First publ. in: Rundbrief des Fachausschusses 5.2 Informationssystems-Architekturen 5(1)

Zeitorientierte Informationssysteme auf Basis objektorientierter Datenbanken — Möglichkeiten und Grenzen

Holger Riedel*

Zusammenfassung

Zeitorientierte Informationssysteme erfordern Leistungskenngrößen von den zugrundeliegenden Datenbanksystemen, die relationale DBMS nicht zur Verfügung stellen können. Obwohl objektorientierte DBMS weitergehende Modellierungskonzepte anbieten, scheint auch hier eine orthogonale und leistungsfähige Integration von Zeitdaten zumindest schwierig zu sein. In diesem Papier werden diese Probleme eingehend analysiert und eine objektorientierte Datenbankarchitektur für Zeitdaten vorgestellt.

1 Realisierung zeitorientierter Informationssysteme

Zeitbezogene Daten werfen spezielle Problemstellungen auf, die gegenwärtig beim Entwurf adäquater DV-Systeme nur unbefriedigend berücksichtigt werden können. Insbesondere bei Informationssystemen, wo ein wesentlicher Teil aus den zu verarbeitenden Datenmengen besteht, fehlt es an geeigneten Modellierungswerkzeugen und leistungsfähiger Datenbanktechnologie. Dies hat zur Konsequenz, daß gegenwärtig Zeitabhängigkeiten in Datenbeständen mit traditionellen Methoden modelliert und später auf relationale Datenbanken abgebildet werden. Temporale Datenbankkonzepte bieten hier eine bessere Unterstützung, da hier spezielle zeitorientierte Problematiken (wie die adäquate Verwendung von Kalendern oder die Berücksichtigung von verschiedenen Zeitdimensionen und Granularitäten) explizit konzeptuell modelliert werden und nicht als Implementierungsdetail im Design der relationalen Datenbank versteckt werden. Da der TSQL2-Vorschlag eine umfassende Darstellung temporaler Konzepte enthält, gibt es mittlerweile erste Informationssysteme, die dies bei der Abbildung auf relationale Datenbanken berücksichtigen [BS97].
Die Verwendung objektorientierter Modellierungswerkzeuge beim Entwurf von Informationssystemen wirft insbesondere die Frage auf, inwieweit die Abbildung auf relationale Datenbanken adäquat ist bzw. welche Möglichkeiten hier objektorientierte Datenbanken bieten. Dabei kann festgestellt werden:

- Die Abbildung von Konzepten der objektorientierten Analyse auf relationale Datenbanken ist insbesondere dann problematisch, wenn Konzepte verwendet werden, die nicht mit relationalen Konzepten (Schlüssel, Fremdschlüssel) ausgedrückt werden können, da dies zu einer ineffizienten Normalisierung führen kann (bei Verwendung von Aggregation/Assoziation und Komponentenobjekten) und die Integritätsbedingungen nicht automatisch nachgebildet werden können.
- Die gegenwärtige OODBMS-Technologie kann zur Zeit nur bedingt bei der Realisierung von Informationssystemen eingesetzt werden, da mangels eines allgemeinen Standards für Objektdatenbanken zur Zeit nur für bestimmte OOODBMSe spezialisierte Entwurfswerkzeuge existieren.
- Es besteht die berechtigte Hoffnung, daß durch die Etablierung des ODMG-Standards [Cat96] viele OODBMSe durch dieselbe Schnittstellenbeschreibung modelliert werden können, so daß objektorientierte Modellierungswerkzeuge (etwa auf Basis von UML) übergreifend eingesetzt werden können.

Ein wesentliches Problem für die Realisierung von zeitorientierten Informationssystemen ist die Nichtverfügbarkeit von temporalen Datenbankmanagementsystemen, da die bisher entwickelten Konzepte für temporale Datenbanken nur teilweise in Forschungsprototypen realisiert worden sind.

*Fakultät für Mathematik und Informatik, Universität Konstanz, 78457 Konstanz;
Email: Holger.Riedel@uni-konstanz.de

2 Einsatz von objektorientierten Datenbanken

In [Sno95a] wird festgestellt, daß zur Zeit kein temporales Objektdatenbankmodell mit einer zugehörigen Anfragesprache gibt, das eine formale Grundlage hat. Dagegen hat sich die Verwendung eines formalen Modells bei relationalen Datenbanken bewährt. So können die Mächtigkeit von Anfragen oder die Umsetzung in die relationale Algebra exakt charakterisiert werden, so daß weitergehende Analysen von konzeptuellen und implementationstechnischen Details davon profitiert haben. Wir werden in diesem Abschnitt untersuchen, welche formalen Werkzeuge im Umfeld temporaler Objektdatenbanken zur Verfügung stehen.

