



KfK 4822
Januar 1991

Datenbank/Wissensbasis- Konsistenzmonitor

W. Cellary, A. Jaeschke, T. Morzy, H. Orth, G. Zilly
Institut für Datenverarbeitung in der Technik
Projekt Schadstoffbeherrschung in der Umwelt

Kernforschungszentrum Karlsruhe

Kernforschungszentrum Karlsruhe
Institut für Datenverarbeitung in der Technik
Projekt Schadstoffbeherrschung in der Umwelt

KfK 4822

DATENBANK / WISSENSBASIS-KONSISTENZMONITOR

W. Cellary¹, A. Jaeschke, T. Morzy¹, H. Orth, G. Zilly

¹Technische Universität Poznan, 60-965 Poznan (Polen)

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Database - Knowledge Base Consistency Monitor

Abstract

Expert systems composed of a database and a knowledge base, which cooperate, are considered. In such systems mutual consistency of the database and the knowledge base must be enforced. A consistency monitor is proposed for this purpose, whose task is double: to enforce internal consistency of the database and mutual consistency of the database and the knowledge base. The first task comprises enforcement of referential constraints imposed on the database. An efficient technique is proposed. Referential constraints are enforced by the use of DBMS journals instead of looking up the entire database. The consistency monitor described in this paper is applied in the expert system ELAN installed in the analytical laboratories of the nuclear fuel reprocessing pilot plant at the Kernforschungszentrum Karlsruhe.

Zusammenfassung

Im vorliegenden Bericht werden Expertensysteme bestehend aus einer Datenbank und einer Wissensbasis beschrieben. In Systemen dieser Art muß sichergestellt sein, daß Datenbank und Wissensbasis konsistent sind. Zu diesem Zweck wurde ein Konsistenzmonitor entwickelt, der sowohl für Konsistenz innerhalb der Datenbank als auch für Konsistenz von Datenbank und Wissensbasis sorgen soll. Im ersten Fall bedeutet dies, daß referentielle Bedingungen, die an die Datenbank gestellt werden, erfüllt werden müssen. Dafür wird hier ein sehr effizientes Verfahren vorgeschlagen. Die Erfüllung der referentiellen Bedingungen erfolgt unter Verwendung von DBMS-Journals. Ein Durchsuchen der gesamten Datenbank ist damit nicht mehr erforderlich. Der im vorliegenden Bericht beschriebene Konsistenzmonitor wurde für das in den analytischen Labors der Wiederaufarbeitungsanlage Karlsruhe eingesetzte Expertensystem ELAN entwickelt.

1 Einleitung

Zahlreiche Expertensysteme bestehen aus einer Datenbank und einer Wissensbasis, die miteinander kooperieren. In derartigen Systemen ist das Problem der Konsistenz von Datenbank und Wissensbasis von entscheidender Bedeutung. Zur Lösung dieses Problems wurde ein Konsistenzmonitor entwickelt. Dieser soll in dem für die analytischen Labors der Wiederaufarbeitungsanlage Karlsruhe konzipierten Expertensystem ELAN eingesetzt werden.

Zur Kontrolle der Wiederaufarbeitung von Kernbrennstoffen werden leistungsfähige analytische Labors benötigt, in denen physikalische und chemische Untersuchungen der verschiedenen Prozeßbestandteile und entstehenden Produkte durchgeführt werden können. Hauptaufgabe des ELAN-Systems ist die Unterstützung des Schichtführers bei der Erstellung von Analysenplänen. Dazu gehört auch die Auswahl geeigneter Analyseverfahren, die Bestimmung der einzelnen Arbeitsschritte (Probenvorbereitung, Messung, Auswertung, Auswahl der Meßgeräte, etc.) und die Verteilung der Arbeitsschritte auf die Laborarbeitsplätze.

Das ELAN-System ist aus zwei Rechnern aufgebaut. Der VAX-Rechner mit der INGRES-Datenbank dient zur Erfassung und Speicherung von Daten. Es handelt sich um ein Mehrbenutzersystem mit Terminals an den einzelnen Laborarbeitsplätzen. Auf dem zweiten Rechner, derzeit einem TI-Explorer, befindet sich die Wissensbasis mit der Erklärungskomponente. Es ist ein Einbenutzersystem, das den Schichtführer bei Entscheidungen bezüglich der Planung unterstützen soll. Die Kopplung zwischen beiden Komponenten ist nicht stark. Damit wird sichergestellt, daß sie unabhängig voneinander arbeiten können (s. Abb. 1).

