Umgang mit temporalen Daten

Einführung für Anwendungsfachleute

Version vom 13. Mai 2024 (deutsch)

Informationen und Kontakt: www.interlis.ch, info@interlis.ch

Inhalt

Vorwort

Im November 2022 erteilte swisstopo/KOGIS Sepp Dorfschmid, Adasys AG den Autrag das Thema „Umgang mit temporalen Daten“ in einer Voranalyse zu beleuchten und damit die Grundlage für ein zielgerichtetes Vorgehen im Rahmen einer Arbeitsgruppe zu legen.

Mit der Präsentation dieser Voranalyse anlässlich des Spirgartentreffen 2023 wurde zur Mitarbeit in dieser Arbeitsgruppe aufgerufen. Die folgenden Personen haben sich dann als Experten gemeldet.

Rolf Brändle/TBA/BD/ZHKT <rolf.braendle@bd.zh.ch>	Tiefbauamt ZH
Sepp Dorfschmid <do@adasys.ch>	Adasys AG
Claude Eisenhut <ce@eisenhutinformatik.ch>	Eisenhut Informatik AG
Erb Delia <Delia.Erb@sh.ch>	Amt für Geoinformation SH
Hans Rudolf Gnaegi <hgnaegi@ethz.ch>	
Oliver Grimm <oliver.grimm@geowerkstatt.ch>	Geowerkstatt GmbH
Stefan Henrich <stefan.henrich@moflex.ch>	moflex Infra GmbH
Fredy Spring <fredy.spring@gmx.ch>	

Die Arbeitsgruppe wurde im Oktober 2023 beauftragt, das Thema für Anwendungsfachleute verständlich aufzubereiten. Die Arbeitsgruppe befasst sich dann zunächst mit der Erarbeitung einer grundsätzlichen, eher abstrakt beschriebenen Vorstellung. Diese wurde am Jahresende der Beobachter-Gruppe zugestellt, für welche sich 8 Personen gemeldet hatten.

Die weitere Arbeit in der Arbeitsgruppe führte dann zur vorliegenden Fassung, welche nun wiederum der Beobachtergruppe zugestellt wird.

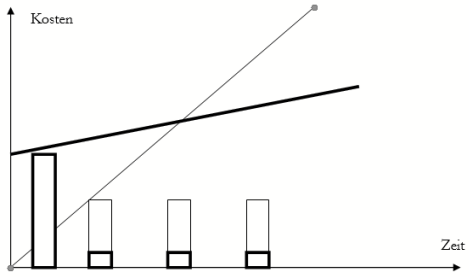
1 Überblick

Verschiedene Daten der öffentlichen Verwaltung und verwandter Gebiete haben temporalen Charakter: Infrastruktur-Objekte werden projektiert, werden oft intensiv abgefragt bevor sie bestehen und müssen zeitgerecht unterhalten werden; Regelungen (z.B. Gesetze, Eigentumsverhältnisse, Zonenpläne, Fahrpläne) interessieren vor und während ihrer Gültigkeit; vergangene Zustände interessieren um die Gegenwart zu verstehen.

Verbreitete Lösungsansätze sind weitgehend spezifisch auf ein konkretes Anwendungsgebiet ausgerichtet. Sie führen damit zwangsläufig für jedes Anwendungsgebiet zu einem erheblichen Entwicklungsaufwand, der mit dem Anwendungsgebiet eigentlich nichts zu tun hat. Dies lässt sich auch dann nicht vermeiden, wenn der Umgang mit Temporalität mittels heutigem INTERLIS-Modellen formuliert wird.

Es wird darum ein neuer Lösungsansatz beschrieben, mit dem wesentliche Anliegen wirkungsvoll so unterstützt werden, dass Datenbank-Konfiguration, Bearbeitung und Abfrage generisch – also unabhängig von der konkreten Anwendung – mit vernünftigem Aufwand realisierbar sind. Dabei wird insbesondere auch berücksichtigt, dass Programmsysteme, welche Temporalität nicht unterstützen – also nur die Bearbeitung eines bestimmten zeitlichen Zustandes ermöglichen – weiterhin zum Einsatz kommen können.