2.1 Relationale Konzepte

Auch die Erweiterung des relationalen Modells um temporale Konzepte wirft einige Fragen auf:

- Bei der Verwendung von attributorientierten Zeitstempeln wird die erste Normalform verlassen. Dies führt dazu, daß große Teile der Begriffsbildung für relationale Grundkonzepte neu konzipiert werden müssen, so daß die gegenwärtig verfügbaren Modelle sich weitgehend vom Relationenmodell unterscheiden.
- Bei der Abbildung von Entity-Integritätsbedingungen werden zumeist nur Beziehungen berücksichtigt, die sich auf die Tupel einer Relation beziehen. So erfordert die Homogenität [GN93], daß alle Attribute eines Tupels über die gleiche Zeitspanne definiert sind, während beim *Coalescing* Tupel zusammengefaßt werden, die auf ihren (nichttemporalen) Attributen übereinstimmen.
- Die Unterstützung von variablen Konstanten (wie *now* oder *until changed*) erfordert eine weitergehende Analyse bei der Interpretation der Datenbank, die erst in aktuellen Arbeiten ausführlich analysiert wurde [CDI+97].

Da die Berücksichtigung aller Feinheiten zu aufwendig ist, scheint das BCDM [Sno95b] eine realistischer Kompromiß zur Realisierung von temporalen Datenmodellen auf Basis des Relationenmodells zu sein.
Bei der Entwicklung von adäquaten Anfragesprachen kann beobachtet werden, daß die meisten nur (informale) Erweiterungen von SQL sind, während eine formale Grundlage nur in wenigen Sprachen vorhanden ist (etwa für TQUEL [Sno93]). Eine weitgehende Kompatibilität zu SQL2 bietet TSQL2, wofür auch eine adäquate Algebra vorgeschlagen wurde. Es bleibt festzustellen, daß zur Erreichung einer adäquaten Mächtigkeit dieser Anfragesprachen vielfältige Erweiterungen notwendig waren, so daß eine detaillierte Analyse von Details, wie sie im relationalen Modell möglich war, nur in Ansätzen möglich sein wird.

2.2 Objektdatenmodelle

Objektorientierte Datenbankmodelle stellen höhere Anforderungen an die Modellierungskonzepte als das relationale Modell, da durch eine adäquate Modellierung der Anwendung die weitere Anwendungsentwicklung vereinfacht werden soll. Dies führt insbesondere dazu, daß die Modellierung der Integrität der zu verarbeitenden Daten einen wesentlichen Schwerpunkt eines Objektmodells ausmacht. Da die grundlegenden Begriffsbildungen (etwa: Was ist ein Objekt? Welche Strukturelemente kann ein Objekt haben? Welche Beziehungen zwischen Objekten sollen explizit modelliert werden können?) nicht detailliert geklärt worden sind, unterscheiden sich die diversen Objektbankmodelle von Forschungs- und kommerziellen OODBMSen im Detail erheblich. So können in einem Smalltalk-basiertem OODBMS wie Gemstone Objekte unabhängig von ihrem Typ behandelt werden, während O2 sowohl eine Typ-, wie auch eine Klassenhierarchie unterstützt. Dagegen verwendet der ODMG-Standard [Cat96] eine Typhierarchie, getrennt für Objekte und Werte, die jeweils strukturiert sein können.
Dies hat für die Entwicklung geeigneter temporaler objektorientierter Datenbankmodelle (TOODM) die Konsequenz, daß bereits ein gemeinsamer Startpunkt fehlt. Während in der Vergangenheit auf unterschiedlichen objektorientierten Datenbankmodellen (OODM) temporale Konzepte aufgesetzt worden sind (ein Überblick findet sich in [Sno95a]), wird man in Zukunft versuchen, möglichst nah am ODMG-Standard zu bleiben, zumal dann die berechtigte Hoffnung besteht, daß eine Integration mit dem SQL3-Standard möglich ist.
Allerdings birgt die Entwicklung von TOODMen auf Basis von OODMen auch Vorteile. Die Unterstützung von Zeitstempeln für komplexe Attribute kann durch die Verwendung geeigneter Typkonstruktoren unterstützt werden. So enthält das T_Chimera-Modell [BFG96] einen Typkonstruktor `temporal()`, mit dem beliebige (Valid-time)-Typstrukturen aufgebaut werden können.

