

Betrifft: Bitemporale Datenhaltung – anspruchsvoll aber keine Hexerei!
Art der Info: Methodische Grundlage
Quelle: Erfahrungen aus einem erfolgreichen Projekt
Autor: Tony Fräfel (tony.fraefel@trivadis.com)

Im Beitrag „Vorsicht Datenhistorisierung!“ (SOUG Newsletter 2/99) beschreibt Marcel Caviola verschiedene Formen der Historisierung. Dabei zeigt es sich, dass für die Erfüllung der meisten Anforderungen bezüglich Historisierung nur die temporale oder bitemporale Datenhaltung in Frage kommen. Doch wie soll man vorgehen, wenn Daten bitemporal verwaltet werden sollen bzw. müssen?

Zu Recht verweist M. Caviola darauf, dass temporale und bitemporale Datenhaltung komplexe Konzepte erfordern. Es ist deshalb um so wichtiger, in diesem Bereich keine Ad hoc Lösungen anzupeilen, sondern sich auf fundierte und praxiserprobte Lösungsansätze abzustützen. Wie die Lösung der Trivadis AG aussieht und wie damit in einem Projekt gearbeitet wird, zeige ich in diesem Beitrag.

Warum überhaupt zeitabhängige Datenverwaltung? Die Welt ist einem stetigen Wandel unterworfen, und demzufolge auch die Informationen über diese Welt, welche wir in unseren Applikationen verwalten. Beispiele von Informationen, die sich mit der Zeit ändern, gibt es genügend: die Adresse einer Person (wie viele Adressänderungen haben Sie nach Ihrem letzten Umzug verschickt?), das Arbeitsverhältnis einer Person (wie hoch ist die Fluktuationsrate in Ihrer Firma?), Beziehungen zwischen Personen (ich habe vernommen, es soll vereinzelte Scheidungen geben...), Klassifizierungen und Gruppenzugehörigkeiten und und und.

In gewissen Fällen interessiert uns nur der aktuelle Zustand einer Information - solche Informationen zeitabhängig zu verwalten macht keinen Sinn -, in anderen Fällen müssen wir jedoch festhalten, wie die vergangenen Zustände der Information waren. Ausserdem gibt es Anwendungen, in denen jede Veränderung festgehalten werden muss, um z.B. nachweisen zu können, wie der Informationsstand zu einem bestimmten Zeitpunkt war oder wer wann was geändert hat.

Oft stimmt auch der Zeitpunkt der Änderung nicht mit dem Zeitpunkt der Mutation in unserer Anwendung überein: wir erfahren erst zwei Wochen später, dass jemand umgezogen ist, oder vielleicht schon einen Monat früher, dass jemand heiraten wird. Wir dürfen also den Mutationszeitpunkt in unserer Applikation nicht gleichsetzen mit dem Zeitpunkt an dem die Änderung effektiv erfolgt.

Mit der bitemporalen Datenverwaltung erreichen wir unter anderem folgende Ziele:

- vergangene (und zukünftige) Zustände einer Information gehen nicht verloren
- Informationen können unabhängig vom effektiven Zeitpunkt der Änderung in der Applikation mutiert werden
- es ist erkenn- und nachweisbar, über welchen Zeitraum eine Information einen bestimmten Zustand hatte bzw. hat

Zusammenfassend können wir festhalten, dass uns die bitemporale Datenhaltung eine vollständige Historisierung aus Sicht der Benutzer und aus Systemsicht bietet, d.h. welchen Zustand hatte ein Objekt zu einem bestimmten Zeitpunkt (bzw. über einen Zeitraum) und wie war der Informationsstand in unserer Anwendung zu einem bestimmten Zeitpunkt (bzw. auch hier über einen Zeitraum).

Dass die zeitabhängige Datenverwaltung ein ernstzunehmendes Thema ist, zeigt sich auch daran, dass sich verschiedene Gremien und Universitäten mit diesem Thema auseinandersetzen. Als Beispiele seien erwähnt der Arbeitskreis ZobIS (Zeitorientierte betriebliche Informationssysteme) der Universität Bern (www.ie.iwi.unibe.ch/zeit/zobis) oder das Standardisierungsgremium ANSI, welches für SQL3 temporale Erweiterungen vorschlägt (SQL/Temporal).