Bestimmte im System abgespeicherte Informationen sind in beiden Rechnern vorhanden. Der Grund dafür liegt zum einen in der Notwendigkeit unterschiedlicher Darstellungen von Informationen in den beiden Rechnern und zum anderen in der Tatsache, daß die Kommunikation zwischen den beiden Komponenten relativ langsam erfolgt. Durch diese Redundanz von Daten wird das Volumen von zu übertragenden und umzuwandelnden Daten minimiert. Auf dieses Prinzip kann also auch in Zukunft nicht verzichtet werden.

Die Kooperation zwischen der Datenbank und der Wissensbasis findet nicht kontinuierlich statt. Sie ist lediglich bei der Erstellung eines neuen Analysenplans durch den Schichtführer erforderlich.

Das ELAN-System arbeitet folgendermaßen: Auf der VAX werden die Daten kontinuierlich erfaßt und verarbeitet. Unter Einbeziehung der Wissensbasis auf

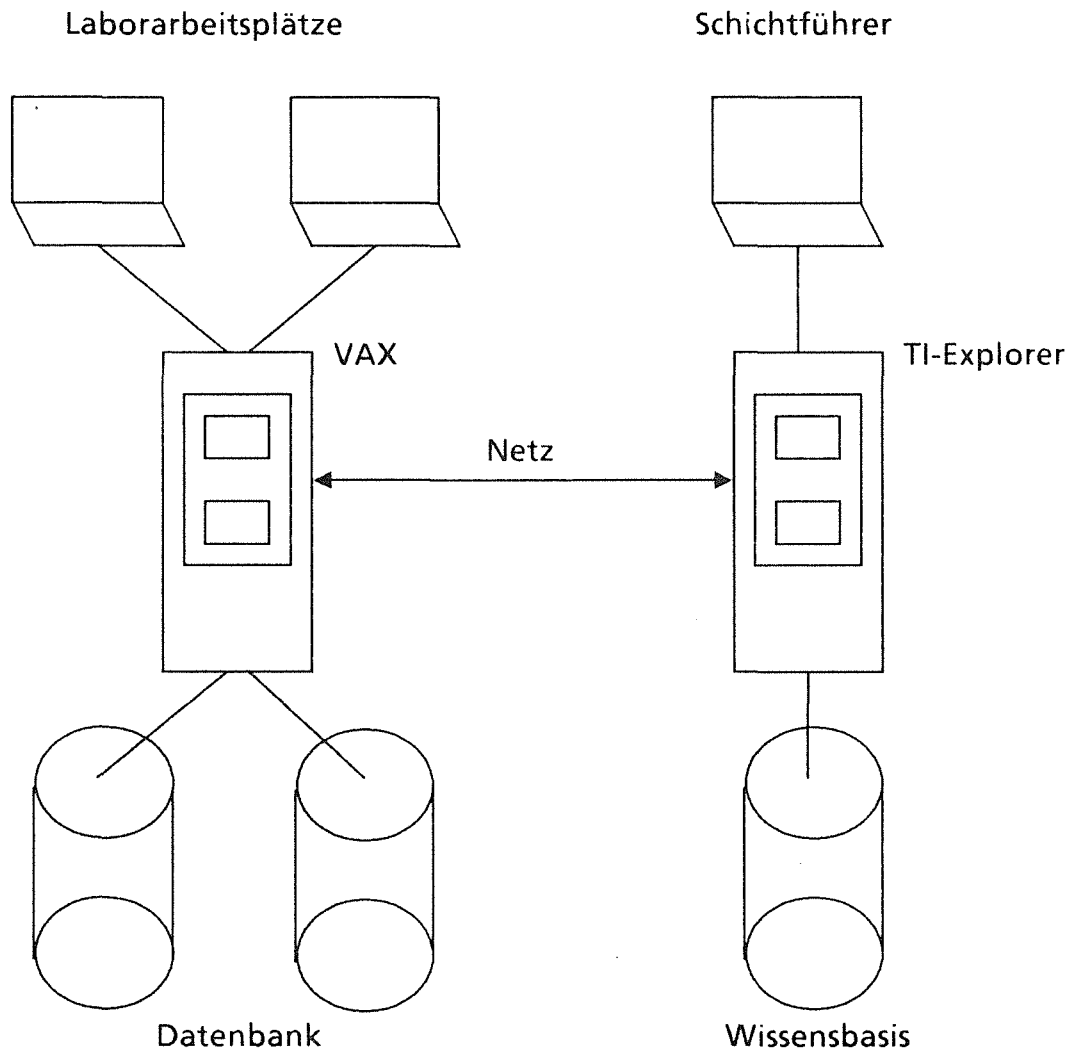


Abb. 1: Struktur des ELAN-Systems

dem TI-Explorer wird von Zeit zu Zeit ein neuer Analysenplan erstellt. Gerade dazu ist die Konsistenz von Datenbank und Wissensbasis unbedingt erforderlich. Ihre Realisierung erfolgt durch Einfügen neuer Tatsachen (facts) in die Wissensbasis bzw. Löschen bereits vorhandener Daten in Übereinstimmung mit den Änderungen, die seit der letzten Erstellung eines Analysenplans in der Datenbank vorgenommen worden sind.

Um Konsistenz zwischen der Datenbank und der Wissensbasis herzustellen, muß die Datenbank ebenfalls konsistent sein. Andernfalls könnten falsche Daten in die Wissensbasis eingebracht oder richtige Informationen aus ihr gelöscht werden, was zu einer fehlerhaften Erklärung führen würde. Konsistenz innerhalb der Datenbank selbst liegt vor, wenn die Datenbankintegritätsbedingungen erfüllt sind. Man unterscheidet zwei Arten von Integritätsbedingungen:

Wertebedingungen und referentielle Bedingungen /5/. Während erstere automatisch durch INGRES realisiert werden, ist dies bei den zweiten nicht der Fall. Bisher war zur Erfüllung der referentiellen Bedingungen ein Durchsuchen der gesamten Datenbank erforderlich, was das Verfahren sehr aufwendig und wenig effektiv machte. Im vorliegenden Bericht wird nun ein neues Verfahren beschrieben, das auf der Verwendung des DBMS-Journaling-Systems beruht. Zur Erfüllung der referentiellen Bedingungen muß nun nicht mehr die gesamte Datenbank durchsucht werden.