Ein wesentliches Problem für TOODMe ist die Integrität von Objekten, die an verschiedenen Stellen der Objekthierarchie auftreten, bzw. die Beziehungen zu anderen Objekten haben. So fordert das T_Chimera-Modell, daß der Subtyp eines temporalen Typs selber ein temporaler Typ ist. Dies hat weitgehende Konsequenzen für die Modellierung der Typhierarchie. Wenn z.B. ein *Angestellter* mit seiner Valid-Time-Historie modelliert wird, muß auch die Untertyp *Koch* temporal modelliert werden, wenn auch aus der Anwendung keine zeitabhängigen Informationen für diese Klasse vorliegen. Ebenso verhält es sich mit der Referenz auf andere Objekte, die ebenfalls temporal modelliert werden müssen, wenn das referenzierte Objekt temporal ist. Zur Zeit ist die Behandlung von Datenbankschemata, die nur partiell zeitabhängige Daten enthalten, noch nicht befriedigend gelöst. Ein Versuch, die weitreichenden Vorgaben an die Modellierung zu umgehen, wird in [Rie96] beschrieben, wobei durch die Unterscheidung von gespeicherter und intensionaler Sicht auf die Datenbank eine konsistente Sicht auf eine partiell-temporal modellierte Datenbank ermöglicht wird.

2.3 Anfragesprachen

Für die Entwicklung von objektorientierten Anfragesprachen (OOQL) gilt das gleiche wie für OODMe. Aufgrund der unterschiedlichen Startvoraussetzungen sind die Prinzipien für gute Anfragesprachen [HS91] in verschiedener Weise umgesetzt worden. Dabei muß festgehalten werden, daß es zu einigen OODBMSen keine oder nur sehr einfache Anfragesprachen gibt, da zum Teil die Meinung vertreten wird, daß bei der Entwicklung von Anwendungssystemen auf Basis von OODBMS die geradlinige Umsetzung von OOA/OOD-Methoden in OO-Programmcode im Vordergrund steht. Desweiteren konnten im Vergleich zu relationalen Anfragesprachen folgende Prinzipien nicht oder nur unvollständig erreicht werden:

- *Einfachheit:* Während das NF^2-Modell noch mit zwei zusätzlichen Operatoren auskam, haben die Anfragesprachen für umfassendere Objektbankmodelle eine Vielzahl von syntaktischen Konstrukten, die den variablen Zugriff und die Konstruktion strukturierter Objekte unterstützen. Etwa hat ODMG-OQL mehr als 30 verschiedene Klauseln.
- *Die Formalisierung* fehlt oft und ist auch für den ODMG-Vorschlag nur in Forschungsarbeiten [RS97, FM95] untersucht worden.
- *Sicherheit:* Dieses Grundkonzept relationaler Sprachen wurde für ODMG-OQL aufgehoben, um Anfragen mit Laufzeitfehlern zuzulassen, die etwa durch auftretende Typisierungsfehler oder durch die Verwendung von Methodenaufrufen verursacht werden.
- *Einordnung von Anfrageergebnissen:* Wenn eine Anfrage Objekte zurückliefert, ist es sinnvoll, diese Objekte gemäß der Typstruktur klassifizieren zu können. Dies ermöglicht in darauf aufbauenden Anfragen einen korrekten Typcheck.

Dies zeigt, daß eine Kombination von temporalen und objektorientierten Anfragesprachen eine große Anzahl von Detailproblemen aufwerfen, die aus temporaler Sicht in [Sno95a] analysiert werden. Aus der Sicht objektorientierter Anfragesprachen bleibt folgendes festzuhalten:

- Die Beschreibung von temporalen Konzepten durch Zugriffsmethoden von OOQLs ist weitreichend möglich, da OOQLs den Zugriff und die Konstruktion von strukturierten Daten direkt unterstützen. So kann etwa ein Valid-Time-Attribut wie ein mengenwertiges Attribut behandelt werden.
- Die meisten Detailkonzepte von temporalen und objektorientierten Anfragesprachen sind orthogonal zueinander, so daß eine Integration prinzipiell möglich ist.
- Die vollständige Formalisierung einer umfassenden Anfragesprache ist zumindest technisch sehr aufwendig wird, zumal dies für wesentliche Vorarbeiten wie ODMG-OQL oder TSQL2 ebenfalls nur in Ansätzen vorliegt.