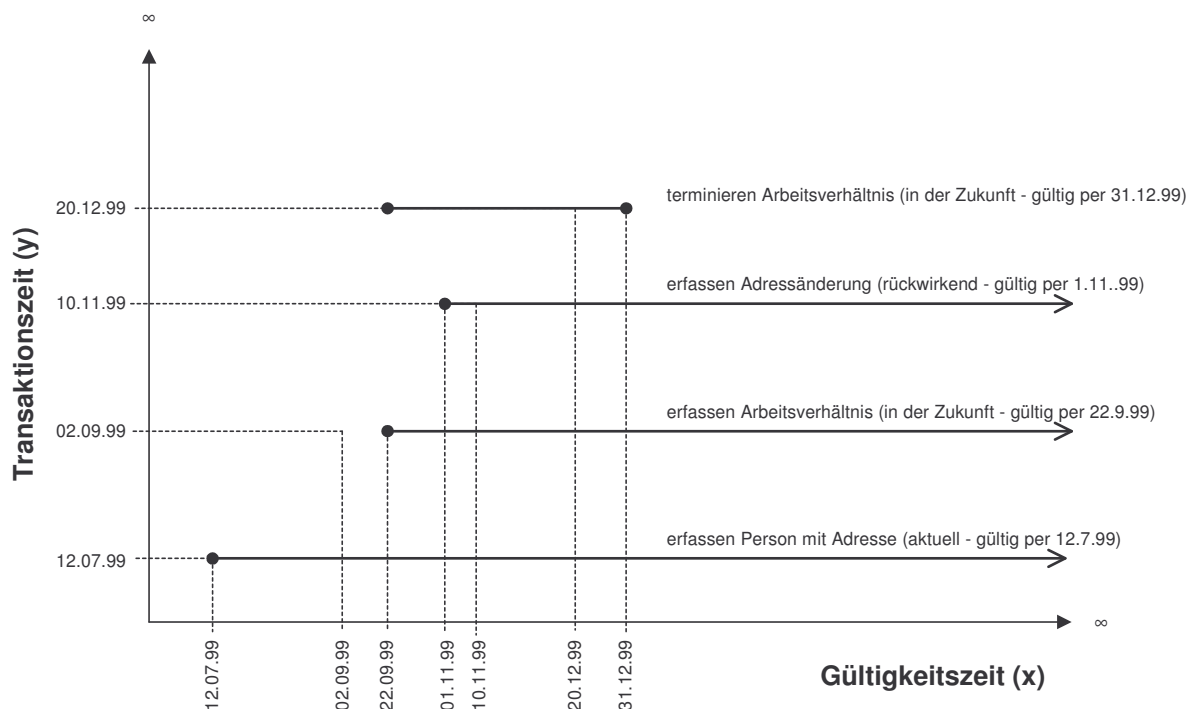
Die Lösung der Trivadis AG

Unsere Lösung basiert auf folgenden Komponenten:

- **Oracle Designer 2.1**
eingesetzt wird die Möglichkeit des Designers, Table- und Modul-API's zu generieren.
- **Oracle Developer (Forms 5.0)**
eingesetzt wird die Möglichkeit von Forms, einen Block auf einer Package (dem aus dem Designer generierten Modul-API) zu basieren. Als Frontend könnten natürlich auch andere Werkzeuge eingesetzt werden.
- **PL/SQL-Library der Trivadis AG**
Die von Trivadis entwickelte und vertriebene PL/SQL-Library umfasst verschiedene Features für die PL/SQL-Programmierung, darunter auch den TSQL-Generator, mit welchem Table-API's für die bitemporale Datenhaltung und Views basierend auf dem time-view Algorithmus generiert werden können (detaillierte Beschreibung folgt weiter unten).
- **Designer-Templates der Trivadis AG**
Die von Trivadis erstellten Designer-Templates, Preference Sets und Object Libraries für die Generierung von Modules (in diesem Fall Forms) aus dem Oracle Designer 2.1

Datenstrukturen für die bitemporale Datenhaltung

Die bitemporale Datenhaltung zeichnet sich unter anderem dadurch aus, dass sowohl die Gültigkeitszeit als auch die Transaktionszeit festgehalten wird, also in welchem Zeitraum ist eine Information in der realen Welt gültig (Gültigkeitszeit) und zu welchem Zeitpunkt wurde die Information im System erfasst bzw. verändert (Transaktionszeit). Die folgende Abbildung veranschaulicht diese beiden Zeiträume.