Selbstverständlich ist dieser Lösungsansatz mit diesem Dokument nicht einsatzbereit. Es braucht dafür weiter gehende Arbeiten (INTERLIS-Zusätze, Anpassung von INTERLIS-Werkzeugen, Erstellungen von Software-Komponenten). Diese werden zwar einiges kosten (in der Grafik rechts fett gezeichnet), die sonst nötigen individuellen Massnahmen (dünn gezeichnet) würden in der Summe aber zweifellos wesentlich teurer.



2 Motivation

2.1 Anliegen

In vielen Fällen wird mit Datenobjekten der gegenwärtige Zustand beschrieben. Von Zeit zu Zeit wird diese Beschreibung wieder gemäss der Realität nachgeführt.

Oft genügt das aber nicht. Man benötigt etwa:

- Daten zur Vergangenheit
  Wer war zu einem früheren Zeitpunkt Eigentümer eines bestimmten Landstücks?
  Wie war der Zonenplan zum Zeitpunkt, als das Haus gebaut wurde?
  Wer war damals Mitglied der Behörde?
- Daten zur Zukunft
  Bauten (Gebäude, Strassen, Bahnlinien) werden projektiert. Solche Projekte werden schrittweise erarbeitet, werden durch Gremien beurteilt, entsprechend wieder verändert. Solche Datenobjekte sollen darum auf den Systemen verfügbar sein, bevor sie real sind. Häufig soll auch der Projektierungsprozess (wer hat wann welche Änderung vorgenommen) dokumentiert sein.
  Ähnliches gilt für Regelungen (z.B. Gesetze, Eigentumsverhältnisse, Zonenpläne, Fahrpläne). Von besonderer Bedeutung ist hier der Zeitpunkt, ab dem die Regelungen gültig sind. Dieser

Zeitpunkt soll frühzeitig im System festgelegt werden, damit dann ohne weitere Massnahmen auf dem System die richtigen Daten abgefragt werden können und Abfragen für wählbare zukünftige Zeitpunkte möglich sind.

- Wiederkehrende und temporäre Daten
  Es gibt auch Sachverhalte, die immer wieder (z.B. Sonntagsfahrverbot) oder nur vorübergehend gelten (z.B. Provisorium während der Bauzeit). Hier möchte man den vorübergehenden Sachverhalt nicht immer wieder für einen bestimmten Zeitraum neu definieren bzw. den Originalzustand nach Ende des Provisoriums wieder zu erstellen.

2.2 Verbreitete Lösungsansätze

Selbstverständlich gibt es bereits Systeme, die temporale Möglichkeiten anbieten, z.B.:

- Fahrplansysteme: Eine Fahrplanabfrage kann auch für einen Zeitpunkt nach dem nächsten Fahrplanwechsel gemacht werden.
- Neue Datenmodelle der Amtlichen Vermessung: Daten können auch vergangene Objekte umfassen.

Deren Lösungsansätze sind in aller Regel auf das konkrete Problem ausgerichtet:

2.2.1 Problemspezifische Lösung

In solchen Lösungen, in denen Datenbank-Tabellen, Abfrage-Views und weitere Massnahmen spezifisch für das Problem definiert/programmiert werden, führen zu erheblichem Aufwand und ergeben kaum Nutzen für andere Anwendungen.

2.2.2 Modellierung mit heutigem INTERLIS

Es werden Klassen und Beziehungen modelliert, die beschreiben, wie die temporalen Daten organisiert sein sollen.

Leider ist damit das Problem nicht gelöst.