Abschnitt 2 des vorliegenden Berichts beschäftigt sich mit der Herstellung der internen Datenbankkonsistenz durch die Verwendung von Journals. Das Problem der Konsistenz von Datenbank und Wissensbasis wird dann in Abschnitt 3 untersucht. Die Schlußfolgerungen werden abschließend in Abschnitt 4 aufgeführt.

2 Sicherstellung der internen Datenbankkonsistenz

Die Integritätsbedingungen stellen die zeitlich invariante Semantik der in der Datenbank gespeicherten Daten dar /5/. Sie müssen erfüllt werden, um sicherzustellen, daß die Datenbank die von ihr modellierte reale Welt richtig widerspiegelt. Ein Datenbankzustand ist konsistent, wenn alle Integritätsbedingungen, denen die Datenbank unterliegt, erfüllt werden. Mit Hilfe des Datenbank-Managementsystems (DBMS) wird dafür gesorgt, daß jeder Datenbankzustand konsistent ist. Dazu wird sichergestellt, daß bei jeder Änderung der Datenbank die Integritätsbedingungen erfüllt bleiben.

Man unterscheidet zwei Arten von Integritätsbedingungen: Wertebedingungen und referentielle Bedingungen. In den Wertebedingungen werden Informationen über zulässige Werte der in der Datenbank gespeicherten Objekte verschlüsselt. Werte eines das Alter einer Person beschreibenden Attributs können beispielsweise weder kleiner als 0 noch größer als 110 sein. Werden Integritätsbedingungen dieser Art durch das Statement DEFINE INTERITY ausgedrückt, findet die Erfüllung automatisch mit Hilfe von INGRES DBMS statt. In referentiellen Bedingungen werden strukturelle Beziehungen zwischen Relationen, die Objekte der realen Welt beschreiben, verschlüsselt. Referentielle Bedingungen treten auf, wenn in einer Relation Referenzen auf eine andere Relation zu finden sind. So muß beispielsweise der Wert des Attributs Employee__name in der Relation University__teachers als Wert des Attributs Employee__name in einem Tupel der Relation University__employee auftreten. Allgemein ausgedrückt bedeutet dies: Eine referentielle Bedingung sagt aus,