2.4 Implementierung

Realisierungen von den bisher beschriebenen Konzepten sind bisher nur in Ansätzen vorhanden. In kommerziellen OODBMSen werden Objekte zumeist zusammenhängend gespeichert. Dies ist für die Realisierung temporaler Anfragen zumindest sehr problematisch, da bei dem Retrieval jeweils die komplette Historie zurückgeliefert wird. Die effiziente Anfrageauswertung wird aber auch für OODBMS fast ausschließlich in der Forschung untersucht, während in kommerziellen OODBMSen besonders die

anwendungsprogramm-orientierte Transaktionsabarbeitung optimiert wird. So sind die notwendigen Voraussetzungen, wie die Abbildung der Anfrage auf formale Strukturen und deren Optimierung durch Umformungsregeln nur in Ansätzen erforscht worden. Ein wesentliches Problem ist die für die Auswertung von Standardanfragen notwendige Flexibilität der Speicherung, die es ermöglicht, abhängig von einer vorgegebenen Transaktionslast, eine optimale Speicherung anzugeben. Dies erfordert insbesondere die Fragmentierung von komplexen Objekten, um ggf. größere Teile der Objektstrukturen bei der Auswertung von Anfragen ausblenden zu können.

3 Eine Architektur für temporale Objektdatenbanksysteme

Hier beschreiben wir die Architektur für ein temporales OODBMS, das folgende wesentlichen Leistungsmerkmale umfaßt:

- Erweiterung des ODMG-Objektmodells um temporale Konzepte,
- Erweiterung der ODMG-OQL um Konzepte aus TSQL2,
- Realisierung auf Basis des OODBMS CROQUE, das eine flexible Speicherung von Objekten und die Erweiterung um weitergehende Konzepte ermöglicht.

Erweiterbarkeit. Wegen der im vorhergehenden Kapitel geschilderten Probleme bei der Integration von relationalen, objektorientierten und temporalen Konzepten verfolgen wir die Strategie, daß ausgehend von einem Grunddatenbankmodell eine adäquate Anfragesprache konzipiert wird. Das Fernziel ist die vollständige Integration des ODMG-Datenmodells und ODMG-OQL mit den Konzepten von TSQL2. Dieselbe Vorgehensweise verfolgen wir bei der Implementierung. Durch eine schrittweise Erweiterung des CROQUE-Speichermanagers um temporale Konzepte kann die Leistungsfähigkeit dem gegenwärtigen Stand der konzeptuellen Ebene angepaßt werden.

Das TCROQUE-Datenmodell ist eine Erweiterung des CROQUE-Modells [RS97] um temporale Konzepte. Dabei wurde versucht, möglichst die Ideen des BCDMs zu berücksichtigen, wobei im Konfliktfall aber den Vorgaben des Objektmodells Vorrang eingeräumt wurde. Als zusätzliche Grunddatentypen stehen die TSQL2-Datentypen *Datetime*, *Interval* und *Period* zur Verfügung, auf die ggf. die Aggregatsfunktionen `count`, `min`, `max`, `sum` und `avg` angewendet werden können.
Es werden die von der ODMG vorgeschlagenen Typkonstruktoren `set`, `bag`, `list`, `array` und `struct` unterstützt. Zusätzlich gibt es die Typkonstruktoren `transaction` und `valid`, die auf beliebige Typen angewandt werden können. Dies ermöglicht die flexible Modellierung von temporalen und nichttemporalen Attribute eines Objekts. So beschreibt

```
struct<Ang-Nr:string, Gehaltshistorie:valid<Gehalt:string>,
   Qualfikationen:set<Qualifikation:transaction<valid<string>>>>
```