Bei der Erfassung bzw. Änderung einer Information können Transaktionszeit und Gültigkeitszeit gleich sein, d.h. die Information wird aktuell – gültig per sofort – erfasst. Es können aber auch rückwirkende Änderungen (Transaktionszeit > Gültigkeitszeit) oder Änderungen in die Zukunft (Transaktionszeit < Gültigkeitszeit) erfasst werden.

Das heisst, diese Zeiträume müssen in der Datenstruktur abgebildet werden können. Dazu werden folgende Columns in den bitemporalen Tabellen geführt:

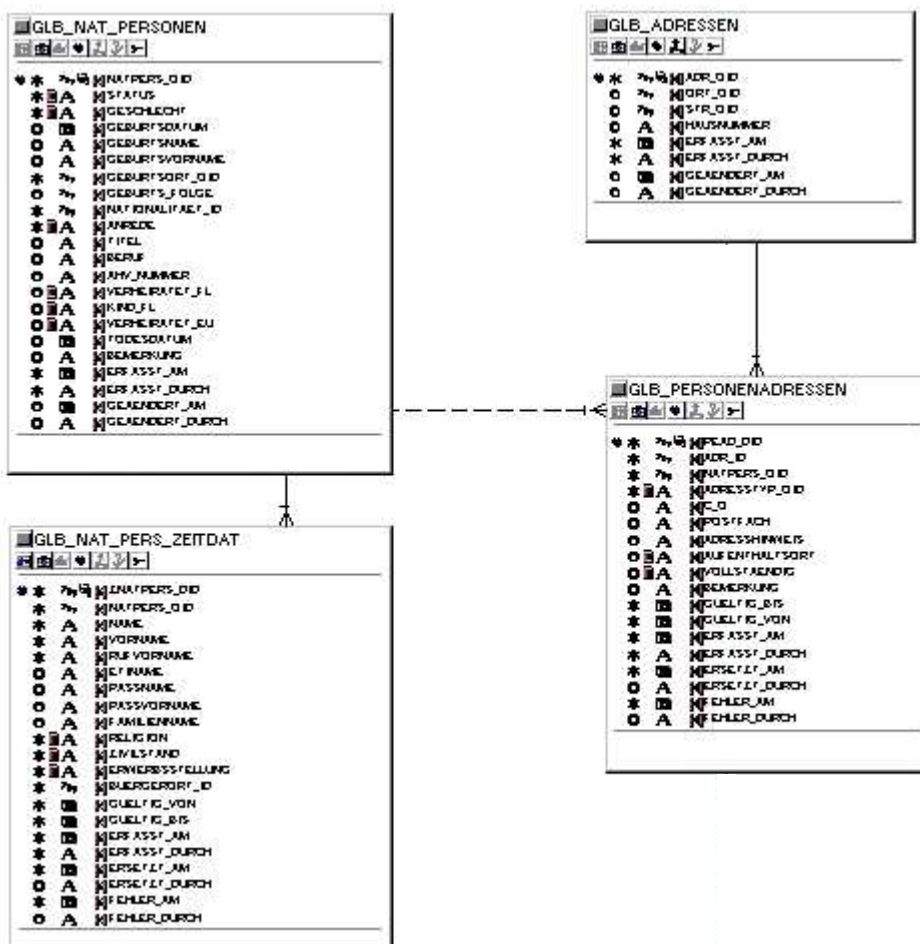
GUELTIG_VON	Startzeitpunkt der Gültigkeitsperiode
GUELTIG_BIS	Endzeitpunkt der Gültigkeitsperiode
ERFASST_AM	Zeitpunkt der Erfassung
ERSETZT_AM	Zeitpunkt der Stempelung von GUELTIG_BIS
FEHLER_AM	Zeitpunkt einer Fehlerkorrektur

Gültig_Von und Gültig_Bis werden bereits in der Analysephase festgehalten (es sind diejenigen Columns, welche die Benutzer interessieren), die restlichen Columns sind Zeitstempel der Transaktionszeit, d.h. für die Benutzer normalerweise nicht sichtbar (diese werden erst im Design hinzugefügt).