Dazu ein Beispiel:

```
CLASS BauVorhaben =
  Gueltig_ab: INTERLIS.XMLDateTime;
END BauVorhaben;


CLASS Gebaeude =
  EGID: 1 .. 900000000;
  Geometrie: MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                               VERTEX GeometryCHLV95_V2.Coord2
                               WITHOUT OVERLAPS > 0.002;
END Gebaeude;


ASSOCIATION Erstellung =
  ErstellungDurch -- {1} BauVorhaben;
  ErstelltesGebaeude - {0..*} Gebaeude;
END Erstellung;


ASSOCIATION Abbruch =
  AbbruchDurch -- {0..1} BauVorhaben;
  AbgebrochenesGebaeude - {0..*} Gebaeude;
END Abbruch;


VIEW Gebaeude_Gueltig
  PROJECTION OF Gebaeude;
```

```
    WHERE DEFINED(Gebaeude -> ErstellungDurch) AND
      NOT(DEFINED(Gebaeude -> AbbruchDurch));
      =
      ALL OF Gebaeude;
    SET CONSTRAINT INTERLIS.areAreas(ALL, UNDEFINED, >> Geometrie);
  END Gebaeude_Gueltig;
```

Wird nun dieses Modell mit den üblichen INTERLIS-Werkzeugen oder sonst systematisch in Datenbank-Tabellen umgesetzt, müssen die bearbeitenden Personen dieses Modell recht gut kennen. Sie müssen:

- Für eine neues Gebäude
    - Ein Objekt BauVorhaben erzeugen
    - Ein Objekt Gebäude erzeugen und die Attribute ausfüllen.
    - Das Gebäude-Objekt mittels der Erstellungs-Beziehung dem BauVorhaben-Objekt zuordnen
- Für einen Gebäudeabbruch
    - Das Gebäudeobjekt mittels der Abbruch-Beziehung dem BauVorhaben-Objekt zuordnen

Aus Sicht der bearbeitenden Person wäre es aber wünschenwert, wenn man im Rahmen einer Änderungstätigkeit einfach ein neues Gebäude definieren bzw. ein Gebäude löschen könnte. Der Rest (die Zuordnung über die entsprechende Beziehung) erfolgt im Hintergrund durch das System. Dafür müsste aber entsprechender Programmcode erstellt werden.

Dieser Programmcode müsste ohne zusätzliche Massnahmen sehr spezifisch für das jeweilige Thema erstellt werden, weil es nicht maschinell erkennbar ist, welche Klassen und Beziehung den temporalen Umgang organisieren.

Weitere Nachteile sind:

- Die Modellierung für den temporalen Umgang, ist nicht trivial, nimmt einen erheblichen Umfang an und reduziert damit die Lesbarkeit des Modells.
- Bei solchen Modellen kann die Formulierung von Konsistenzbedingungen, die nur für einen bestimmten zeitlichen Zustand gelten, recht schwierig sein (vgl. View Gebäude_Gueltig).
- Die Modellierung legt fest, WIE die temporalen Daten organisiert sein müssen. Eine andere Organisation der Daten wird damit verunmöglicht.
- Systeme, die Temporalität wirksam unterstützen, könnten mit der ausmodellierten Temporalität nur schwer eingesetzt werden.
- Der Austausch von Daten für einen bestimmten zeitlichen Zustand (d.h. nicht alle vorhandenen Daten) ist durch das Modell und die INTERLIS-Transfer-Festlegungen nicht möglich.

2.3 Fazit

Es lohnt sich, eine Lösung zu suchen, die einerseits die Anliegen unterstützt und anderseits die Nachteile der heutigen Lösungsansätze vermeidet, indem sie den Umgang mit temporalen Daten generisch, also unabhängig von der konkreten Anwendung, löst.

3 Lösung

3.1 Zielsetzung

Mit der beschriebenen Lösung für den Umgang mit temporalen Daten sollen vor allem die in Kap. 2.1. aufgeführten Anliegen erfüllt und die in Kap. 2.2. erwähnten Nachteile vermieden werden.