einen Tupeltyp, bei dessen Komponente *Gehaltshistorie* die Valid-Time gespeichert wird, während für jede einzelne Qualifikation die bitemporale Zeitinformation gespeichert wird. Bei Objekttypen wird zusätzlich noch unterschieden, ob in dem Extent auch die Lebensdauer des Objekts (unabhängig von seinen Attributwerten) gespeichert wird. Diese beiden Konzepte sind orthogonal zueinander im TCROQUE-Modell, d.h. bei der Objekttypdefinition

```
interface Person{
   (extent Personen)
     Name:string,
     Kinder:set<Person>};

interface Angestellte:Person{
   (extent Angestellten as valid and transaction)
     Ang-Nr:string,
     Gehaltshistorie:valid<Gehalt:string>,
     Qualfikationen:set<Qualifikation:transaction<valid<string>>>};
```

wird für jede Person der Zeitraum, wo sie Angestellte ist, bitemporal gespeichert. Dies ermöglicht eine flexible Kombination von Zustandseigenschaften eines Objekts. So sind in diesem Beispiel u.a. der *Name*, die *Ang-Nr* und die Zugehörigkeit zum Extent *Personen* statisch, während die Attribute *Kinder* und *Gehaltshistorie* valid-time sind. Die Datentypen `valid<`*T*`>` und `transaction<`*T*`>` sind jeweils Funktionen von *Datetime* nach *T*, d.h. im TCROQUE-Modell werden stets zeitpunktabhängige Daten beschrieben. Zur Vereinfachung können auch Zeitintervalle oder -elemente in Anfragen oder bei der Visualisierung von Daten verwendet werden.
Die im letzten Kapitel aufgezeigten Probleme bei der Interpretation von partiell-temporalen Datenbanken werden im TCROQUE-Modell wie folgt gelöst: Die Homogenität wird aufgegeben, so daß prinzipiell unterschiedliche Zeitangaben für die Attribute eines Objekts möglich sind. So ist jede typkorrekte Instanz des Objektschemas eine korrekte Objektinstanz. Da auch (ggf. komplex strukturierte) Werte ohne Objektidentität im Datenbankschema modelliert werden können, gilt dies auch für i.a. für temporale Relationen im TCROQUE-Modell. Allerdings kann die Homogenität durch geeignete Typstrukturen simuliert werden. So beschreibt

```
set<valid<struct<Ang-Nr:string, Gehalt:int>>>
```

eine homogene temporale Relation, da die Zeitmarken für jedes zusammenhängende Tupel vergeben werden. *Coalescing* wird im TCROQUE-Modell nur durch die Integritätsforderungen an Objekte unterstützt. Da alle Attribute als globale Funktionen modelliert sind, und die Typkonstruktoren `valid<>` und `transaction<>` ebenfalls Funktionen darstellen, kann ein Objekt nicht mehrere Zustände zu einem Zeitpunkt annehmen. Diese Forderung greift natürlich nicht bei Werten, da diese ohne die Verwendung von temporal-funktionalen Abhängigkeiten [Wij95, WBB97] beliebig verwendet werden können. Im TCROQUE-Modell können die variablen Zeitkonstanten *now* und *forever* (für `valid<>`), bzw. *until changed* (für `transaction<>`) verwendet werden, wobei die Semantik wie in [CDI+97] definiert wird.
Zukünftig soll das TCROQUE-Modell an folgenden Stellen erweitert werden:

- Auf nichttemporale und partiell temporale Daten soll eine bitemporale Interpretation möglich sein, die eine zu jedem Zeitpunkt konsistente Sicht auf die Daten ermöglicht. Ein erster Ansatz befindet sich hierzu in [Rie96].
- Es sollen Integritätsbedingungen berücksichtigt werden, sowohl statische, wie auch dynamische in einer entsprechenden temporalen Logik.

