships between medical events interactively. Figure 14 shows the acquisition of a causal context during a chemotherapy N1 of the neuroblastoma protocol.

In this figure the physician specifies that he has reduced the CISPLATIN-infusion on 5 May 1995 because of a decreased creatinine-clearance detected on 4 May 1995. This specification is done by dragging with the mouse from the CISPLATIN-window to the field representing the data responsible for the re-

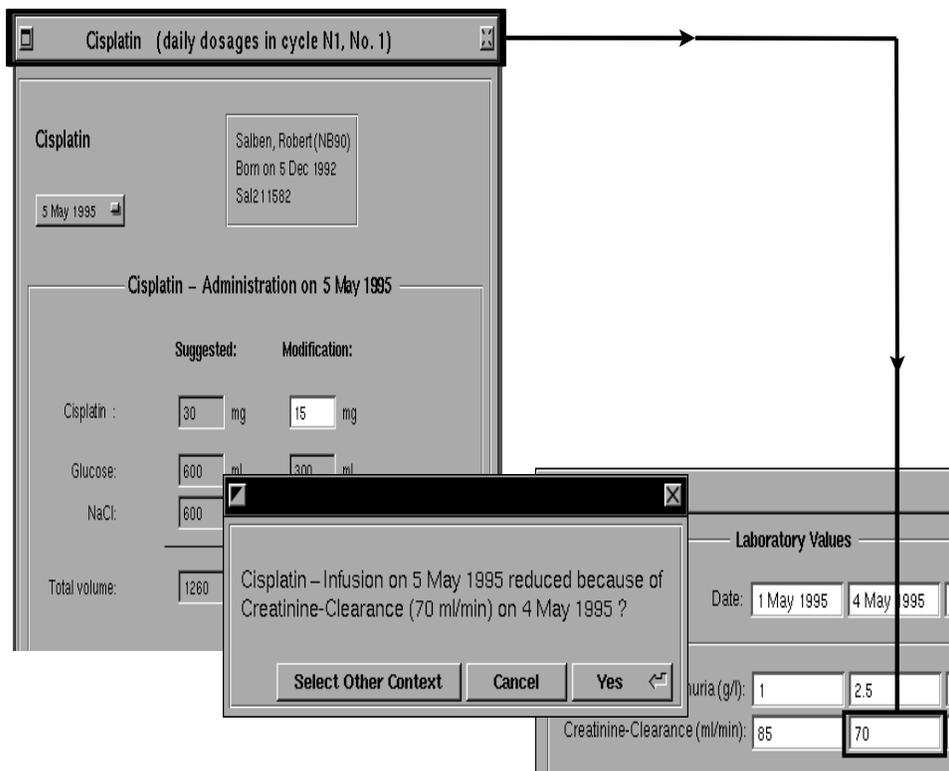


Fig. 14. Screen dump of a graphical context specification (fictitious patient).

duction of the drug dosages (e.g., creatinine-clearance of 70 ml / min). During the drag process a black arrow appears on the screen indicating the connection between the two events. The program, which has registered that the dosages have been reduced (by comparing the two dosage columns), assumes that the physician wants to specify a *reducedBecauseOf*-context because he has connected a drug-window with a window showing laboratory data. If the physician confirms this assumption, the program generates a *reducedBecauseOf*-edge in the patient graph between the CISPLATIN-node and the creatinine-node.

Principally, the patient graph does not only allow the representation of contexts such as causal relationships and inconsistency clusters, but also reflects the main classes of queries occurring in the daily routine (i.e. *interval-oriented* queries and *parameter-oriented* queries). This is done by linking every *MedEvent*-node a) to the *PatDay*-node representing the day of the event (as shown in figure 12) and b) by linking all nodes of a given *MedEvent*-subclass together via *next*-edges to represent the temporal course of a parameter (see figure 15).