daß, wenn eine Relation R_2 als Attribut einen fremden Key (FK) enthält, der mit dem primären Key (PK) einer anderen Relation R_1 übereinstimmt, dann muß jeder Wert von FK in R_2 entweder gleich dem Wert von PK in einem bestimmten Tupel von R_1 oder undefiniert sein. R_2 ist in diesem Fall eine referentielle Relation, während R_1 hier als referenzierte Relation bezeichnet wird. Die Attribute FK und PK werden referentielle Attribute genannt. Für eine referenzierte Relation kann es mehrere referenzierende Relationen geben. Ebenso kann eine referentielle Bedingung mehrere referentielle Attribute umfassen. Eine referentielle Bedingung läßt sich folgendermaßen darstellen:

$$RX : R_2.Y \rightarrow R_1.X$$

In diesem Fall wird also die referentielle Bedingung mit RX bezeichnet. Y ist der fremde Key von R_2 und X ist der primäre Key von R_1 . Gelesen wird diese Bedingung wie folgt:

„Ist ein R_2 -Tupel mit dem Wert Y für den fremden Key Y gegeben, ist auch ein R_1 -Tupel mit dem Wert Y für den primären Key X vorhanden“.

Die Implementierung der referentiellen Bedingungen hat sich als überaus schwierig erwiesen /1, 2, 6, 8, 9, 10, 12, 13, 14/. Dies ist vor allem darauf zurückzuführen, daß bei referentiellen Bedingungen, anders als bei Wertebedingungen, die Attributwerte von zwei Relationen verglichen werden, um zu überprüfen, ob der Wert des fremden Keys der referenzierenden Relation der Wert des Keys in der referentiellen Relation ist. Zur Feststellung der Konsistenz eines Datenbankzustandes ist es daher erforderlich, eine Vielzahl von Daten zu überprüfen. Der Prozeß ist also sehr zeitaufwendig, was auch der Grund dafür ist, daß bei vielen kommerziellen DBMS, einschließlich INGRES, die referentiellen Bedingungen nicht automatisch realisiert werden. In Expertensystemen wie ELAN ist dafür eine spezielle Software, beispielsweise der hier beschriebene Konsistenzmonitor, erforderlich.

Die Erfüllung referentieller Bedingungen mit Hilfe des Konsistenzmonitors kann auf zwei unterschiedliche Arten erfolgen: Bei der ersten wird der Konsistenzmonitor als Front-end des DBMS eingesetzt. Der Ablauf der Datenbankoperation wird kontrolliert und die Konsistenz des Datenbankzustandes wird mit Hilfe von sogenannten 'Triggern' und 'Demons' realisiert /4, 15/. Auf diese Weise wird zwar für eine ständige Konsistenz der Datenbank gesorgt, der zusätzliche Platzbedarf ist jedoch sehr hoch /7, 11/. Beim zweiten Verfahren wird die referentielle Bedingung asynchron realisiert /3/. Dabei werden Update-Anomalien, die die referentiellen Bedingungen nicht erfüllen, nicht blockiert oder vermieden. In bestimmten Intervallen wird von dem Konsistenzmonitor eine komplette

“Säuberung” der Datenbank durchgeführt, wobei die Benutzer über Inkonsistenzen informiert und die referentiellen Bedingungen erfüllt werden. Leider ist auch dieses Säubern der gesamten Datenbanken sehr zeitaufwendig /7, 11/.

Der für das ELAN-System entwickelte Konsistenzmonitor basiert auf dem zweiten Prinzip, wobei allerdings auf das Säubern der gesamten Datenbank verzichtet wird. Dieses ist nur möglich, weil die Datenbankkonsistenz im ELAN-System nicht ständig benötigt wird. Die Konsistenz der Datenbank ist lediglich die Voraussetzung für die Herstellung der Konsistenz zwischen Datenbank und Wissensbasis. Um eine Säuberung der gesamten Datenbank zu vermeiden, wird das INGRES-Journaling-System verwendet.