Die Anfragesprache TCROQUE-TOQL umfaßt möglichst weitgehend alle Konzepte aus TSQL2 und ODMG-OQL. Eingeschränkt auf OQL können alle Standardanfragen an temporale und nichttemporale Daten gestellt werden. So liefert

$Q_1 \equiv$
```
select a.Gehaltshistorie
from Angestellten as a
where a.Name = 'Holger Riedel'
```

alle Gehälter von Holger Riedel, wobei die Typstruktur `valid<int>` nicht aufgelöst wird. Da aber `valid<>` und `transaction<>`, als Funktionstypen betrachtet, mehrwertige Attribute darstellen, können diese auch durch Verwendung von Pfadvariablen aus zugegriffen werden, wobei `valid(o)` bzw. `transaction(o)` wie in TSQL2 verwendet werden können. So liefert die Anfrage

$Q_2 \equiv$
```
select ([Name,Gehaltshistorie]) a
from Angestellten as a, a.Gehaltshistorie as g
where a.Ang-Nr < 100 and valid(g) contains '1/1/97' and g.Gehalt > 4000
```

alle Angestellte mit den Attributen *Name* und *Gehaltshistorie*, wenn deren *Ang-Nr* < 100 ist und sie am 1.1.97 mehr als 4000 DM verdienten.
Im wesentlichen konnten die Konzepte aus TSQL2 zum Zugriff und Konstruktion temporaler Daten in TOQL übernommen werden. Da TSQL2 aufwärtskompatibel zu SQL2 konzipiert wurde, während eine entgültige Integration von SQL und ODMG-OQL noch aussteht, ergaben sich bei der Festlegung von TCROQUE-TOQL besonders da Probleme, wo sich SQL2 und ODMG-OQL im Detail unterscheiden:

- Da das TCROQUE-Modell zeitpunkt-basiert ist, können in den Anfragen stets Bezüge zu einzelnen Zeitpunkten hergestellt werden. Zusätzlich können durch die Funktionen `valid()` und `transaction()` die vorliegenden Zeitwerte in die Darstellung von Zeitelelementen, also einer Menge von Zeitintervallen, überführt werden, wobei undefinierte Zeitpunkte nicht berücksichtigt werden.

- Da TOQL die Auflösung von komplexen Werten ermöglicht, ist keine explizite Unterscheidung von `select` und `select snapshot` notwendig. Dies kann durch geeignete `select-from-where`-Blöcke ausgedrückt werden.
- In TSQL2 werden Surrogate (als eine Form zur Objektidentifikation) betrachtet, die in TCROQUE-TOQL nicht vorhanden sind, da Objektanfragen hier wie in ODMG-OQL formuliert werden.
- In OQL können Anfragen formuliert werden, die nichtrelationale Ergebnisse (einzelne Objekte oder Werte, Listen, etc.) liefern. Dies konnte in TCROQUE-TOQL integriert werden.
- TCROQUE-TOQL ist wie ODMG-OQL und TSQL2 eine streng typisierte Sprache. Dies hat zur Folge, daß alle Anfragen, die Objekte zurückliefern und ggf. eine Veränderung des Objektzustands bewirken, sorgfältig definiert werden müssen, um die Konsistenz von Anfrageergebnissen sicherzustellen. Dies betrifft die Mengenoperationen, wo stets globale Funktionen betrachtet werden. Seien z.B. durch zwei Anfragen zwei unterschiedliche Instanzen A_1 und A_2 des Angestellten-Extents gebildet worden, so wird bei dem Durchschnitt A_1 `intersect` A_2 für die Gehaltshistorie entweder die (global eindeutige) Gehaltshistorie genommen, wenn diese noch in beiden Instanzen unverändert enthalten ist, oder es existieren zwei neue Gehaltsfunktionen, wobei der Typcheck dann schon bei der Definition von A_1 bzw. A_2 eine Umbenennung dieser veränderten Gehaltsfunktionen erzwingt.
- In OQL können komplexe Werte im `select`-Teil durch die Verwendung von Typkonstruktoren aufgebaut werden. Dadurch können einige Standardanfragen (z.B. `group by`) durch `select-from-where`-Blöcke simuliert werden, so daß eine gesonderte Betrachtung von `group by` in OQL nicht notwendig ist. Diese Vorgehensweise wurde für TCROQUE-TOQL übernommen.
- Dies hat insbesondere den Vorteil, daß die meisten Aggregatsfunktionen über temporalen Daten durch eine geschachtelte `select-from-where`-Anfrage beantwortet werden können, indem zuerst die zu aggregierenden Werte in einer Menge zusammengestellt werden und dann eine Standard-Aggregation darauf angewandt wird. So kann die maximale Zeitdauer der Beschäftigung eines Angestellten durch

  $Q_3 \equiv$
  ```
  select struct(Name:a.Name, M-Dauer:  select max(end(i) - begin(i))
                                       from i in valid(a))
      from Angestellten as a
  ```

  bestimmt werden.