Therefore, by using the query-oriented graph structure, querying of the patient graph is - in most of the cases - done in two steps: The query first takes advantage of the graph organization to reduce the search space, and then uses OQL (= OBJECT QUERY LANGUAGE⁵)-commands of the database sys-

⁵OQL is the object-oriented standard query language designed by the OBJECT

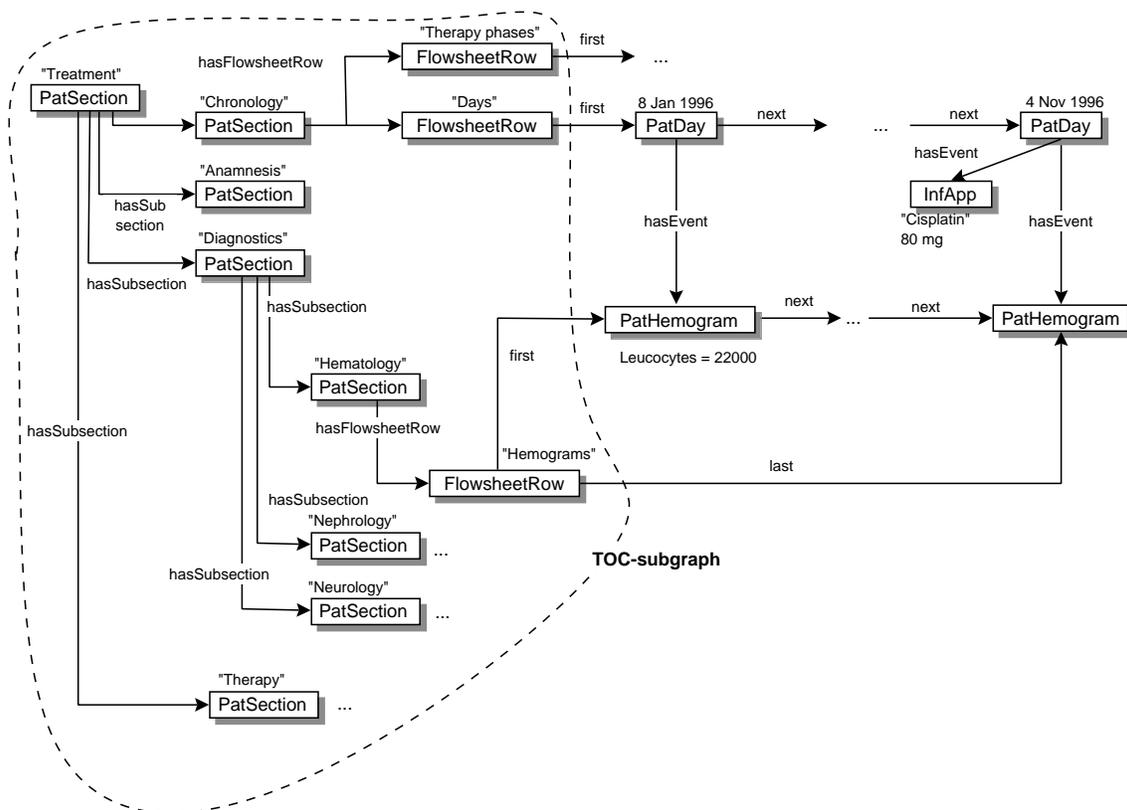


Fig. 15. The subgraph representing the table of contents (TOC) of a patient's record. It is updated dynamically according to the specific procedures applied to the patient, and serves as an entry point for queries to reduce search space (e.g. by jumping directly to the *FlowsheetRow*-node for the hemogram reports when the hemogram values are requested for a longer time-period). Furthermore, the user interface menu of the patient's record is dynamically generated from this TOC.

