

Ein graph- und objektorientiertes Datenbank-Modell für die Kinderonkologie

R. Müller

Institut für Informatik, Abteilung Datenbanken,
Universität Leipzig, Germany

`mueller@informatik.uni-leipzig.de`

M. Sergl, K. Pommerening

Institut für Medizinische Statistik und Dokumentation, Universitätskliniken Mainz,
Germany

`{sergl,pommerening}@imsd.uni-mainz.de`

Zusammenfassung

Dieser Beitrag stellt ein graph- und objektorientiertes Datenbank-Modell zur Abbildung komplexer kideronkologischer Behandlungsverläufe vor, das im Rahmen des Systems THEMPO (**T**herapy **M**anagement in **P**ediatric **O**ncology) entwickelt wurde. Der besondere Schwerpunkt lag hierbei auf einer expliziten Repräsentation der in diesem medizinischen Gebiet vielfach vorkommenden Quervernetzungen und medizinischen Beziehungen zwischen Daten. Ferner wurden graphische Werkzeuge für die Akquisition und das Retrieval von Daten-Vernetzungen entwickelt.

1 Einleitung

Die moderne Kinderonkologie [1] ist durch eine hohe Komplexität gekennzeichnet, die an eine Datenbank-Anwendung zur Dokumentation kideronkologischer Behandlungsverläufe hohe Anforderungen stellt. Das einem solchen Datenbanksystem zugrundeliegende Modell muß nicht nur in der Lage sein, die umfangreichen Maßnahmen zu Diagnostik, Therapie, Monitoring und Nachsorge abzubilden (z.B. Chemotherapie-Dosierungen, Laborwerte etc.), sondern sollte auch die zwischen diesen Maßnahmen vorliegenden vielfältigen Beziehungen repräsentieren, die z.B. kausale Kontexte ausdrücken können und somit Aussagen über den medizinischen Schlußfolgerungsprozeß beinhalten. Der vorliegende Beitrag stellt ein graph- und objektorientiertes Datenbank-Modell für die Kinderonkologie vor und beschreibt, wie sich komplexe Vernetzungen zwischen den Dateneinträgen in einer Patienten-Datenbank komfortabel durch graphorientierte Datenstrukturen abbilden lassen. Das hier vorgestellte Datenbank-Modell wurde im Rahmen des THEMPO-Projektes (**T**herapy **M**anagement in **P**ediatric **O**ncology) [2] entwickelt, dessen Zielsetzung die wissenschaftliche Unterstützung der Therapie maligner Erkrankungen des Kindes- und Jugendalters ist.

Abschnitt 2 skizziert eine Anforderungsanalyse für ein für die Kinderonkologie geeignetes Datenmodell, wobei vor allem die verschiedenen Vernetzungsmöglichkeiten zwischen Patientendaten erläutert werden und aufgezeigt wird, wie die explizite Abbildung dieser Vernetzungen die Arbeit des Arztes erleichtern kann. Abschnitt 3 diskutiert die Vorteile einer graphorientierten Sichtweise und Datenmodellierung, die Abschnitte 4 und 5 beschreiben Modell und Implementierung der kinderonkologischen Patienten-Datenbank. Abschnitt 6 schließlich geht auf Aspekte der Erfassung und des Retrievals von Daten-Vernetzungen ein, während in 7 und 8 verwandte Arbeiten im Bereich elektronischer Patienten-Datenbanken sowie noch offene Probleme des vorgestellten Ansatzes diskutiert werden.

2 Anforderungsanalyse

Eine im Rahmen des THEMPO-Projektes durchgeführte Analyse ergab, daß ein kinderonkologisches Datenbanksystem neben der naheliegenden Basisfunktionalität der Abbildung von Laborwerten, Befunden, Diagnosen und therapeutischen Maßnahmen auch tieferliegende Zusammenhänge und Vernetzungen zwischen den Daten persistent repräsentieren können sollte. Dabei sind vor allem die folgenden Vernetzungstypen zu berücksichtigen:

2.1 Kausale Zusammenhänge

Eine medizinische Maßnahme steht nicht für sich allein, sondern ist stets in einen *kausalen* Kontext eingebettet. So wird z.B. eine Diagnose *wegen* bestimmter vorliegender Befunde gefolgert, ein Therapie-Element z.B. *wegen* bestimmter kritischer Blutwerte abgebrochen. Gerade im Bereich der Erkennung und Behandlung von Nebenwirkungen, die ein ernstes Problem vor allem der aggressiven Chemotherapie darstellen, kann eine *explizite* Repräsentation des kausalen Zusammenhangs z.B. einer Medikationsmodifikation dem Arzt die retrospektive Rekonstruktion des Sachverhalts, den z.B. sein Kollege in der vergangenen Woche durchgeführt hat, wesentlich erleichtern. So gibt es z.B. viele Gründe für den Abbruch des Zytostatikums ASPARAGINASE (Hyperglykämie, Allergie etc.), die von Patient zu Patient variieren; speichert die Patienten-Datenbank ab, welche Befunde bei einem bestimmten Patienten für den Abbruch ausschlaggebend waren, kann dem Arzt das umständliche, manuelle „Blättern“ in der Datenbank erspart werden.