Ein Journal hat grundsätzlich zwei Aufgaben: Verwaltung der Sicherungskopie zur Vermeidung des Datenverlustes sowie Protokollieren von Transaktionen. Alle Update-Operationen - Einfügen, Löschen, Updating -, die im Rahmen der Option WITH JOURNALING an Relationen vorgenommen werden, werden also automatisch in der Journaldatei protokolliert. Die beiden Systemkommandos CKPDB und AUDITDB ermöglichen die Manipulation des Journals durch die Benutzer. Mit Hilfe des Kommandos CKPDB wird ein neuer Checkpoint für die Datenbank festgelegt. Alle Journaleinträge vor diesem Checkpoint werden zerstört. Mit Hilfe des Kommandos AUDITDB kann das Datenbankjournal beginnend beim letzten Checkpoint von dem Benutzer gelesen werden. Darüberhinaus können die an einer bestimmten Relation vorgenommenen Änderungen mit einem lesbaren Prüfpfad dargestellt werden und sind damit nachvollziehbar. Der Pfad wird in einer Datei erstellt und kann mit Hilfe des Kommandos COPY auf eine INGRES-Relation übertragen werden. Diese wird dann auch als 'audit relation' bezeichnet. Die 'audit relation' enthält alle Informationen über die nach dem letzten Fixpunkt an einer bestimmten Relation vorgenommenen Änderungen. Die 'audit relations' enthalten damit also alle zur Realisierung der Konsistenz durch den Konsistenzmonitor erforderlichen Daten. Das Verarbeiten der 'audit relations' ist dabei sehr viel effizienter als das Säubern der gesamten Datenbank.

Im folgenden soll nun die Funktionsweise des Konsistenzmonitors beschrieben werden.

Einfügen eines Tupels

Das Einfügen eines Tupels in eine referenzierte Relation wird von dem Konsistenzmonitor ignoriert, da das Übertragen auf die referenzierende Relation zu Unsicherheiten führen kann. Bei einer Bedingung $RX: R_2.Y \rightarrow R_1.X$ wird demnach lediglich der Wert des Attributs Y von der Einfügung in R_1 bestimmt.

Läßt also ein bestimmtes Attribut Z von R₂ keinen Nullwert zu, müßte bei der Übertragung der Einfügung ein bestimmter Nichtnullwert für Z festgesetzt werden, was jedoch kaum sinnvoll ist. INGRES DBMS läßt in Attributen, die Relationskeys oder Teile davon sind, keine Nullwerte zu.

Löschen eines Tupels

Der Konsistenzmonitor ignoriert das Löschen eines Tupels aus einer referenzierenden Relation. Wird jedoch ein Tupel aus einer referenzierten Relation gelöscht, wird dies von dem Konsistenzmonitor erfaßt und auf die referenzierende Relation übertragen.

Updating eines Tupels

Das Updating eines nicht-referentiellen Attributs einer referenzierten oder referenzierenden Relation wird von dem Konsistenzmonitor ignoriert. Dies gilt ebenfalls für das Updating eines referentiellen Attributs einer referenzierenden Relation. Wird im Gegensatz dazu ein referentielles Attribut einer referenzierten Relation dem Updating unterzogen, wird dies von dem Konsistenzmonitor auf die referenzierende Relation übertragen.

Zur Festlegung der referenziellen Bedingungen werden ähnliche Tabellen wie in /5/ verwendet. Sie enthalten Informationen über referenzierende und referenzierte Relationen sowie über referentielle Attribute und deren Typen.

3 Realisierung der Konsistenz von Datenbank und Wissensbasis

Informationen werden in der Datenbank in Form von Relationen und in der Wissensbasis in Form von Tatsachenmengen (sets of facts) gespeichert. Bei den doppelt vorhandenen Informationen handelt es sich einerseits um bestimmte Tatsachenmengen und andererseits um Relationen, die aus der Bewertung bestimmter relationaler Ausdrücke über eine Datenbankinstanz abgeleitet wurde. Diese Ausdrücke, die auch als 'expressions' bezeichnet werden, sind Kombinationen der Operationen project, select, join und group by. Eine 'expression' kann folgendermaßen dargestellt werden:

$$E = \pi_A \sigma_C (R_{i1} \Psi R_{i2} \Psi \dots \Psi R_{iK})$$

Hier bezeichnen R_{i1}, R_{i2}, ..., R_{iK} die Relationen, π den Operator project, σ den Operator select bzw. group by und Ψ den Operator join. C ist eine Auswahlbedingung und A = A₁, ..., A_L ist eine Menge von Attributen dieser Projektion.

Die Attributmenge A entspricht der Menge von Feldern einer Tatsache (facts) in der Wissensbasis.