tem to filter the node-lists in the reduced search space. If, for example, the physician wants to inspect all cytostatic drugs for a given time interval such as a particular chemotherapy, the query algorithm first jumps to the node representing this chemotherapy (by following the *next*-edges connecting the *PatChemotherapy*-nodes), and then explores the subtree consisting of the *hasDay*- and *hasEvent*-edges (by filtering the list of all *MedEvent*-nodes within this chemotherapy with OQL). On the other hand, if for instance all time intervals have to be found where the hemograms of the patient fulfill a specific condition (such as $\text{leucocytes} > 20000$), the query jumps to the first *PatHemogram*-node in the patient record by using a subgraph representing the patient-specific table of contents (see figure 15), and then uses OQL to filter the *PatHemogram*-nodes and to determine the requested intervals. This

DATA MANAGEMENT GROUP and implemented in most of the available object-oriented database systems. It provides query capabilities similar to those of SQL, with additional object-oriented features such as invoking object methods and filtering their return values, and path expressions to navigate through object networks.

approach of additional query-oriented graph structuring has been considered necessary, as the record of a patient usually covers long time periods (from 6 months up to two years), which implies hundreds of instantiations of a particular parameter on the time axis.

A detailed description of the context-oriented THEMPO-record can be found in [17].

6 Implementation Issues

To implement the protocol and patient model, the object-oriented C++-database POET in the NEXTSTEP-environment has been used. To represent nodes and edges, the abstract data types *Graph* and *GraphNode* have been implemented in C++/POET. These data types encapsulate elementary graph-functionality (e.g., adding and removing nodes and edges). For internal graph representation adjacency-*lists* were used instead of adjacency-*matrices* because the question in which contexts a node is embedded arises often. With adjacency-*lists* – separated by incoming and outgoing edges – each node is informed about his ‘next-door’-neighborhood and can quickly answer questions such as ‘Are there any outgoing *next*-edges leaving you?’⁶. Graph productions and graph transactions have been implemented as C++-functions using the low-level transaction-mechanism of POET.

To realize the graphical acquisition tools described in section 2 the APPKIT-OBJECTIVE-C libraries of NEXTSTEP [19] were used allowing comfortable programming of complex graphical user interfaces. Rules, which have been defined with the RULEEDITOR, are parsed and translated to CLIPS with a compiler generated by YACC and OX [20,21].

7 Related Work

The knowledge-based system THEMPO clearly stands in the ONCOCIN/OPAL tradition. ONCOCIN [14] is an expert system supporting therapy planning and monitoring in adult oncology, and provides a graphical acquisition tool (OPAL

⁶ Adjacency-*matrices* should be considered as an appropriate graph representation when the question frequently arises whether a path exists between two given nodes n_1 and n_2 . In the context of protocol and patient graphs this problem type is less important than asking for edges of a given label leaving from or pointing to a given node n (for more details about graph representation and graph algorithms see [18]).

[22]) for protocol authoring similar to the tools described in section 2. However, OPAL and THEMPO for example differ in the acquisition of rule knowledge. OPAL provides several structured forms for important classes of rule knowledge which have been considered to be important in the domain; with these forms, the user can easily specify rules of a predefined structure. In the domain of pediatric oncology, however, the structure of rule knowledge used for stratification and therapy modification is too heterogeneous to be covered by an acceptable number of acquisition forms. Therefore, a rule editor has been implemented allowing the specification of rule knowledge not restricted to a small number of rule classes. Concerning the patient records of the two systems, the temporal network TNET/ETNET [23] of the ONCOCIN-record is comparable to the temporal graph of the THEMPO-patient record shown in figure 12. However, the ONCOCIN-record is not able to explicitly represent causal relationships between patient data, as it is for example shown in figure 12, where the causal context is represented, that ASPARAGINASE has been discontinued *because of* a detected hyperglycemia-trend. In the domain of protocol-directed pediatric oncology, where significant therapy modifications may result in removing the patient from the clinical trial associated with the protocol, the explicit representation of the medical decision process has been considered necessary.

CATIPO (COMPUTER AIDED THERAPY IN PEDIATRIC ONCOLOGY) [4] is a system for automated chemotherapy calculation in the domain of childhood cancer. It has a knowledge base for the different types of chemotherapy, and an alpha-numerical editor for chemotherapy definition. However, it does not provide modules supporting other types of therapy (e.g. radiation), has no representation of the protocol structure (such as the one in figure 1), and does not include a persistent patient record. Its encouraging results in clinical routine chemotherapy calculation – it is currently used in about 15 childhood cancer centers in Germany – have been one of the motivations for the THEMPO-project.