In diesem Dokument, das sich an Anwendungsfachleute wendet, steht die konzeptionelle Sichtweise, das WAS im Vordergrund. Mit welchen technischen Massnahmen das WAS letztlich erreicht ist, wird im Sinne von **Methodenfreiheit** nicht näher festgelegt. Das WAS ist aber so konzipiert, dass generische, technische Umsetzungen mit vernünftigem Aufwand realistisch sind.

Für das konzeptionelle Modell ist die **Verständlichkeit** ein wichtiges Anliegen. Es soll darum möglichst weitgehend von technischen Elementen befreit sein.

3.2 Hinweise zur Lösungsbeschreibung

Mit dem Gedankenmodell werden abstrakte Begriffe und ihre Bedeutung eingeführt, mit welchen der Umgang mit temporalen Daten generisch – also losgelöst vom konkreten Fachgebiet – in den Griff genommen wird.

Im nächsten Unterkapitel wird dafür das Grundprinzip vorgestellt, anschliessend folgen Vertiefungen, dank denen weitere Anliegen abgedeckt werden können.Die Beschreibung erfolgt dabei immer anhand des Gebäudebeispiels (vgl. Kap. 2.2.2), das für den Erklärungsbedarf jeweils leicht abgewandelt wird. Das Beispiel ist bewusst einfach gehalten und beansprucht auch keine fachliche Richtigkeit.

Die Beispiele werden immer mittels INTERLIS beschrieben. Dabei werden wenige Spracherweiterung verwendet. In den Beispielen sind sie jeweils unterstrichen dargestellt. Sie sind als Skizzen zu verstehen, welche die Stossrichtung illustrieren. Wie die INTERLIS-Sprache bzw. das INTERLIS-Basis-Modell effektiv erweitert werden, muss in einer weiteren Bearbeitungsphase entwickelt werden.

In weiteren zwei Unterkapiteln wird dargelegt, wie so organisierte Daten bearbeitet und abgefragt werden können, ohne dass sich die bearbeitenden Personen um die technischen Details kümmern müssen.

Im letzten Kapitel des Dokumentes, welches sich an technisch Interessierte wendet, wird das Gedankenmodell nochmals kompakt beschrieben und mit einem erläuternden Beispiel abgeschlossen.

3.3 Grundprinzip

3.3.1 Organisation der Daten

Nicht temporal heissen Objekte bzw. ihre Eigenschaften, wenn ihre Werte unveränderlich sind, d.h. zwar ändern können, aber nur eine Werteversion gespeichert wird. **Temporal** heissen sie, wenn sich ihre Werte im Laufe der Zeit ändern können und die Werteversionen gespeichert werden.

Temporale Daten beschreiben nicht nur einen bestimmten Zustand. Sie umfassen mehrere **Daten-Zustände**. Jeder Daten-Zustand ergibt sich aus dem Vorgänger-Daten-Zustand durch einen **Update,** also ein Paket von Änderungen (Objekt erzeugen, löschen, Attribute ändern) an verschiedenen Daten-Objekten. Zu einem bestimmten Daten-Objekt gibt es pro Daten-Zustand einen **Objekt-Zustand** und zu jedem Attribut einen **Attribut-Zustand**.

Jeder Daten-Zustand umfasst alle Objekte mit ihren Attributwerten, die sich aus allen früheren Updates und dem aktuellen ergeben.

Beispiel

Bei einem existierenden Gebäude soll ein Anbau beim Haupteingang erstellt werden. Entsprechend wird das Attribut *Geometrie* geändert. Diese Änderung soll im Rahmen eines ersten Bauvorhabens erfolgen.

Nebst dem erwähnten Anbau soll im Rahmen eines weiteren (nachfolgenden) Bauvorhabens ein Umbau bei der Anlieferung auf der Rückseite desselben Hauses erstellt werden. Bei diesem Update beschreibt der Geometrie-Wert sowohl den Anbau vorn wie den Umbau hinten.