Damit auch ersichtlich ist, welche Benutzer die Daten erfasst oder verändert haben, werden folgende Columns geführt (sie sind aber nicht Voraussetzung für die bitemporale Datenhaltung):

ERFASST_DURCH	User-ID der Person, welche den Record erfasst hat
ERSETZT_DURCH	User-ID der Person, welche den Record ersetzt hat
FEHLER_DURCH	User-ID der Person, welche einen Fehler korrigiert hat

Als Beispiel zur Veranschaulichung der Lösung dient folgende Datenstruktur:

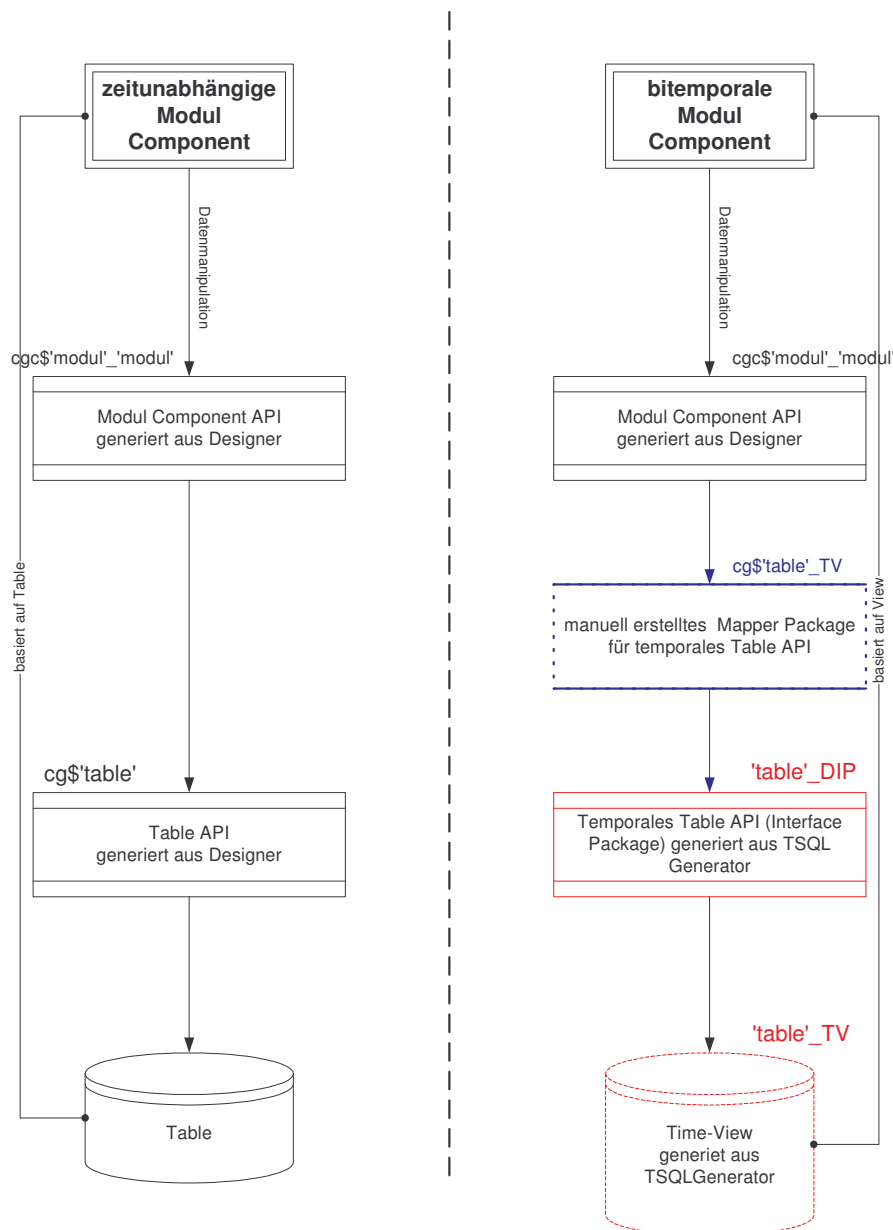


Für eine natürliche Person werden zeitunabhängige Daten (GLB_NAT_PERSONEN) und zeitabhängige Daten (GLB_NAT_PERS_ZEITDAT) gespeichert. Ebenfalls bitemporal ist die Adresse einer Person (GLB_PERSONENADRESSEN), welche eine zeitunabhängige Adresstabelle (GLB_ADRESSEN) referenziert.