2.2 Konfliktkontexte

Nicht selten liegen über einen Patienten Daten vor, die vom behandelnden Arzt als *inkonsistent* zueinander eingestuft werden. So kann es z.B. sein, daß ein bestimmter Laborwert L aus medizinischer Sicht nicht zu einer bereits etablierten Diagnose D paßt, diese Diagnose aber notgedrungen beibehalten werden muß, da plausiblere Alternativ-Diagnosen fehlen. In diesem Zusammenhang kann es

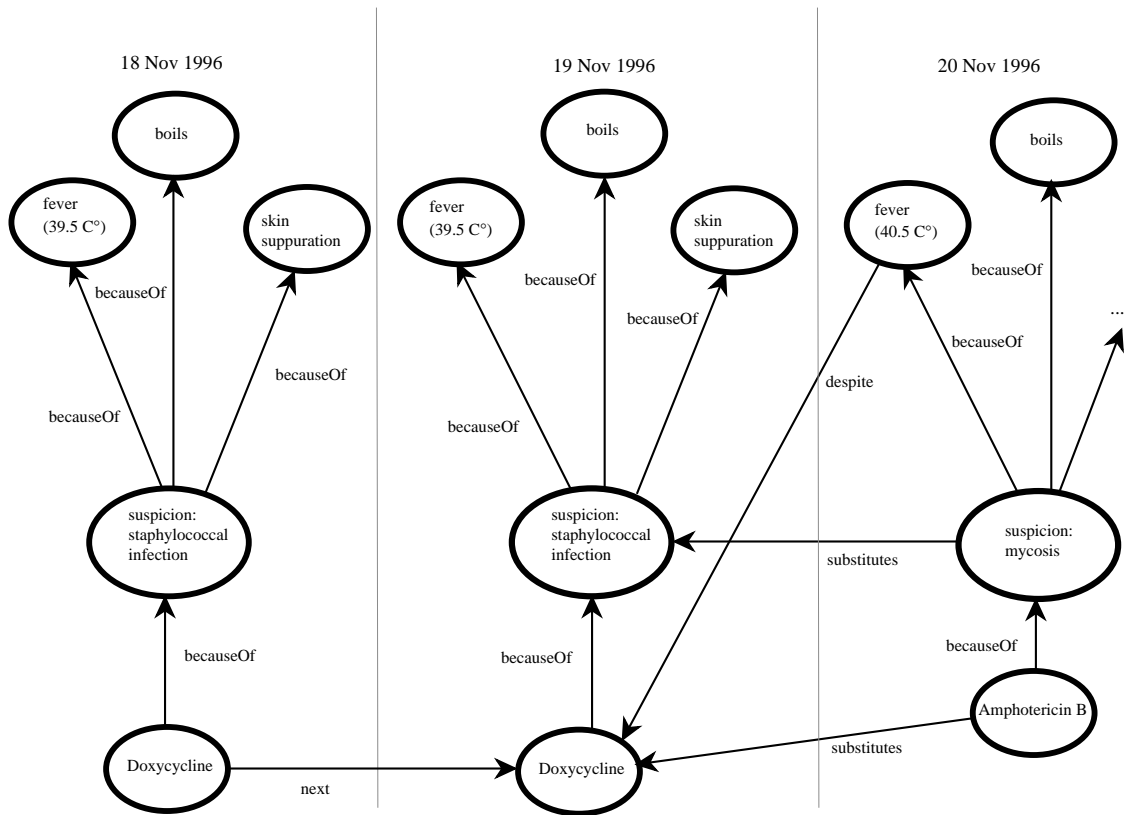


Abbildung 1: Revisionskontext während einer Infektionsbehandlung

sinnvoll sein, diese *Inkonsistenz*-Beziehung zwischen L und D explizit in der Datenbank abzubilden, damit immer dann, wenn einer der beiden Einträge abgerufen wird, vom System auch darauf hingewiesen werden kann, daß es zu dem Eintrag inkonsistente Daten gibt, die berücksichtigt werden müssen.

2.3 Revisionskontexte

Im klinischen Alltag ist das *Revidieren* von diagnostischen und therapeutischen Entscheidungen oft unvermeidbar. Neue Befunde können die momentan favorisierten Diagnosen „kippen“ und andere Medikationen implizieren. Da eine qualitativ hochwertige Dokumentation es stets ermöglichen sollte, *eindeutig* rekonstruieren zu können, über welche Information man zu welchem Zeitpunkt über einen Patienten verfügt hat, sollte es vermieden werden, eine aufgegebene Diagnose einfach in der Datenbank durch die neue Diagnose zu überschreiben; vielmehr sollten sowohl nach wie vor die alte als auch die neue Diagnose abgebildet werden, und zusätzlich die Revisionsbeziehung (Diagnose B ersetzt Diagnose A) zwischen beiden Einträgen repräsentiert werden.