SEPIA [24] is an expert system for monitoring patients hospitalized in hematology departments to undergo clinical protocols. SEPIA, which is based on the control-theory paradigm, provides powerful mechanism for detecting critical laboratory value trends, for modifying therapy and diagnostic procedures as a consequence, and for meeting real-time constraints. From a more abstract, domain-independent point of view, THEMPO and SEPIA can be viewed as complementary systems: THEMPO focuses on elaborate therapy planning and configuration based on complex standardized protocols, SEPIA on the detection and management of critical situations during protocol-application.

ONCOCONS (= ONCOLOGY CONSULTANT) [25] is a knowledge based system covering a broad range of functions in the domain of adult oncology. This includes an oncological record with watch-dog functionality (ONCODOC), a

module for long-term monitoring (ONCOMON), a subsystem for radiotherapy (ONCOCART), and a multimedia-based training and information system (ONCOHELP). Beside the fact, that ONCOCONS is adapted to the domain of adult cancer diseases, which significantly differs from pediatric oncology, THEMPO focuses on knowledge acquisition and processing aspects, while the ONCOtools concentrate on an elaborate knowledge-based documentation and training functionality.

8 Conclusion and Open Problems

This article focused on the acquisition, representation, and processing of oncological protocols in the domain of childhood cancer. The knowledge-based system THEMPO, developed to support physicians in this complex domain, provides graphical tools for the acquisition of protocol knowledge, a problem-solver reasoning about adequate therapy steps for a given patient, and a patient record. THEMPO incorporates several methodologies, as the domain knowledge in the field of pediatric oncology is too heterogeneous to be covered by a single approach. Planning knowledge, for example, is represented through a semantic network; the complex network transformation processes driven by knowledge acquisition, problem solving and data entry, are formally described and controlled by a graph-grammar. Moreover, the network and graph-grammar representation is realized in an object-oriented manner. For example, the nodes in the network are instances of classes within an inheritance hierarchy, providing public methods for external modules to explore or modify their attributes or local neighborhood (e.g., a graph-grammar production inserting a new node and edge into the semantic network simply invokes an *AddNode(aNodeId)*- and an *InsertEdge(aSourceNodeId, aTargetNodeId, anEdgeId)*-method of the interface of the *Graph*-instance representing the network, which then handles the internal updating of the global node list and the affected adjacency-lists). On the other side, rule representation and data-driven rule processing, which is of importance for the subdomain of toxicity management, can be expressed and processed in a more natural and efficient way by a rule-based language and interpreter. In this case, the forward-chaining CLIPS-environment has been chosen, as CLIPS is easy to integrate into a C++-based application. Therefore, rules valid in the context of a particular chemotherapy are stored in the *RuleSet*-node of this chemotherapy, and are loaded by the CLIPS-interpreter (the interpreter itself is invoked by the global problem-solver as a subprocess), when a patient is treated with this chemotherapy.

However, there are several problems and shortcomings which have to be addressed in the future. First of all, the critical tool in the suite of acquisition tools is the RULEEDITOR, as it has not been proved yet whether this com-

plex experimental tool can really be used by physicians. The complexity is caused not mainly by building up the rule itself than by dealing with the error messages of the rule parser. Future efforts have to concentrate on informative and suggestive error messages of the parser understandable not only for computer scientists. Furthermore, as a powerful semantic check of rules detecting for instance subsuming or conflicting rules has not yet been developed, work has to be done addressing this important topic. Another shortcoming of the system is the inability to deal with complex trend knowledge which is of particular importance for an effective monitoring. Currently, THEMPO does not provide a mechanism to acquire, represent and process trend knowledge more complex than for instance the obviously simple hyperglycemia trend of figure 12. Future work, based on the works of Shahar [26] and Kahn [27], has to concentrate on this topic.