Auch sehr komplexe Anfragen wie *„Bestimme im Zeitraum 1990-1997 pro Angestellten die längste Zeitdauer, in der sein Gehalt nicht gesunken ist“*, kann mit normalen TOQL-Anfragen beantwortet werden, allerdings kann es hier sinnvoll sein, wie in TSQL2 spezielle Optionen (“`RISING`”) zu verwenden, um die Anfrageformulierung zu vereinfachen und die Auswertung zu optimieren.

Die physische Speicherung basiert auf dem CROQUE-Speichermanager [SGG+97]. Dieser ermöglicht für nichttemporale Objektbanken eine flexible Speicherung, wobei gemäß eines gegebenen physischen Schemas sowohl die beliebige Replikation wie auch verschiedene Clusterungen von Komponenten des Objektschemas möglich sind. Die physische Ebene wird durch eine Erweiterung von CROQUE-OQL beschrieben. Bei der Auswertung einer OQL-Anfrage wird eine mehrstufige Übersetzung über eine kalkülartige Notation in eine physische Algebra vorgenommen, wobei an mehreren Stellen Optimierungen vorgenommen werden (logische Vereinfachung der Anfrage, Zuordnung von Auswertungsoperationen und deren Anpassung an gegebene Speicherstrukturen). TCROQUE kann auf diesem Konzept orthogonal aufgesetzt werden:

- Da das CROQUE-Modell übernommen wurde, sind nur Anpassungen für die temporalen Teile des Objektschemas notwendig.
- Die Speicherung temporaler Objekte geschieht entweder durch Verwendung schon vorhandener Speicherstrukturen (B-Bäume, Auslagerungen von mengenwertigen Komponenten des Objektschemas) oder durch die Erweiterung des CROQUE-Speichermanagers um weitere Konzepte (etwa Time-Index, spezielle Speicherstrukturen für Valid-Time-Zeitreihen).

So können die Extents für *Person* und *Angestellte* etwa durch fünf Fragmente physisch abgebildet werden. In Fragment $\texttt{F}_1$ werden die Attribute *Name* und *Kinder* zusammen mit dem eindeutigen Objektidentifikator *oid* gespeichert, $\texttt{F}_2$ enthält die *oid*, die *Ang-Nr* und die gesamte Gehaltshistorie, sowie (zur Joinoptimierung) den zugehörigen *Namen*. Die Informationen zu *Qualifikationen* sind in ein eigenes Fragment $\texttt{F}_3$ ausgelagert worden, während $\texttt{F}_4$ ein Index für die *Ang-Nr* eines Angestellten und $\texttt{F}_5$ ein Time-Index für die *Gehaltshistorie* ist.
Bei dieser Speicherung kann die Anfrage Q_2 nur unter der Verwendung von $\texttt{F}_2$, $\texttt{F}_4$ und $\texttt{F}_5$ ausgewertet werden. Die Bedingungen des `where`-Teils werden in Lookups für $\texttt{F}_4$ und $\texttt{F}_5$ umgesetzt, ein nachfolgender Join (über die Angestellten-oids) liefert die Ergebnisobjekte, wobei die Ausgabeattribute aus dem Fragment $\texttt{F}_2$ bestimmt werden können.

4 Zusammenfassung

In diesem Artikel wurden die Möglichkeiten des Einsatzes von objektorientierten Datenbanken bei der Realisierung von zeitorientierten Informationssystemen diskutiert. Dabei hat sich folgendes Bild ergeben:

- Die Umsetzung von OO-entworfenen Anwendungssystemen auf relationale Datenbanksysteme unterliegt einigen Beschränkungen, insbesondere führt die zwingende Normalisierung von Objekt- und Zeitdaten zu unbefriedigenden logischen Datenmodellierungen. Auch ist die Umsetzung von Anfragesprachen wie TSQL2 kaum befriedigend möglich.
- Die zur Zeit kommerziell verfügbaren OO-Datenbanksysteme erleichtern die Modellierung von temporalen Datenbanken, führen aber durch die fehlende Standardisierung zwischen unterschiedlichen OODBMSen zu Insellösungen. Desweiteren sind die in den OODBMSen realisierten Konzepte zum Teil zu unflexibel, um den zusätzlichen temporalen Anforderungen gerecht zu werden.
- Der zukünftige ODMG-Standard kann eine vernünftige Basis sein, wenn zu den angebotenen konzeptuellen Elementen auf der logischen Ebene geeignete flexible Realisierungsmöglichkeiten bereitstehen.
- Sowohl die objektorientierte als wie auch die temporale Modellierung enthalten eine sehr große Zahl von Detailkonzepten, deren vollständige Integration in Zukunft noch ausgearbeitet werden muß.