Warum dies sinnvoll ist, zeigt Abb. 1. Diese Abbildung illustriert den Sachverhalt, daß ein Patient zwei Tage vergeblich mit einem

Antibiotikum (Doxycycline) behandelt wurde, um eine schwere Infektion (diese bedingt durch die immunschädigende Chemotherapie) zurückzudrängen; am dritten Tag mußte man den Verdacht auf eine bakterielle Infektion ersetzen durch den Verdacht auf eine systemische Pilzinfektion (mycosis), und demzufolge auch das Antibiotikum durch ein Pilzmedikament (Amphotericin B) austauschen. Werden in der Datenbank alle Informationen außer den mit *revidiert*, *wegen* und *trotz* markierten Kanten gespeichert, so kann bei einem Arzt, der sich zuerst den dritten Tag des Vorgangs anschaut, die Frage auftauchen, warum man das relativ ungewöhnliche Pilzmedikament eingesetzt hat. Werden zusätzlich die oben genannten Vernetzungen abgebildet, so ist eine eindeutige Beantwortung dieser Frage *durch das System* durch Abwandern der entsprechenden Kanten möglich; fehlen die Kanten, so kann die manuelle Rekonstruktion des Vorgangs zeitaufwendig werden. In komplizierteren Fällen kann es zudem sein, daß eine Rekonstruktion gar nicht mehr möglich ist.

3 Objekt- und graphorientierte Modelle

Objektorientierte Methoden (z.B. [3, 4]) bieten eine Fülle von Modellierungskonzepten (wie *Vererbung*, *Verkapselung*, *Aggregation* etc.), die bei komplexeren Anwendungen wesentlich vorteilhafter sind als das relativ einschränkende Relationenmodell. Allerdings kann ab einem gewissen Grad an Objektvernetzungen (wie dies z.B. bei den kinderonkologischen Patientendaten aus Abb. 1 der Fall ist) leicht der Überblick verloren gehen, wenn diese Vernetzungen durch (Referenz-)Attribute innerhalb der Attributstruktur einer Objektklasse modelliert werden. In einem solchen Fall beschreiben Attribute sowohl Wert-Eigenschaften, die einem Objekt lokal zukommen (wie z.B. Name und Code einer Diagnose, Dosierung einer Infusion), als auch - im Falle der referenzorientierten Attribute - Beziehungen zu anderen Objekten. Es kann in solchen Fällen von Vorteil sein, Objektbeziehungen nicht mehr über Attribute, sondern in sehr viel expliziterer Form durch das zusätzliche Modellierungsprimitiv *Kante* zu modellieren. Die Objektinstanzen selbst werden dann als (attributierte) Knoten aufgefaßt, während Beziehungen zwischen den Knoten durch (möglicherweise gerichtete) Kanten ausgedrückt werden. Zur Modellierung der Knoten und ihrer internen Struktur können dabei nach wie vor objektorientierte Konzepte wie *Vererbung*, *abstrakter Datentyp*, *Polymorphismus* etc. herangezogen werden. Die Vorteile, die graphorientierte Sichtweisen ab einer gewissen Komplexität der zu modellierenden Domäne bieten, haben in den letzten Jahren zu verstärkten Forschungsaktivitäten auf dem Gebiet der graphorientierten Modellierung und der graphorientierten Datenbanken geführt [6, 7, 8].

Formal betrachtet, ist ein (gerichteter) Graph G über eine Menge von Knotenklassen *NodeClasses* und einer Menge von Kantenmarkierungen *EdgeLabels* im folgenden ein Tupel $G = (Nodes, Edges)$ mit