Furthermore, as large hospitals are characterized by extremely heterogeneous electronic information processing environments, additional work has to be done to integrate the system into existing, WINDOWS/SQL-dominated clinical networks and information systems.

A first phase of clinical testing and evaluation of the system has started in the begin of 1997 at the Children Hospital of the University of Mainz. The first step of evaluation has focused on the testing of the modules calculating a chemotherapy such as the one shown in figure 14, where the system has suggested the dosages of one of the involved drugs. This testing has been done by the following substeps:

- First of all, the systems had to *recalculate* a significant number of former chemotherapy treatments, which have already been applied to former patients, and which have been considered to be correct. The results of the recalculation then have been compared with the dosages that have been given. The basic intention of this step was to detect a number of relatively trivial system bugs, which have not been detected in the computer laboratory when using only a few generated test data of fictitious patients, and to demand the time of physicians only minimally, as this step was done mainly by the developers of the system. Furthermore, during this step, the relatively simple, though time-consuming acquisition of the chemotherapy knowledge was mainly done by the developers too (with the THERAPYEDITOR). The latter is possible, as protocols specify chemotherapy treatments precisely.
- The next step then consisted of calculating chemotherapy treatments of patients (currently being hospitalized at that time) simultaneously by the program and by an experienced physician, and by comparing the results. Again, chemotherapy acquisition has been done by the developers, to reduce the amount of work for the physicians during this step. Currently, the results of these two steps are analyzed.

- If the modules calculating the chemotherapy have been considered to work correct and to provide an ergonomic user interface, the next step consists of calculating the drug dosages of current patients by the system solely (including a final plausibility inspection of the dosages by the physician). As the manual calculation of complex chemotherapy units is one of the most time-consuming activities for physicians, a success concerning this step has to be viewed as being critical for the acceptance and motivation of physicians in using the complex system.
- If the physicians accept the chemotherapy calculation modules, testing shifts to acquisition aspects. First of all, the acquisition of the chemotherapy knowledge (with the THERAPYEDITOR) through physicians is tested, and it is analyzed how to optimize this process concerning user interaction and ergonomic issues.

If the chemotherapy modules (including the THERAPYEDITOR) as the core of the system have been proved to be correct and have been accepted by the medical staff, evaluation is extended to the more complex tools such as the PROTOCOLEEDITOR, the RULEEDITOR, and the problem solvers processing risk group assignments and suggesting new therapy units after recovering phases.

Acknowledgement

The authors thank Carlo Combi, Yuval Shahar and the two anonymous reviewers for their instructive remarks and suggestions to improve the paper.

References

- [1] M. Schrappe et al., Concept and interim analysis of trial ALL-BFM-90 for the treatment of children and adolescents with acute lymphoblastic leukemia: Significance of therapy response in peripheral blood and bone marrow (in German), *Klinische Pädiatrie*, **206** (1994) 208–221.
- [2] D. Pinkel, Lessons from 20 years of curative therapy of childhood acute leucaemia, *British Journal of Cancer* (1992) 148–153.
- [3] C.A. Stiller, Population based survival rates for childhood cancer in Britain 1980-91, *British Medical Journal* **309** (1994) 1612–6.
- [4] A. Bacherth, C.F. Classen, Wissensbasierte Chemotherapieplanung in der pädiatrischen Onkologie: Ein Beispiel zur Therapieunterstützung (Knowledge-based chemotherapy planning in pediatric oncology: an example for decision support - in German). In: W. Buchholz and R. Haux (eds.):