Die verschiedenen Zeitstempel sind schon alles, was in der Datenstruktur bzw. in den Tabellen für die bitemporale Datenhaltung vorgesehen werden muss.

Das API für die bitemporale Datenhaltung

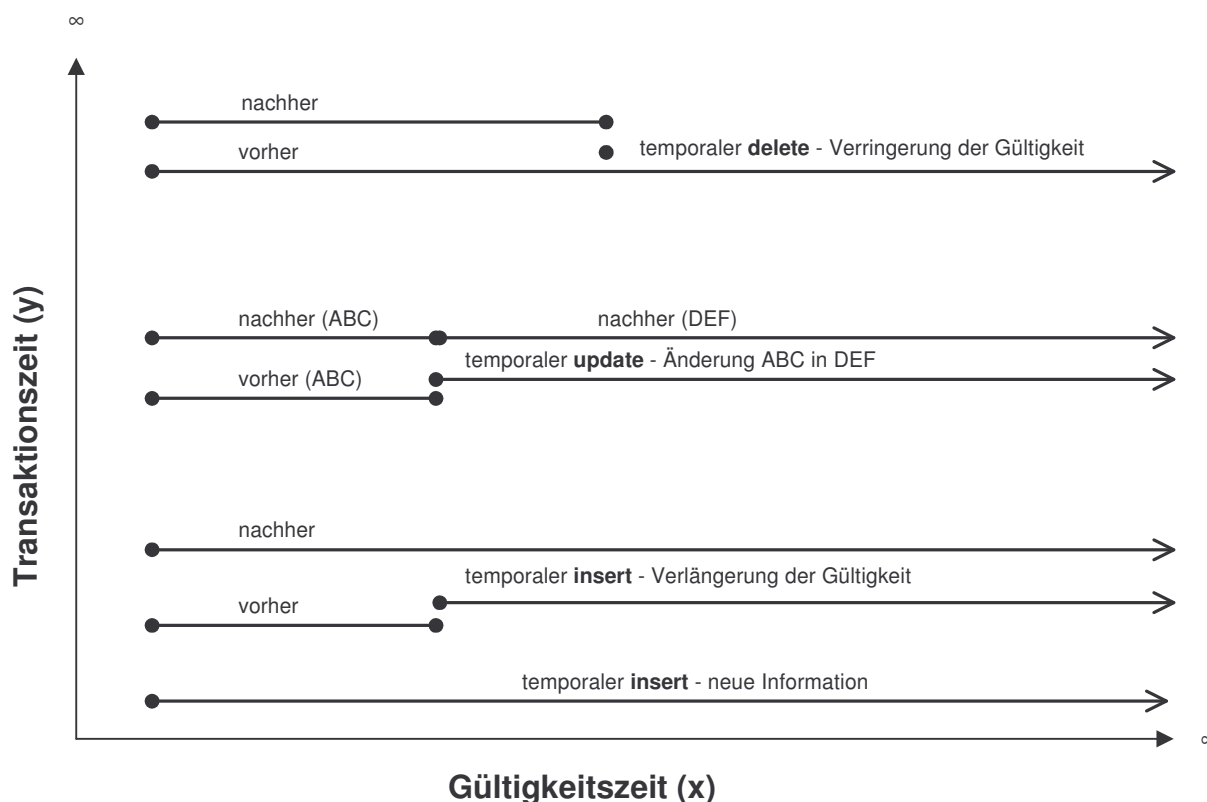
Die System- oder Modul-Architektur baut darauf auf, dass lesender Zugriff nur über eine View, mutierender Zugriff nur über ein API, d.h. mit PL/SQL-Packages, erfolgt. Die folgende Grafik zeigt, wie das API für zeitunabhängige Daten (generiert aus dem Oracle Designer) und dasjenige für bitemporale Tabellen aussieht.