1. o.B.d.A $Nodes \subset \mathbb{N}$ („Knoten-Id's“) und

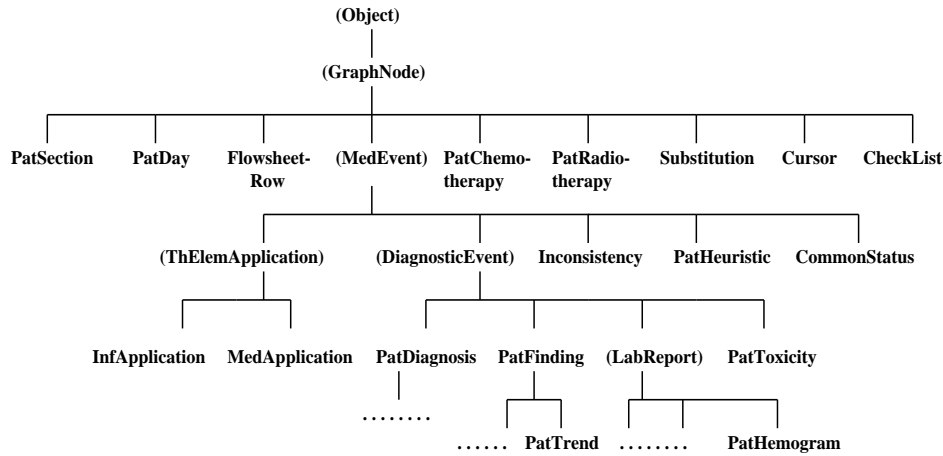


Abbildung 2: Vererbungshierarchie der verwendeten Knotenklassen. Mit runden Klammern gekennzeichnete Klassen sind *abstrakt*, d.h. es können nur Knoten der Unterklassen im Graphen vorkommen. Mit Punkten (...) sind in dieser Abbildung ausgesparte Unterbäume gekennzeichnet. *InfApp* und *MedApp* sind Abkürzungen für *InfApplication* und *MedApplication*.

$$Class(n) \in NodeClasses \text{ für alle } n \in Nodes^1$$

$$2. Edges \subset Nodes \times Nodes \times EdgeLabels$$

Ein Element $(n_1, n_2, e) \in Edges$ wird als gerichtete Kante der Markierung e vom Quell-Knoten n_1 zum Ziel-Knoten n_2 aufgefaßt. Die Definition von $Edges$ als Teilmenge eines kartesischen Produkts impliziert ferner, daß es zwischen zwei Knoten eines Graphen nicht zwei Kanten gleicher Richtung und Markierung geben kann. Fordert man zusätzlich für alle $(n_1, n_2, e) \in Edges$, daß $n_1 \neq n_2$, so heißt dies, daß keine *Schleifen* zugelassen sind.

4 Das graphorientierte Patienten-Datenmodell

Im folgenden wird nun das der Datenbank-Anwendung zugrundeliegende Patientenmodell beschrieben. Abb. 2 zeigt die Vererbungshierarchie der verwendeten Knotenklassen; jedes medizinische Ereignis, das im Verlauf der Behandlung auftritt - wie z.B. die Erhebung eines Befundes oder die Applikation einer zytostatischen Infusion - wird durch einen Knoten repräsentiert, dessen Klasse von *MedEvent* (für *Medical Event*) abgeleitet ist.

Jeder neue *MedEvent*-Knoten wird hierbei zuerst in zwei Basis-Kontexte eingebettet, den *Tageskontext* und den *Verlaufskontext* (siehe Abb. 3). Im Tageskontext wird der Ereignisknoten mit demjenigen *PatDay*-Knoten verbunden, der

¹Die Funktion $Class()$ ordnet einem Knoten dessen Klasse zu.

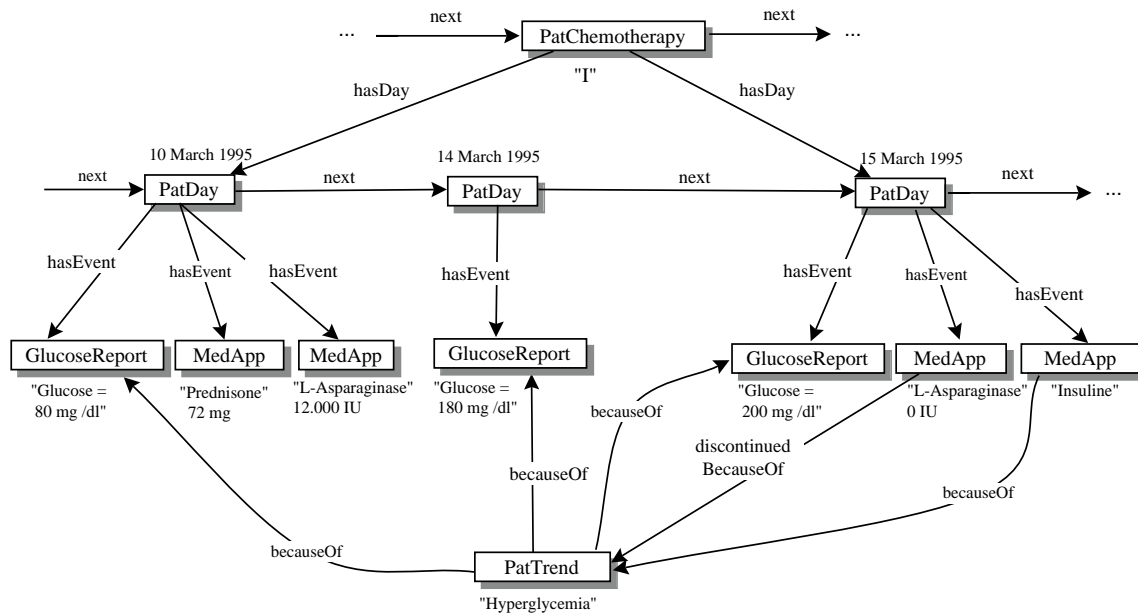


Abbildung 4: Teil eines Patientengraphen, der kausale Vernetzungen zwischen Daten repräsentiert. ('I' ist eine spezielle Chemotherapie des Leukämieprotokolls 'ALL-BFM-95').