- [5] H. Ehrig, M. Nagl, and G. Rozenberg, eds., *Proceedings of the 3rd International Workshop on Graph Grammars and Their Applications to Computer Science (Dec. 1986)* (Springer, Berlin, 1987).
- [6] H. Ehrig, H.-J. Kreowski, and G. Rozenberg, eds., *Proceedings of the 4th International Workshop on Graph Grammars and Their Applications to Computer Science* (Springer, Berlin, 1991).
- [7] M. Nagl, A software development environment based on graph technology, in: [5] (1987) 458–478.
- [8] H. Göttler, *Zweistufige Graphmanipulationssysteme für die Semantik von Programmiersprachen*, PhD-Thesis (Institut für Mathematische Maschinen und Datenverarbeitung - Universität Erlangen-Nürnberg, 1977).
- [9] J. Egar and M.A. Musen, Graph-grammar productions for the modeling of medical dilemmas, in: *Proceedings 16th. Annual Symposium on Computer Application in Medical Care* (1992) 349–353.
- [10] J. Egar, A. Puerta, and M.A. Musen, Graph-grammar assistance for modeling of decisions, Technical report (Knowledge Systems Laboratory, Stanford University, 1992).
- [11] H. Göttler, *Graphgrammatiken in der Software-Technik. Theorie und Anwendungen* (Springer, Berlin, 1988).
- [12] H. Göttler, Semantic description by two-level graph grammars for quasi-hierarchical graphs, in: M. Nagl and H.J. Schneider, eds., *Proceedings of the Workshop on Graph-Theoretic Concepts in Computer Science* (Hanser, Munich, 1979) 207–226.
- [13] J. Giarratano and G. Riley, *Expert Systems: Principles and Programming* (PWS Publishing Company, 1993).
- [14] S.W. Tu, M.G. Kahn, M.A. Musen, et al., Episodic skeletal-plan refinement based on temporal data, *Communications of the ACM* **32(12)** (1989) 1439–1455.
- [15] G. Schreiber, B. Wielinga, H. Akkermans, W. Van de Velde, and A. Anjewierden, CML: The CommonKADS Conceptual Modelling Language, in: L. Steels, G. Schreiber, W. Van de Velde, eds., *A Future for Knowledge Acquisition. Proceedings 8th European Knowledge Acquisition Workshop (EKAW94)* (Springer, Berlin, 1994).
- [16] J. A. Breuker, W. Van de Velde (eds.), *The CommonKADS Library for Expertise Modelling* (IOS Press, Amsterdam, 1994).
- [17] R. Müller, O. Thews, C. Rohrbach, M. Serogl, and K. Pommerening, A graph-grammar approach to represent causal, temporal and other contexts in an oncological patient record, *Meth. Inform. Med.* **35(2)** (1996) 127–141.

- [18] T. H. Cormen, Ch. E. Leiserson, and R. L. Rivest, *Introductions to Algorithms* (MIT-Press, Cambridge, Massachusetts London, England, 1990).
- [19] D. Larkin, M. Morse, J. Neider, and C. Rose, *NeXTstep Concepts* (NeXT Computer, Inc., 900 Chesapeake Drive, Redwood City, CA 94063, 3.2 edition, 1993).
- [20] St. C. Johnson, Yacc: Yet anothers compiler compiler, Technical Report 32 (AT&T Bell Laboratories, New Jersey, 1975).
- [21] K. M. Bischoff, *Ox: An Attribute-Grammar Compiling System based on Yacc, Lex and C: User Reference Manual* (1993).
- [22] J.D. Walton, M.A. Musen, D.M. Combs, C.D. Lane, E.H. Shortliffe, and L.M. Fagan, Graphical access to medical expert systems: III. Design of a Knowledge Acquisition Environment, *Meth. Inform. Med.* **26(3)** (1987) 78–88.
- [23] M.G. Kahn, M. Fagan, and S. Tu, Extensions to the time-oriented database model to support temporal reasoning in medical expert systems, *Meth. Inform. Med.* **30** (1991) 4–14.
- [24] B. Séroussi, V. Morice, F. Dreyfus, and J.F. Boisvieux, Control theory as a conceptual framework for intensive care monitoring, *Artificial Intelligence in Medicine* **7** (1995) 155–177.
- [25] L. Novopashenny, and M.B. Wischnewsky, ONCODOC/ONCOHELP - ein wissensbasiertes Dokumentations- und Informationssystem für die Onkologie, Technical report (University of Bremen. Artificial Intelligence Laboratory, 1995).
- [26] Y. Shahar, *A Knowledge-Based Method for Temporal Abstraction of Clininal Data*, PhD-Thesis (Departments of Medicine and Computer Science, Stanford University, CA, 1994).
- [27] M.G. Kahn, *Model-Based Interpretation of Time-Ordered Medical Data*, PhD-Thesis (University of California, San Francisco, 1989).