Der Gebäudeidentifikator (EGID) wird dabei nicht verändert.

Entsprechend ergeben sich drei Objekt-Zustände:

Orginal	Ohne Anbauten
Nach Bauvorhaben 1	Mit Anbau vorn
Nach Bauvorhaben 2	Mit Anbau vorn und Umbau hinten

Wie diese Objekt-Zustände im Datenbestand organisert sind, ist aus konzeptioneller Sicht unerheblich. Wichtig ist dass sie je nach Daten-Zustand, der in einer Abfrage verlangt wird, richtig angezeigt werden.

Modellskizze

```
UPDATE BauVorhaben =
  Gueltig_ab (ORDER): INTERLIS.XMLDateTime;
  Kommentar: TEXT;
END BauVorhaben;

CLASS Gebaeude TEMPORAL BY BauVorhaben =
    EGID: 1 .. 900000000;
    Geometrie (TEMPORAL):
               MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                                 VERTEX GeometryCHLV95_V2.Coord2
                                 WITHOUT OVERLAPS > 0.002;
END Gebaeude;

CLASS Wohnung TEMPORAL BY BauVorhaben =
  AnzahlZimmer (TEMPORAL): 1 .. 99;
END Wohnung;

ASSOCIATION Wohnungen TEMPORAL BY BauVorhaben =
  Gebaeude -<#> {1} Gebaeude;
  Wohnung -- {0..*} Wohnung;
END Wohnungen;
```

Ausganspunkt für ein konzeptionelles Modell mit Temporalität ist immer das entsprechende nicht temporale Modell. Dieses gilt (inkl. aller Konsistenzbedingungen) für jeden zeitlichen Zustand. Die Angaben zur Temporalität müssen nur noch ergänzt werden:

- *BauVorhaben* ist nicht eine normale Klasse, sondern ist ausdrücklich als Update gekennzeichnet.
- Updates und damit Daten-Zustände sind immer sequentiell geordnet. Diese Ordnungkann zusätzlich mit einem Attribbut (markiert als ORDER), insbesondere einem Zeitpunkt beschrieben sein.
- Bei der Klasse *Gebäude* wird ausgesagt, dass Gebäude (eines oder mehrere) im Rahmen eines *BauVorhabens* entstehen, untergehen bzw. verändert werden.
- Da der *EGID* unveränderlich ist, ist es nicht als temporales Attribut gekennzeichnet.

- Da der Wert des Attributs *Geometrie* im Laufe der Zeit ändern kann, ist das Attribut als temporal gekennzeichnet
- Entsprechend bei Wohnung und deren Beziehung zum Gebäude. Da es keinen Sinn macht, dass eine Wohnung einem anderen Gebäude zugeordnet wird, muss die Beziehung nur erzeugt und gelöscht, nicht aber geändert werden.
- Die Kardinalität bei der Rolle Gebäude bleibt 1, obwohl im Laufe der Zeit mehrere Objektzustände des Gebäudes existieren.

Im Datenmodell muss man sich also betreffend Temporalität nur um das WAS und nicht um das WIE kümmern. Das WIE ergibt sich aus den Regelungen zum INTERLIS-Transfer und aus der Umsetzung im jeweiligen System (vgl. Kapitel 4).

3.3.2 Bearbeitung temporaler Daten

3.3.2.1 Prinzip

Personen, welche temporale Daten bearbeiten, sollten sich möglichst wenig darum kümmern müssen, wie temporale Daten gespeichert sind.

Im Rahmen eines Bearbeitungsschrittes wählen oder erzeugen sie einen Daten-Zustand und bearbeiten diesen: Erzeugung oder Wahl des Objektes, Wert zu Attribut festlegen. Alles Weitere wird durch das jeweilige System geleistet.

Zusätzlich gibt es wenige Spezialfunktionen, wie z.B. die Änderung der Reihenfolge von Daten-Zuständen.