Tages- und Verlaufskontext bilden das Grundgerüst eines Patientengraphen; graphentheoretisch gesehen, läßt sich über jeden der beiden Kontexte jeweils ein *spannender* Baum definieren, also ein Baum, der alle Knoten des Graphen umspannt (i.a. aber nicht alle Kanten).

Alle weiteren medizinischen Beziehungen und Vernetzungen, die zwischen Knoten vorliegen können und in Abschnitt 2 klassifiziert wurden, werden dann durch weitere Kanten abgebildet. Abb. 4 zeigt die Repräsentation eines kausalen Kontextes, bei dem man bei einem Patienten das Zytostatikum ASPARAGINASE abbrechen mußte, weil der Patient eine lebensbedrohliche Hyperglykämie entwickelt hat (letztere ist eine mögliche Nebenwirkung dieses Medikaments).

Die Knotenklassen selbst sind – ganz im objektorientierten Sinne – strukturierte Typen mit eigenen Attributen und Methoden, die entweder auf die Attribute zugreifen oder die lokale Umgebung des Knotens erkunden. So verfügt z.B. die Knotenklasse *PatSection*, deren Instanzen die im **TOC**-Graph in Abb. 3 gezeigten Abschnitte und Unterabschnitte einer Patientenakte repräsentieren, über das folgende Interface:

```

interface PatSection: GraphNode
{
    key: sectionName;

    string          sectionName;          // Name des Abschnitts

    Integer         numSubSections();     // liefert Anzahl der
                                           // direkten Unterabschnitte
    List<string>    subSectionNames();    // liefert Namensliste aller
                                           // direkten Unterabschnitte
    Tree<string>    subSectionTree();     // liefert Namensbaum aller
                                           // Unterabschnitte
}

```

Hierbei liefert z.B. die Methode *subSectionTree* den Baum aller Unterabschnitte, der zum Aufbau eines hierarchischen Inhaltsverzeichnis für die Benutzungsoberfläche dienen kann.

Zur formalen Beschreibung der Menge der syntaktisch zulässigen Graphen wurde eine Graphgrammatik erstellt, die definiert, wie - ausgehend von einem *Startgraphen* (dem *Startsymbol* der Graphgrammatik) - durch Graphproduktionen größere Graphen aufgebaut werden können. Diese Graphgrammatik wird nicht *analytisch* verwendet, d.h. es wird nicht erst durch den Anwender ein Graph definiert, dessen syntaktische Zulässigkeit dann durch die Graphgrammatik überprüft werden muß, sondern *generativ*, d.h. vom Arzt an der Benutzungsoberfläche getätigte Aktionen (wie das Verordnen einer neuen Infusion) lösen im Graphen entweder eine einzelne Graphproduktion oder mehrere, zu einer Graph-Transaktion zusammengefaßte Produktionen aus, die dann den Graphen entsprechend erweitern und umbauen. Der Term *Graph-Transaktion* bezeichnet hierbei - im Sinne des Transaktionskonzeptes - eine ACID-Operation; ferner wird dieses high-level Transaktionskonzept auf Graph-Ebene durch das Transaktionskonzept des für die Implementierung der graphorientierten Patientenakte verwendeten Datenbanksystems realisiert (siehe Abschnitt 5). Eine detaillierte Beschreibung der Patienten-Graphgrammatik, die hier aus Platzgründen nicht gegeben werden kann, findet sich in [9].

5 Implementation

Die Implementation des graphorientierten Kernmodells wurde mit Hilfe des objektorientierten Datenbanksystems POET vorgenommen². POET ist ein Client/Server-System und bietet - wie die meisten anderen objektorientierten

²Die Verwendung einer rein graphorientierten Datenbank (wie z.B. GRAS [8]) kam wegen der für klinische Anwendungen noch nicht ausreichenden Stabilität dieser Forschungsprototypen zu diesem Zeitpunkt vorerst nicht in Frage.