Deswegen wird in diesem Artikel eine erweiterbare Kernarchitektur vorgeschlagen, die zum einen die aktuellen Entwicklungen des ODMG-Standards und der TSQL2-Vorschlags umfaßt, zum anderen aber auch flexible Erweiterungen ermöglicht.

Literatur

[BFG96] Bertino, E., E. Ferrari und G. Guerrini: *A Formal Temporal Object-Oriented Model.* In: *5th Intl. Conference on Advances in Database Technology (EDBT'96)*, Seiten 342–356, 1996.

[BS97] Barnert, R. und G. Schmutz: *Die zeitbezogene Datenhaltung bei den Schweizer Regionalbanken.* Wirtschaftsinformatik, 39(1):45–53, 1997.

[Cat96] Cattell, R.G.G. (Herausgeber): *The Object Database Standard: ODMG-93, Release 1.2.* Morgan Kaufmann, 1996.

[CDI+97] Clifford, J., C. Dyreson, T. Isakowitz, C. Jensen und R. Snodgrass: *On the Semantics of Now in Databases.* ACM Transactions on Database Systems, 22(2):171–214, 1997.

[FM95] Fegaras, L. und D. Maier: *Towards an Effective Calculus for Object Query Languages.* In: *Intl. Conference on Management of Data (SIGMOD)*, 1995.

[GN93] Gadia, S. und S. Nair: *Temporal Databases: A Prelude to Parametric Data.* In: *[TCG+93]*, Seiten 28–66, 1993.

[HS91] Heuer, A. und M.H. Scholl: *Principles of Object-Oriented Query Languages.* In: *GI-Fachtagung Datenbanksysteme für Büro, Technik und Wissenschaft, BTW 91, Kaiserslautern*, Band 270 der Reihe *Informatik-Fachberichte*, Seiten 178–191. Springer, März 1991.

[Kim95] KIM, W. (Herausgeber): *Modern Database Systems.* ACM Press, 1995.

[Rie96] RIEDEL, H.: *On Consistency in Temporal Object Bases.* In: *6th Intl. Workshop on Foundations of Models and Languages for Data and Objects, Dagstuhl*, Seiten 149–156, 1996.

[RS97] RIEDEL, H. und M.H. SCHOLL: *A Formalization of ODMG Queries.* In: *7th Intl. Conference on Data Semantics (DS-7)*, 1997.

[SGG+97] SCHOLL, M.H., D. GLUCHE, T. GRUST, A. HEUER und J. KRÖGER: *Optimierung von generischen Operationen und Speicherstrukturen in Objekt-Datenbanken.* Technischer Bericht 32/97, Mathematik und Informatik, Universität Konstanz, 1997.

[Sno93] SNODGRASS, R.: *An Overview of TQUEL.* In: *[TCG+ 93]*, Seiten 141–182, 1993.

[Sno95a] SNODGRASS, R.: *Temporal Object-Oriented Databases: A Critical Comparison.* In: *[Kim95]*, Seiten 386–408, 1995.

[Sno95b] SNODGRASS, R. (Herausgeber): *The TSQL2 Temporal Query Language.* Kluwer Academic Publishers, 1995.

[TCG+93] TANSEL, A.U., J. CLIFFORD, S. GADIA, S. JAJODIA, A. SEGEV und R. SNODGRASS: *Temporal Databases: Theory, Design, and Implementation.* Benjamin/Cummings, 1993.

[WBB97] WANG, X.S., C. BETTINI und A. BRODSKY: *Logical Design for Temporal Databases with Multiple Granularities.* ACM Transactions on Database Systems, 22(2):115–170, 1997.

[Wij95] WIJSEN, J.: *Design of Temporal Relational Databases Based on Dynamic and Temporal Functional Dependencies.* In: *Workshop on Temporal Databases, Zurich*, Seiten 61–76, 1995.