3.3.2.2 Direkte Bearbeitung

Als direkte Bearbeitung wird die Bearbeitung bezeichnet, wenn sie direkt auf dem temporalen System erfolgt. Realistische Umsetzungsmöglichkeiten sind in Kap. 4 skizziert.

3.3.2.3 Externe Bearbeitung

Häufig wird mit Systemen gearbeitet, die keine Temporalität unterstützen. Man möchte aber weiterhin mit ihnen arbeiten, weil man sich so gewohnt ist, weil sie günstig sind oder weil sie spezielle, auf das jeweilige Anwendungsgebiet ausgerichtete Funktionalität anbieten.

Dafür wird auf dem temporalen System ein neuer Daten-Zustand definiert, dann dessen Ausschnitt (als Menge von Objekten) definiert. Daraus ergibt sich der Ausgangszustand für die externe Bearbeitung, dessen Daten exportiert und dann im externen System importiert und dort bearbeitet werden.

Nach Erreichnung eines brauchbaren Zwischenzustandes oder des Endzustandes werden die Daten des externen Systems exportiert und im temporalen System (nach Wahl des Daten-Zustandes) importiert. Dabei kann auch erkannt werden, ob auf dem temporalen System in der Zwischenzeit Veränderungen vorgenommen wurden, welche zu einem anderen Ausgangszustand geführt hätten. Solche Fälle werden angezeigt und dann (direkt oder extern) bereinigt.

3.3.3 Abfrage temporaler Daten

Analog zur Bearbeitung sollen sich Personen, die temporale Daten abfragen, nicht darum kümmern müssen, wie diese gespeichert sind.

Eine Abfrage orientiert sich primär am (nicht temporalen) Datenmodell, wird aber je nach Art der Abfrage um die nötigen temporalen Parameter ergänzt.

3.4 Vertiefung

Dieses Unterkapitel befasst sich mit einer Reihe zusätzlicher Regelungen, welche insbesondere den Umgang mit grösseren Projekten wirksam unterstützen.

3.4.1 Reihenfolge der Updates ändern

Beispiel-Zusatz

Da sich das Bauvorhaben im Eingangsbereich verzögert, wird das Gültigkeits-Attribut auf einen Zeitpunkt nach dem Bauvorhaben im hinteren Hausteil gesetzt.

Gedankenmodell

Es ist offensichtlich problemlos, wenn im Datenbestand zwischen den beiden Umbauprojekten weitere Daten-Zustände eingefügt werden, welche andere Gebäude betreffen. Ändert man aber die Reihenfolge der beiden Bauprojekte (Anlieferung hinten zuerst), dann würden ohne weitere Massnahme fehlerhafte Grundrisse angezeigt (zuerst mit beiden Änderungen, dann nur diejenige beim Haupteingang vorne). Wird ein Daten-Zustand, der an einem bestimmten Objekt und Attribut eine Änderung vornimmt, vor einen anderen verschoben, der am selben Objekt und Attribut eine Änderung vornimmt, besteht diese Gefahr. Solche Situationen müssen algorithmisch erkannt, gemeldet und dann entsprechend korrigiert werden.

3.4.2 Statuswesen

Beispiel-Zusatz

Oft möchte man bei Änderungenvorhaben mit einer Status-Angabe über den Fortschritt informiert sein: Baugesuch eingereicht, bewilligt, im Bau, gebaut.

Es stellt sich z.B. die Frage, welche Änderungenvorhaben, die zu einem bestimmten Daten-Zustand führen, noch nicht „bewilligt“ sind.

Gedankenmodell

Der Status weist – anders als ein gewöhnliches Attribut – im Rahmen eines Daten-Zustandes nicht einen, sondern mehrere Werte auf, die nebst dem eigentlichen Wert den Zeitpunkt festhalten, ab dem der Status-Wert gilt.