Das MAPI wird in beiden Fällen aus dem Oracle Designer generiert. Das TAPI für bitemporale Tabellen wird mit dem TSQL-Generator (PL/SQL-Library der Trivadis) generiert. Darin werden unter anderem „normale“ SQL-Befehle umgesetzt in temporale Manipulationen gemäss TSQL2-Standard:

- **temporaler Insert**
fügt eine neue Information ein oder verlängert die Gültigkeit einer existierenden Information
- **temporaler Update**
verändert die Gültigkeitsdauer existierender Informationen (Wert ABC ist gültig bis x und wird ersetzt durch DEF gültig ab x)
- **temporaler Delete**
verringert die Gültigkeitsdauer existierender Informationen

Möchte ein Benutzer z.B. eine Adresse löschen (temporaler Delete), so wird daraus ein Update (nämlich das Setzen des Gültig_Bis der entsprechenden Adresse). Aus dem Einfügen einer neuen Adresse (temporaler Insert) wird ein Update (die alte Adresse ist bis zu diesem Zeitpunkt gültig) und ein Insert der neuen Adresse gültig ab diesem Zeitpunkt (siehe auch folgende Abbildung).



Um die Strukturen des MAPI und des TAPI bei den bitemporalen Tabellen in Übereinstimmung zu bringen, wird manuell ein Interface- oder Mapper-Package erstellt. Diese Arbeit ist relativ einfach (mapping einer Datenstruktur in eine andere) und erfordert wenig Aufwand.

Die Time-View

Sowohl lesender Zugriff als auch das TAPI einer bitemporalen Tabelle basieren auf einer sogenannten Time-View. Diese View stellt sicher, dass nur die zu einem bestimmten, definierbaren Zeitpunkt bzw. einer Zeitperiode gültigen Informationen angezeigt werden. Das heißt auch, dass als fehlerhaft gestempelte Records nicht angezeigt werden.

Als Beispiel enthält die Tabelle mit den zeitabhängigen Personendaten unter anderem folgende vier Records:

NAME	VORN	FAM	REL	ZIV	EWS	GUELTIG_VON	GUELTIG_BIS	ERFASST_AM	ERF_DURCH	ERSETZT_AM	ERS_DURCH	FEHLER_AM	FEHL_DURCH
Fräfel	Tony		BA	V	U	12.07.1999 00:00:00	21.07.1999 00:00:00	12.07.1999 13:11:20	FP	22.07.1999 14:29:41	KART	31.12.4712 23:59:59	
Fräfel	Tony		OR	V	U	22.07.1999 00:00:00	31.12.4712 23:59:59	22.07.1999 14:29:41	KART	31.12.4712 23:59:59		22.07.1999 14:34:19	KART
Fräfel	Tony	Fräfel	OR	V	U	22.07.1999 00:00:00	21.09.1999 00:00:00	22.07.1999 14:34:19	KART	22.09.1999 17:42:52	FRTO	31.12.4712 23:59:59	
Fräfel	Tony	Fräfel	OR	V	ER	22.09.1999 00:00:00	31.12.4712 23:59:59	22.09.1999 17:42:52	FRTO	31.12.4712 23:59:59		31.12.4712 23:59:59	

Wird für die Time-View kein Gültigkeitszeitpunkt gesetzt, so wird als Default SYSDATE genommen, d.h. aktuell gültig ist folgender Record:

NAME	VORN	FAM	REL	ZIV	EWS	GUELTIG_VON	GUELTIG_BIS	ERFASST_AM	ERF_DURCH	ERSETZT_AM	ERS_DURCH
Fräfel	Tony	Fräfel	OR	V	ER	22.09.1999 00:00:00		22.09.1999 17:42:52	FRTO	31.12.4712 23:59:59	

Wird für die Time-View ein Gültigkeitsbereich gesetzt (z.B. vom 1.7.1999 bis 31.12.1999), so werden die folgenden drei Records angezeigt – der am 22.07.1999 14:29:41 erfasste Record wurde am 22.07.1999 14:34:19 als fehlerhaft gestempelt und wird deshalb nicht angezeigt:

NAME	VORN	FAM	REL	ZIV	EWS	GUELTIG_VON	GUELTIG_BIS	ERFASST_AM	ERF_DURCH	ERSETZT_AM	ERS_DURCH
Fräfel	Tony		BA	V	U	12.07.1999 00:00:00	21.07.1999 00:00:00	12.07.1999 13:11:20	FP	22.07.1999 14:29:41	KART
Fräfel	Tony	Fräfel	OR	V	U	22.07.1999 00:00:00	21.09.1999 00:00:00	22.07.1999 14:34:19	KART	22.09.1999 17:42:52	FRTO
Fräfel	Tony	Fräfel	OR	V	ER	22.09.1999 00:00:00		22.09.1999 17:42:52	FRTO	31.12.4712 23:59:59	