Zur Veranschaulichung betrachten wir folgendes Beispiel. Es sei angenommen, daß die Wissensbasis eine Tatsachenmenge KB__fact enthält. Dabei hat jede Tatsache die Form:

KB__fact (cleaning__period, box__type, device__type)

Hier bezeichnen cleaning__period, box__type und device__type die Tatsachenfelder. Es sei weiter angenommen, daß die Tatsachenmenge KB__fact der über den folgenden Relationsmengen R₁, R₂ und R₃ abgeleiteten Relation entspricht.

R₁(Box#, device#, cleaning__period)

R₂(box#, box__type)

R₃(device#, device__type)

Der Teil der Datenbank, der die Relationen R₁, R₂ und R₃ enthält, verfügt über die in Abb. 2 dargestellte Struktur.

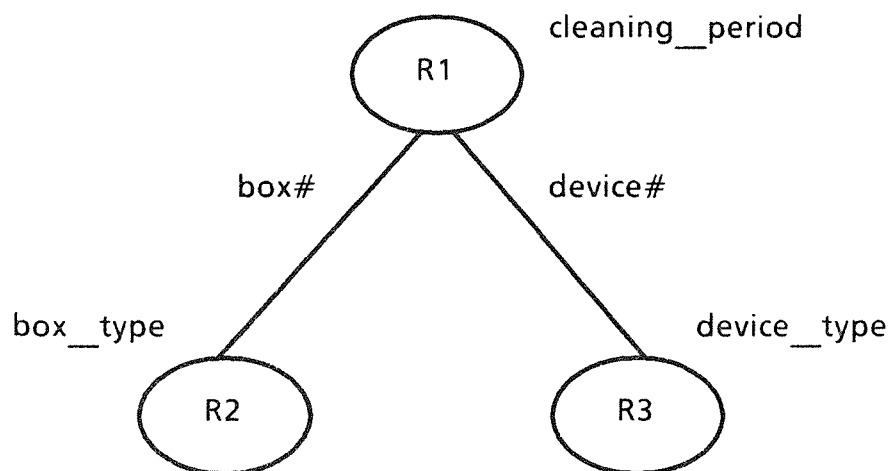


Abb. 2: Beziehungen zwischen R₁, R₂ und R₃

Der relationale algebraische Ausdruck E hat die Form:

$E = \pi \{ \text{cleaning_period, box_type, device_type} \} (R_1 \psi R_2 \psi R_3)$

Dieser Ausdruck entspricht dem folgenden QUEL-Kommando in INGRES:

```
Retrieve (R1.cleaning__period, R2.box__type, R3.device__type)
where R1.box# = R2.box#
and R1.device# = R3.device#
```

Bei der Ausführung dieses Kommandos für die in Abb. 3 gezeigte Datenbankinstanz erhält man abgeleitete Relationen, deren Tupel der folgenden Tatsachenmenge entsprechen:

- KB__fact (50, entry__box, spectrometer)
- KB__fact (60, entry__box, thermometer)
- KB__fact (04, gamma__box, thermometer)
- KB__fact (16, cleaning__box, thermometer)
- KB__fact (18, cleaning__box, ph-meter)

Eine Änderung der Datenbankrelationen, deren Attribute den Tatsachenfeldern in der Wissensbasis entsprechen, kann, muß jedoch nicht, zum Erzeugen bzw. Löschen von Tatsachen führen. Wird beispielsweise ein neues Tupel (B₄, new__box) in die Relation R₂ eingebracht (s. Abb. 3), wird keine neue Tatsache erzeugt, da die Information unvollständig ist: KB__fact (?, new__box, ?). Eine neue Tatsache wird erzeugt, wenn auch bei R₁ ein Updating, z.B. durch Hinzufügen eines Tupels (B₄, D₃, 45) vorgenommen wird. Dann lautet die neue Tatsache: KB__fact (45, new__box, ph__meter).