Datenbanken - eine die Persistenz und Verwaltung von Objekten unterstützende Erweiterung der Sprache C++ an, sowie eine Implementation von OQL. Für die Graph-Repräsentierung und Graph-Verwaltung wurden die abstrakten Datentypen *Graph* und *GraphNode* implementiert; die Klasse *Graph* bietet z.B. Methoden zum Einbau und Löschen von Knoten (*GraphNode*-Objekten) an und realisiert Routinen zum Abspeichern des Graphen. Für die Repräsentierung von Kanten wurde ein an *Adjazenz-Listen* orientierter Ansatz gewählt. Dieser ist gegenüber einer *Adjazenz-Matrizen*-Darstellung deswegen besser geeignet, da im Rahmen der kideronkologischen Anwendung vor allem Anfragen nach der Nachbarschaft eines Knotens häufig sind. So wird z.B. häufig die Anfrage gestellt, mit welchen anderen Knoten ein gegebener *ThElemApplication*-Knoten über kausale Kanten verbunden ist, um so Informationen über die medizinischen Gründe der durch diesen Knoten repräsentierten therapeutischen Maßnahme abzufragen. Eine an *Adjazenz-Matrizen* orientierte Repräsentierung wäre bei häufigen Pfadproblemen von Vorteil (wie z.B.: Gibt es einen Pfad der Länge k zwischen zwei gegebenen Knoten n_1 und n_2 ?; solche Anfragen können durch Multiplikation der Adjazenz-Matrix beantwortet werden). Anfragen des letzteren Typs sind im Kontext dieser speziellen Anwendung aber eher selten.

6 Akquisition und Retrieval

Das in diesem Beitrag vorgestellte Modell macht keine Annahmen darüber, auf welche Art und Weise medizinische Vernetzungen zwischen Daten *erfaßt* werden. Im Rahmen des THEMPO-Projektes werden grundsätzlich zwei Arten der Erfassung verfolgt, nämlich die *automatisierte* Akquisition und die Erfassung durch den Arzt selbst. Unter automatisierter Erhebung ist zu verstehen, daß das Inferenzmodul von THEMPO, welches versucht, anhand der vorliegenden Patientendaten mit Hilfe eines Regelinterpreters Schlüsse über den Patienten zu ziehen, z.B. *kausale* Vernetzungen zwischen Patientendaten automatisch ermittelt und über eine Schnittstelle der Patienten-Datenbank zur persistenten Abbildung übergibt. Im Falle von Abb. 4 ist es z.B. denkbar, daß der Abbruch von ASPARAGINASE wegen der pathologischen Glucose-Werte von der Inferenzmaschine vorgeschlagen wurde, und der kausale Zusammenhang zwischen den für das „Feuern“ entsprechender Regeln verantwortlichen Glucose-Werten und der finalen Folgerung (*Abbruch von ASPARAGINASE*) über Schnittstelle der Datenbank-Anwendung mitgeteilt worden ist, welche dann für die interne Abbildung im Graphen (durch das Einfügen einer *discontinuedBecauseOf*-Kante) sorgt. Da allerdings keine Wissensbasis und automatische Inferenz das Wissen einer medizinischen Disziplin vollständig abzudecken vermag, stellt THEMPO zusätzliche Werkzeuge bereit, die es dem Arzt ermöglichen, medizinische Datenvernetzungen interaktiv in der Datenbank abzubilden. Abb. 5 illustriert diesen Vorgang: Die linke Spalte des Fensters zeigt CISPLATIN-Dosierungen, die von der THEMPO-Inferenz vorgeschlagen wurden, die rechte Spalte führt die vom Arzt nachträglich manuell reduzierten Dosierungen auf. Durch Markieren des Fensters in einem speziellen Mausmodus und Dragen der Maus zu dem Feld mit dem Creatinine-Laborwert bildet sich ein Pfeil auf dem Bildschirm, der