Modellskizze

```
UPDATE BauVorhaben =
  Gueltig_ab (ORDER): INTERLIS.XMLDateTime;
  Kommentar: TEXT;
END BauVorhaben;


CLASS Gebaeude TEMPORAL BY BauVorhaben =
  EGID: 1 .. 900000000;
  Geometrie (TEMPORAL):
              MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                                VERTEX GeometryCHLV95_V2.Coord2
                                WITHOUT OVERLAPS > 0.002;
  Fortschritt (STATUS): (Gesuch, bewilligt, imBau, gebaut);
END Gebaeude;
```

Hinweise

- Der Status kann im Rahmen desselben Updates mehrfach (für verschiedene Zeitpunkte) geändert werden.

- Ein Status-Attribut eines Objektes kann auch funktional aus Zeit-Angaben (z.B. der Bewilligung, des Baubeginns) oder aus einem gleichartigen Status-Attribut des Update-Objektes abgeleitet werden.
- Wird nun ein Zustand (gemäss einem bestimmten Datum) abgefragt, ergibt sich primär, welche Daten-Zustände berücksichtigt werden müssen. Zusätzlich kann verlangt werden, dass der Status für ein früheres Datum angezeigt wird. So kann z.B. abgefragt werden, welche Bauprojekte, die bis zum Ende des nächsten Jahres fertig sein sollten, bereits bewilligt oder gar im Bau sind.

3.4.3 Schrittweise Bearbeitung und Änderungsdokumentation

Beispiel

Beim Anbau beim Haupteingang vorne musste das Vorhaben geändert werden, damit es bewilligt werden konnte. Das bedingt, dass ein Daten-Zustand auch geändert (=korrigiert), d.h. schrittweise aufgebaut werden kann. Es soll auch möglich sein, dass zu den einzelnen Änderungen festgehalten wird (und abfragbar ist), wer die Änderung wann gemacht hat.

Dies gilt erst recht bei grösseren Updates (z.B. einem Strassenprojekt).

Gedankenmodell

Ein neuer Daten-Zustand muss nicht in einem einzigen **Bearbeitungsschritt** eingebracht werden. Die Bearbeitung soll schrittweise erfolgen können. In jedem solchen Schritt können zusätzliche oder bisherige Objekte/Attribute geändert werden. Werden bisherige Änderungen **korrigiert**, stellt sich die Frage, ob der bisherige Wert-Zustand erhalten bleiben soll oder nicht. Dies soll durch das jeweilige Fachgebiet im Datenmodell festgelegt werden können (auch für nicht temporale Attribute möglich).

Der einzelne Bearbeitungsschritt ist dem Daten-Zustand zugeordnet und kann ebenfalls Objektcharakter haben und Angaben wie den Zeitpunkt der Änderung und die ändernde Person bzw. Stelle beinhalten.

Alle Änderungen können so gemäss den Anforderungen des Fachgebietes dokumentiert werden.

Modellskizze

```
STEP Bearbeitung =
  Person: TEXT;
END Bearbeitung;


UPDATE BauVorhaben STEPS Bearbeitung =
  Gueltig_ab (ORDER): INTERLIS.XMLDateTime;
  Kommentar: TEXT;
END BauVorhaben;


CLASS Gebaeude TEMPORAL BY BauVorhaben =
  EGID: 1 .. 900000000;
  Geometrie (TEMPORAL, CORRECTIONS):
              MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                                VERTEX GeometryCHLV95_V2.Coord2
                                WITHOUT OVERLAPS > 0.002;
END Gebaeude;
```

Hinweise

- Zum Bearbeitungsschritt (STEP) können nebst dem Zeitpunkt (automatisch durch das System erfassbar) weitere Attribute festgelegt werden. Bei der Update-Klasse wird die Step-Klasse dann erwähnt.
- Bei jedem Attribut-Zustand wird festgehalten, im Rahmen welches Steps der Wert entstanden ist.