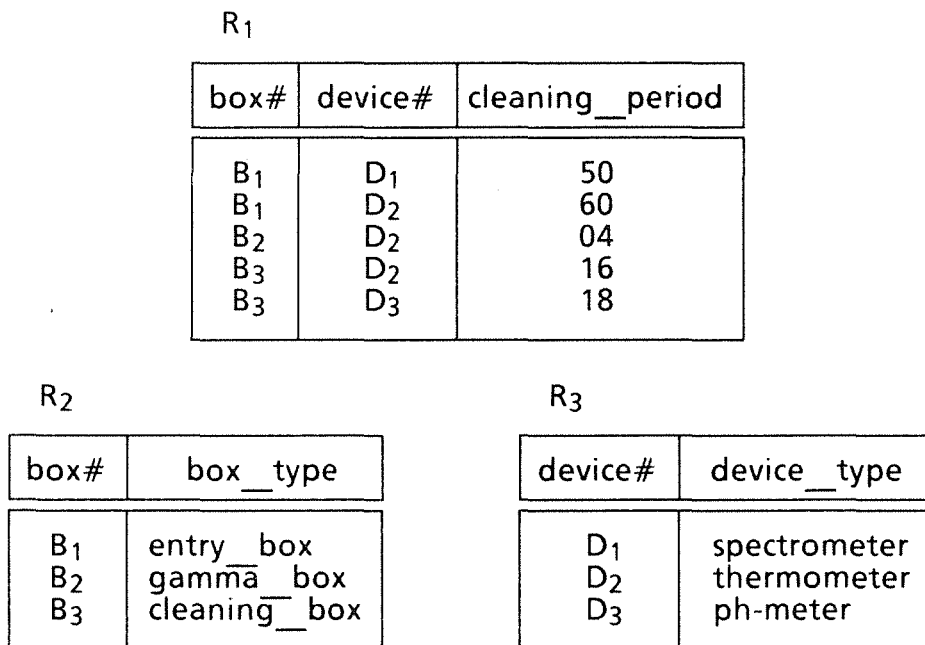


Abb. 3: Datenbankinstanz

Zur Sicherstellung von Konsistenz zwischen den doppelt vorhandenen Teilen der Datenbank und der Wissensbasis, werden alle an der Datenbank vorgenommenen Änderungen, die zur Erzeugung oder zum Löschen von Tatsachen geführt

haben, von dem Konsistenzmonitor auf die Wissensbasis übertragen. Diese Änderungen werden mit Hilfe der folgenden Methode bestimmt (Differencing).

Die Tatsachenmengen, die die doppelt vorhandenen Informationen enthalten, werden mit Fact₁, Fact₂, ..., Fact_i, ..., Fact_m bezeichnet. Eingeführt werden nun drei mit diesen Tatsachenmengen verbundene Relationen. Die Relationen Fact₁_new, Fact₂_new, ..., Fact_i_new, ..., Fact_m_new stellen den aktuellen Zustand der Datenbank dar. Die Relationen Fact₁_dif, Fact₂_dif, ..., Fact_i_dif, ..., Fact_m_dif beschreiben den Unterschied zwischen dem neuen und dem alten Zustand der Wissensbasis. Das heißt, sie enthalten Tupel, die zum Update mittels Erzeugen oder Löschen von Tatsachen zur Wissensbasis geschickt werden müssen. Zur Unterscheidung von Erzeugen und Löschen wird ein spezielles Attribut, das als 'operation' bezeichnet wird, verwendet. Dieses Attribut kann zwei Werte besitzen: "+" oder "-". Alle oben genannten Relationen werden nur für den Konsistenzmonitor benötigt. Für die Benutzer sind sie unzugänglich.

Um zu überprüfen, ob Datenbankänderungen zu Änderungen in der Wissensbasis führen, und auch zur Herstellung der Konsistenz von Datenbank und Wissensbasis, wird folgender Algorithmus verwendet:

Für jede Tatsachenmenge Fact_i, $i=1, \dots, m$, sind folgende Schritte durchzuführen:

1. Ableitung der Relation Fact_i_new.
2. Festlegung des Inhalts von Relation Fact_i_dif in zwei Schritten (\setminus bezeichnet den Differenzoperator):
 - * Hinzufügen aller Tupel der Menge Fact_i_new \setminus Fact_i_old zu Fact_i_dif. Das Attribut operation hat den Wert "+". (Tatsachen, die diesen Tupeln entsprechen, müssen in die Wissensbasis eingefügt werden.)
 - * Hinzufügen aller Tupel von der Menge Fact_i_old \setminus Fact_i_new zu Fact_i_dif. Das Attribut operation hat den Wert "-". (Tatsachen, die diesen Tupeln entsprechen, müssen aus der Wissensbasis gelöscht werden.)
3. Senden von Fact_i_dif zur Wissensbasis zum Aktualisieren.
4. Speichern des neuen Zustands der Wissensbasis: Fact_i_old \leftarrow Fact_i_new.
5. Löschen der Relationen Fact_i_dif und Fact_i_new.