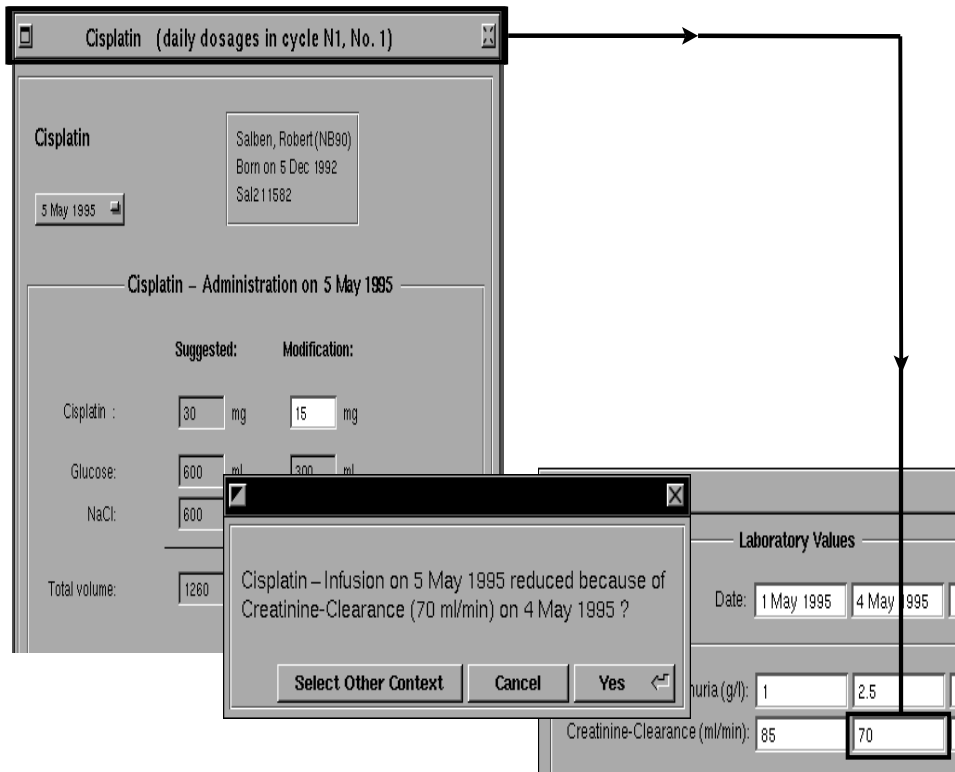


Abbildung 5: Bildschirmabzug der graphischen Erfassung einer medizinischen Datenbeziehung (fiktiver Patient). Weitere Erläuterungen im Text.

die zu spezifizierende Beziehung zwischen den Dosierungen auf der einen Seite und dem Creatinine-Wert auf der anderen Seite visualisiert. Die Semantik der Beziehung (*Reduktion wegen*) wird in einem anschließenden PopUp-Menu spezifiziert. Bestätigt der Arzt die dann vom System generierte Nachfrage, wird in der Datenbank zwischen dem Knoten, welcher die CISPLATIN-Infusion repräsentiert, und dem den Creatinine-Clearance-Wert abbildenden Knoten eine Kante mit der Markierung *reducedBecauseOf* gezogen.

6.1 Retrieval von Datenvernetzungen

Auch das *Retrieval* von Datenvernetzungen kann der Arzt mit mausbasierten Werkzeugen durchführen. Wurde z.B. zuerst nur das Fenster mit den CISPLATIN-Dosierungen vom Arzt geöffnet, und taucht die Frage auf, *warum* die Dosierungen um die Hälfte reduziert wurden, so kann der Arzt wiederum in einem PopUp-Menu spezifizieren, daß er an den *kausalen* Vernetzungen der CISPLATIN-Infusion interessiert ist. Die Datenbank-Anwendung sucht daraufhin im Patientengraphen nach kausalen Kanten, die von dem die Infusion repräsentierenden Knoten ausgehen. Im Falle des in Abb. 5 spezifizierten kausalen Kontextes wird eine mit *reducedBecauseOf* markierte Kante gefunden werden, die zu dem Creatinine-Clearance-Knoten führt. Daraufhin wird ein Fenster mit den Creatinine-Clearance-Daten generiert, und es erscheint analog zu Abb. 5 ein

Pfeil auf dem Bildschirm mit dem Hinweis, daß die CISPLATIN-Infusion wegen der besagten Laborwerte reduziert wurde.

7 Vergleich mit anderen Arbeiten

Im Bereich der Medizin-Informatik wurde das Problem der medizinischen Vernetzungen zwischen Daten bereits in [10] und [11] diskutiert; allerdings beschränken sich die dortigen Ansätze ausschließlich auf einfache kausale Kontexte. Ferner sind Modellierung und Implementierung in den beiden genannten Arbeiten relational orientiert; die medizinischen Beziehungen zwischen Daten-Objekten müssen in zusätzlichen Tabellen repräsentiert werden, wodurch die Kontext-Repräsentation auf unübersichtliche Weise aufgesplittet wird.

8 Zusammenfassung und offene Probleme

Im Beitrag wurde gezeigt, daß eine graphorientierte Sichtweise und Modellierung in Domänen wie der Kinderonkologie, bei denen die zwischen Daten auftretenden Vernetzungen und Beziehungen eine gewisse Komplexität erreichen, gegenüber rein objektorientierten Sichtweisen gewisse Vorteile u.a. bzgl. Lesbarkeit, verbesserter Navigation und Übersichtlichkeit des Modells bringen kann. Im Rahmen des THEMPO-Projektes, dessen Zielsetzung die wissensbasierte Unterstützung bei der Behandlung von Kinderkrebs ist, wurde ein solches graphorientiertes Datenmodell als Grundlage einer Patienten-Datenbank für die Abspeicherung onkologischer Behandlungsverläufe entwickelt. Die formale Beschreibung des Graph-Modells erfolgte mit einer Graph-Grammatik, die Implementation wurde mit einer objektorientierten Datenbank realisiert. Für die Erfassung und das Retrieval von Datenkontexten wurden, neben automatisierten Ansätzen, auch interaktive Werkzeuge entwickelt, die es dem Arzt ermöglichen, Vernetzungen zwischen Daten auf dem Bildschirm graphisch zu editieren und zu visualisieren.