Summary

This article introduced the knowledge-based system THEMPO (**T**herapy **M**anagement in **P**ediatric **O**ncology), which supports protocol-directed therapy planning and configuration in pediatric oncology. THEMPO provides several graphical acquisition tools for knowledge base authoring, which hide the internal representation language and enable the physician to edit protocols and units of therapy in a diagram-oriented way known from the paper-based protocols. The internal protocol representation approach is mainly based on semantic networks, but also uses a rule-based approach for specific subsets of protocol knowledge (e.g., toxicity detection and management). Graph-grammars have been used to formally describe the class of legal protocol graphs. A problem-solver, specified with the CML-modeling language, and implemented in C++/CLIPS suggests patient-adapted therapy plans and detects critical toxicity situations. The network/graph-grammar approach of the protocol knowledge base has been used for the corresponding patient record too, as the course of treatment of a particular patient can be viewed as a graph build around a path through the protocol network. Beside storing the diagnostic and therapeutic procedures, the patient record of THEMPO focuses on the explicit representation of semantic relationships between medical data, e.g. the record may represent the causal context that a CISPLATIN-infusion has been reduced *because of* a pathological creatinine-clearance value. The acquisition of contexts is also done by graphical tools. The next main steps have to concentrate on the analysis of the first clinical evaluation phase which has started at the begin of 1997, and an integration of the system into a WINDOWS/SQL-dominated hospital environment.

Biographical sketches

About the Author—ROBERT MÜLLER received his diploma from the faculty of mathematics and computer science of Mainz University, Germany, in September 1994. From October 1994 to June 1996 he worked as a computer research scientist at the University Hospital Mainz, where he was a member of the THEMPO-team at the Department for Medical Statistics and Documentation. Since July 1996, he is working at the Department of Computer Science of the University of Leipzig. His current research topics are conceptual models for electronic patient records, and knowledge-based therapy planning and workflow management in the field of adult oncology.

About the Author—MARITA SERGL received her diploma from the faculty of mathematics and computer science of Mainz University, Germany, in summer 1995. Since then, she is working as a research scientist at the University Hospital Mainz at the Department for Medical Statistics and Documentation. Her current research work focuses on therapy planning in oncology and clinical information systems.

About the Author—DIRK SCHOPPE is a student at the faculty of mathematics and computer science of Mainz University and wrote his diploma thesis about an oncological problem solver within the THEMPO-project.

About the Author—KLAUS POMMERENING received his diploma in mathematics in 1970 at the Freie Universität Berlin, and his Dr.rer.nat-degree in mathematics in 1972 at Mainz University. Since 1987, he is an professor at the University Hospital Mainz at the Department for Medical Statistics and Documentation. His current research activities include medical knowledge-based systems and data security and encryption.

About the Author—MATTHIAS DITTRICH received his board certification as a physician in 1975. From 1976-77 he worked at the Pathological Department of Fulda Hospital, Germany; since 1978 he is a physician at the Children's University Hospital Mainz, where he received his board certification as a pediatrician in 1982. Since 1985, he is a consultant for perinatal diagnostics. Dr. Dittrich was a member of several field studies about tropical pediatrics and schistosomiasis research (Congo Brazzaville 1984, Wad Medani Sudan 1985, Senegal 1993). Currently, he is a senior pediatric resident at the section for pediatric oncology at the Children's University Hospital Mainz. His specialities – beside oncology – are pediatric radiology and ultrasound.