4. Schlußfolgerungen

Das Hauptproblem bei der Gewährleistung der funktionalen Korrektheit des Expertensystems ELAN ist die Konsistenz der beiden Komponenten Datenbank und Wissensbasis. Traditionelle Verfahren, bei denen ein Durchsuchen der gesamten Datenbank erfolgt, sind hier ungeeignet, da die zur Herstellung der Konsistenz von Datenbank und Wissensbasis benötigte Zeit den Zeitraum für die Erstellung des Analysenplans durch den Schichtführer überschreitet. Durch die Verwendung der DBMS-Journals ist ein Durchlauf durch die gesamte Datenbank nicht mehr erforderlich. Die Leistungsfähigkeit des Konsistenzmonitors wurde somit radikal verbessert.

Die Effizienz des Konsistenzmonitors kann auch auf der Implementierungsebene verbessert werden. Mit Hilfe des Kommandos AUDITDB wird ein lesbare Prüfpfad erstellt, mit dem die an einer Relation vorgenommenen Änderungen nachvollzogen werden können. In dem Konsistenzmonitor muß dieses Kommando daher so häufig ausgeführt werden, wie Relationen zu kontrollieren sind. Zur Ausführung des AUDITDB-Kommandos ist die INGRES-Datenbank zu verlassen. Dagegen ist zur Ausführung des Kommandos COPY (Erzeugung einer 'audit relation') ein Wiedereintritt in die Datenbank INGRES nötig. Der gesamte Prozeß des Verlassens der Datenbank und des Wiedereintritts ist sehr zeitaufwendig. Vermeiden läßt sich dies, wenn das Kommando AUDITDB so neu geschrieben wird, daß bereits bei einer Ausführung die Änderungen an allen kontrollierten Relationen in einem lesbaren Prüfpfad dargestellt werden.

5. Literatur

- 1) Badal, D., Popek, G., *Cost Performance Analysis of Semantic Integrity Validation Methods*, Proc. SIGMOD Conf. on Management of Data, Austin, 1979, pp. 109-115
- 2) Bernstein, P.A., Blaustein, B.T., *Fast Method for testing Quantified Relational Calculus Assertions*, Proc. SIGMOD Conf. on Management of Data, Orlando, 1982, pp. 39-50
- 3) Cammarata, S., Ramachandra, P., Shane, D., *Extending a Relational Database with Deferred Referential Integrity Checking and Intelligent Joins*, Proc. SIGMOD Conf. on Management of Data, Austin, 1989, pp. 88-97
- 4) Casanova, M.A., Tucheran, L., Furtado, A.L., *Enforcing Inclusion Dependencies and Referential Integrity*, Proc. 14th VLDB Conf., Los Angeles, 1988, pp. 38-49
- 5) Date, C.J., *Referential Integrity*, Proc. 7th VLDB Conf., Cannes (France), 1981, pp. 2-12
- 6) Furtado, A., dos Santos, D., de Castilho, J., *Dynamic Modeling of a Simple Existence Constraints*, Information Systems 6, 1981, pp. 73-80
- 7) Hatonn, T., *Deferred vs Immediate Checking for Consistency of Database*, Management Information System Week (June 1988)
- 8) Hsu, A., Imielinski, *Integrity Checking for Multiple Updates*, Proc. SIGMOD Conf. on Management Data, Austin, 1985, pp. 152-168
- 9) Hammer, M., Sarin, S., *Efficient Monitoring of Database Assertions*, Proc. SIGMOD Conf. on Management of Data, Austin, 1978, pp. 38-49
- 10) Kobayashi, I., *Validating Database Updates*, Information Systems 9, 1, 1986, pp. 1-17
- 11) Lafne, G.M.F., *Semantic Integrity Dependencies and Delayed Integrity Checking*, Proc. 8th VLDB Conf. Mexico, 1982, pp. 292-299
- 12) Qian, X., Smith, D.R., *Integrity Constraint Reformulation for Efficient Validation*, Proc. 13 VLDB Conf., Brighton, 1987, pp. 417-425
- 13) Stonebraker, M., *Implementation of Integrity Constraints and Views by Query Modification*, Proc. SIGMOD Conf. on Management of Data, San José, 1985, pp. 65-78

- 14) Stonebraker, M., Rowe, L., *The Design of POSTGRES*, Proc. SIGMOD Conf. on Management of Data, Washington, 1986, pp. 340-354