Dieses Prinzip gilt auch für Masken, welche auf Time-Views basieren. Im folgenden Beispiel wurde am 12.7.99 eine Adresse erfasst, am 10.11. wurde sie rückwirkend auf den 1.10. geändert (ergibt einen zweiten Record in der Tabelle), am 10.11. wurde eine zweite Adressänderung erfasst (rückwirkend auf den 1.11.), die noch am gleiche Tag korrigiert wurde.

PEID	TYP	ADR_ID	GUELTIG_VON	GUELTIG_BIS	ERFASST_AM	ERF_DURCH	ERSETZT_AM	ERS_DURCH	FEHLER_AM	FEHL_DURCH
2001095	A	161862	12.07.1999 00:00:00	30.09.1999 00:00:00	12.07.1999 13:11:21	FP	10.11.1999 15:59:10	GLBN	31.12.4712 23:59:59	
2001095	A	146226	01.10.1999 00:00:00	31.10.1999 00:00:00	10.11.1999 15:59:10	GLBN	10.11.1999 16:00:23	GLBN	31.12.4712 23:59:59	
2001095	A	161866	01.11.1999 00:00:00	31.12.4712 23:59:59	10.11.1999 16:00:23	GLBN	31.12.4712 23:59:59		10.11.1999 16:01:26	GLBN
2001095	A	161862	01.11.1999 00:00:00	31.12.4712 23:59:59	10.11.1999 16:01:26	GLBN	31.12.4712 23:59:59		31.12.4712 23:59:59	

Der als fehlerhaft gestempelte Record wird in der Adress-History nicht angezeigt:

Gültig von	Gültig bis	Adress Typ	Adresse	Ort	A. Ort	Bemerkung
12.07.1999	30.09.1999	A	Leimackerstrasse 28,	8355 Aadorf		
01.10.1999	31.10.1999	A	Leimackerstrasse 28,	9999 Unbekann		
01.11.1999		A	Leimackerstrasse 28,	8355 Aadorf		

Die Fehlerstempelung

Das für bitemporale Tabellen generierte TAPI übernimmt auch die Handhabung der Fehlerstempelung. Ein Record wird als fehlerhaft gestempelt, wenn innerhalb einer bestimmten Zeitperiode, der sogenannten Granularität, eine Korrektur der Informationen für den gleichen Gültigkeitsbereich erfolgt.

Im obigen Beispiel wurde als Granularität ein Tag gewählt (im TSQL-Generator ist die Granularität pro Tabelle wählbar). Am 10.11.1999 16:00 Uhr wurde eine Adresse erfasst, gültig vom 1.11.1999 bis „unendlich“ (31.12.4712 ist das maximale Datum in einer Oracle-DB). Noch am gleichen Tag (um 16:01 Uhr) wurde eine andere Adresse erfasst mit dem gleichen Gültigkeitsbereich (1.11.99 bis unendlich). Dadurch wurde die erste Adresse als fehlerhaft gestempelt. Hätte die Korrektur einen anderen Gültigkeitsbereich gehabt (z.B. 1.12.99 bis unendlich), hätte das TAPI die Mutation als Adressänderung behandelt (die erste Adresse wäre dann vom 1.11. bis 30.11.99 gültig gewesen).

Das User Interface

Ein wesentlicher Punkt bei der bitemporalen Datenhaltung ist die Darstellung und Handhabung der zeitbezogenen Informationen für die Benutzer. Diese müssen bei ihrer Arbeitsweise folgende Punkte berücksichtigen:

- an welchem Gültigkeitszeitpunkt soll eine Information angezeigt werden (das „gültig am“ kann und muss gesetzt werden).
- ab welchem Zeitpunkt ist die Änderung einer zeitabhängigen Information gültig (das „gültig von“ kann und muss gesetzt werden).
- in den „normalen“ Masken werden meistens nur Informationen bezogen auf einen Zeitpunkt („gültig am“) dargestellt, Veränderungen über die Zeit werden in History-Masken dargestellt.