Eine Limitation des vorgestellten Ansatzes besteht darin, daß die Abbildung *hierarchischer* Kontexte nicht unterstützt wird. So könnte es z.B. notwendig sein, daß der in Abb. 5 erfaßte kausale Kontext des CISPLATIN-Abbruchs wiederum als Gesamtheit und damit quasi als Knoten einer höheren Ordnung in einen größeren Kontext eingeht, was durch den oben beschriebenen Ansatz nur ungenügend unterstützt wird.

Ein weiteres Problem besteht in der eingeschränkten *Granularität* des Patientengraphen: Beziehungen und damit medizinische Kontexte können nur auf der Ebene von Knoten abgebildet werden, was z.B. dann nicht ausreichend ist, wenn ein Medikamentenabbruch nicht wegen eines Blutbildes allgemein (also eines *PatHemogram*-Knotens) erfolgt, sondern wegen eines bestimmten Attributwertes innerhalb des Blutbildes (also z.B. allein wegen der Leukozytenzahl). Wird ein Blutbild dagegen zu einem Baum aufgefächert, bei dem die einzelnen Bestandteile des Blutbildes in jeweils eigenen Knoten repräsentiert werden,

so kann eine Kante zwischen dem abgebrochenen Medikamenten-Knoten und demjenigen Knoten des Blutbild-Baumes, der kausal für den Abbruch verantwortlich ist, gezogen werden, wodurch der kausale Kontext in eindeutiger Weise repräsentiert wird.

Derzeitige Arbeiten konzentrieren sich daher auf den Einsatz der von J. Sowa entwickelten sogenannten konzeptuellen Graphen [12, 13, 14], die eine Hybridisierung der Prädikatenlogik 1. Stufe mit graphorientierten Strukturen darstellen. Mit konzeptuellen Graphen lassen sich sowohl hierarchische medizinische Kontexte als auch Graphen einer feineren Granularität realisieren.

Literatur

- [1] P. Gutjahr. *Krebs bei Kindern und Jugendlichen*. Deutscher Ärzte-Verlag, Köln, 1993.
- [2] R. Müller, M. Sergl, U. Nauwerth, D. Schoppe, and K. Pommerening. TheMPO: A knowledge-based system for therapy planning in pediatric oncology. *To appear in Computers in Biology and Medicine*, 1997.
- [3] G. Booch. *Object-Oriented Analysis and Design*. Benjamin Cummings, New York, 1994.
- [4] R.G.G. Cattell. *Object Data Management. Object-Oriented and Extended Relational Database Systems*. Addison-Wesley, 1993.
- [5] H. Goettler. *Graphgrammatiken in der Software-Technik. Theorie und Anwendungen (Graph Grammars in Software Engineering. Theory and Applications) (In German)*. Springer-Verlag, Berlin Heidelberg New York, 1988.
- [6] M. Gyssens, J. Paraedaens, and D. van Gucht. A graph-oriented object database model for database end-user interfaces. In *Proc. ACM SIGMOD Int. Conference on Management of Data, SIGMOD Record, vol. 19-2*, pages 24–33. ACM Press, 1990.
- [7] M. Andries, M. Gemis, J. Paredaens, I. Thyssens, and J. Van den Bussche. Concepts for graph-oriented object manipulation. In A. Pirrotte, C. Delobel, and G. Gottlob, editors, *Advances in Database Technology - EDBT'92, 3rd international Conference on Extending Database Technology, Vienna, Austria, March 23-27, 1992 (LNCS 580)*, pages 21–38. Springer, 1992.
- [8] N. Kiesel, A. Schuerr, and B. Westfechtel. GRAS, a graph-oriented (software) engineering database system. In *Information Systems, vol. 20, no. 1*, pages 21–52. Pergamon Press, Oxford, 1995.
- [9] R. Müller, O. Thews, C. Rohrbach, M. Sergl, and K. Pommerening. A graph-grammar approach to represent causal, temporal and other contexts in an oncological patient record. *Methods of Information in Medicine*, 35(2):127–141, 1996.

- [10] A.L. Rector and S. Kay. Descriptive models for medical records and data interchange. In B. Barber, D. Cao, D. Quin, and G. Wagner, editors, *MEDINFO 89*, pages 230 – 234, Amsterdam, 1989. North-Holland.
- [11] R.C. Barrows and S.B. Johnson. A data model that captures clinical reasoning about patient problems. In *Proceedings of the 19th Annual Symposium on Computer Applications in Medical Care (SCAMC)*, 1995.
- [12] J.F. Sowa. *Conceptual Structures*. Addison Wesley, Reading (Massachusetts), 1984.
- [13] J.F. Sowa. Conceptual graphs summary. In T.E. Nagle, J.A. Nagle, L.L. Gerholz, and P.W. Eklund, editors, *Conceptual Structures: Current Research and Practice*, pages 1–51. Ellis Horwood, 1992.
- [14] John Esch and Robert Levinson. An implementation model for contexts and negation in conceptual graphs. In *Proceedings of the 3rd International Conference of Conceptual Structures, Santa Cruz, CA, USA, August 1995*, pages 247–262.