Für den Design des User Interfaces ergeben sich aus den oben erwähnten Punkten folgende Anforderungen:

- Das Setzen des Gültigkeitszeitpunktes („gültig am“) soll möglichst einfach erfolgen und wenn möglich vermieden werden können.
Umsetzung: Default für „gültig am“ ist SYSDATE (weil in mindestens 80% aller Fälle die aktuellen, heute gültigen Informationen abgefragt werden); für die Veränderung des Datums steht eine Kalender-Funktion (Kalender pop-up mit Navigationsfunktionen) zur Verfügung.
- Das Setzen des Startzeitpunktes der Gültigkeitsperiode („gültig von“) soll möglichst einfach erfolgen und wenn möglich vermieden werden können.
Umsetzung: Als Wert für das „gültig von“ wird das Datum des „gültig am“ übernommen. In vielen Fällen ist es auch hier das aktuelle Datum („was ich heute erfahre ist auch per heute gültig“). Bei rückwirkenden Änderungen (z.B. schon seit einem Monat gültig) oder Änderungen in die Zukunft (erst per nächstem 1. gültig) müssen die Benutzer zuerst auf das gewünschte Datum positionieren und sehen dadurch, was sie effektiv ändern (es kann durchaus vorkommen, dass z.B. in einem Monat eine andere Information gültig ist als heute).
- Das Setzen des Endzeitpunktes der Gültigkeitsperiode („gültig bis“) soll möglichst einfach erfolgen und wenn möglich vermieden werden können.
Umsetzung: Als Wert für das „gültig bis“ wird das Datum des „gültig am“ (minus 1 Tag) übernommen. Ändert ein Benutzer eine bestehende Information, so wird das „gültig bis“ des aktuellen Datensatzes automatisch gesetzt (wie das „gültig von“ des neuen Datensatzes). Wird eine Gültigkeitsperiode beendet (ohne eine neue Information zu erfassen), kann der Datensatz „gelöscht“ werden (mit ‚delete record‘), was in einen temporalen delete umgesetzt wird (d.h. das „gültig bis“ wird gesetzt).
- Für die Darstellung der verschiedenen Gültigkeitsperioden mit den entsprechenden Informationen werden spezielle History-Masken zur Verfügung gestellt.
Umsetzung: es werden Read-only History-Masken mit einheitlichem Layout erstellt; der Aufruf der History erfolgt über ein kontextsensitives Icon in der Toolbar (es wird die History-Maske zum aktuellen Block aufgerufen).

Die folgenden Screen-Shots zeigen die Umsetzung der bitemporalen Aspekte in einer konkreten Anwendung.

Adress Typ	Adresse	Ort	Bemerkung	Gültig Von	Gültig bis
Amtliche Adresse	Leimackerstrasse 28.	8355 Aadorf		12.07.1999	

Arbeitgeber	Standortadresse	Zustelladresse		Gültig von	Gültig bis
TRIVADIS AG			J 80 N OS	22.09.1999	

Gültigkeitsbereiche

Übersichtsmaske mit der Darstellung des Gültigkeitszeitpunktes (gültig am) und den Gültigkeitsbereichen (gültig von und gültig bis) bei zeitabhängigen Informationen. In weiteren Tabs können zusätzliche zeitbezogene Informationen dargestellt werden, z.B. erteilte Bewilligungen mit entsprechendem Gültigkeitsbereich:

Übersicht		Staatsbürgerschaft verw.		APA-Bewilligung		APA-Bew. History	
Typ	Gültig von	Gültig bis	Einreisedatum				
C	03.09.2001	31.12.4712	01.08.1999				
C	30.08.2001	31.08.2001	01.08.1999				
C	23.08.2001	24.08.2001	01.08.1999				
B	01.08.1999	31.07.2000	01.08.1999				
BIB	12.07.1999	20.07.1999	12.07.1999				

Mit dem History-Icon wird kontextsensitiv die entsprechende History-Maske mit ausschliesslich zeitbezogenen Informationen aufgerufen. Angezeigt werden auch der Zeitpunkt und der Benutzer einer Änderung. Zur Verbesserung der Übersichtlichkeit werden geänderte Werte farblich hervorgehoben.

Fax: +41 1 808 70 21