- Wird bei einem Attribut angegeben, dass auch Korrekturen (CORRECTIONS = Änderung eines schon im Rahmen des Daten-Zustandes geänderten Attributwerts) dokumentiert werden, bleibt der bisherige Attribut-Zustand damit erhalten, hat aber keine unmittelbare Relevanz mehr.

3.4.4 Umgang mit umfangreichen geplanten Zuständen

Damit auch grössere Projekten wie z.B. eine Strasse einfach handhabbar sind, werden zusätzliche Möglichkeiten vorgesehen. Zur Erläuterung wird aber weiterhin das übliche Beispiel verwendet.

3.4.4.1 Aufteilung in Teilvorhaben

Beispiel

Beide Änderungsvorhaben (Anbau beim Haupteingang vorne, Umbau hinten bei der Anlieferung) werden zuerst in einem einzigen Daten-Zustand beschrieben. Dann erweist es sich, dass man den Eingang zuerst, die Anlieferung erst später realisieren will. Man definiert darum entsprechend zwei neue Daten-Zustände. Das Gesamtprojekt – also der primäre Daten-Zustand - soll aber erhalten bleiben. In den beiden Folge-Daten-Zuständen sind nur noch diejenigen Änderungen enthalten, die wirklich notwendig sind, konkret die Geometrie beim Eingang.

Gedankenmodell

Der primäre Daten-Zustand wird als **Grundlage** markiert. Alle geänderten Daten-Objekte müssen dann in einem der Folge-Daten-Zustände aufgenommen werden. In diesen Folge-Daten-Zuständen können auch zusätzliche Änderungen vorgenommen werden.

Die Objekt-Änderungen gemäss primärem Zustand werden bei der Bestimmung eines weiteren Daten-Zustandes nur berücksichtigt, wenn der Folge-Daten-Zustand für das entsprechende Objekt im gefragten End-Daten-Zustand enthalten ist.

3.4.4.2 Projekt-Varianten

Beispiel

Der Anbau beim Eingang hat zu Diskussionen geführt. Darum wurde eine Projekt-Variante entworfen. Diese wurde dann als Vorzugsvariante bezeichnet.

Gedankenmodell

Ein Daten-Zustand kann als **Projekt-Variante** zu anderen Daten-Zuständen bezeichnet werden. In den Daten muss dafür der Zusammenhang dieser Daten-Zustände festgehalten werden. Einer dieser Zustände muss als **relevant** gekennzeichnet sein, die anderen werden verworfen, bleiben aber im System. Der relevante Zustand wird in „normalen“ Abfragen (ohne dass man von den Varianten weiss) ausgewiesen.

Hinweis

- Dafür braucht es eigentlich keinerlei Zusatzinformation im Datenmodell. Man könnte aber auch verlangen, dass die im Datenmodell ausdrücklich verlangt werden muss, damit man insbesondere den Bearbeitungsdialog gezielt unterstützen kann.

3.4.4.3 Attribut-Varianten

Beispiel

Varianten kann es auch im Detail geben. In unserem Beispiel genügt es, einfach den Wert des Grundriss-Attributs so zu verändern, dass der bisherige und neue Wert erhalten bleibt und einer als der massgebende gilt.

Gedankenmodell

Änderungen zu einem Daten-Objekt und einem Attribut im Rahmen eines Bearbeitungsschrittes zu einem Update (also für einen Daten-Zustand), können als **Attribut-Varianten** bezeichnet werden. Nur eine kann relevant sein.

Modellskizze

```
CLASS Gebaeude TEMPORAL BY BauVorhaben =
  EGID: 1 .. 900000000;
  Geometrie (TEMPORAL, CORRECTIONS, VARIANTS):
              MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                                VERTEX GeometryCHLV95_V2.Coord2
                                WITHOUT OVERLAPS > 0.002;
END Gebaeude;
```

Hinweise

- Da die technische Massnahme sehr ähnlich ausfallen dürfte wie bei temporalen bzw. korrigierten Attrbutwerten, macht es Sinn, diese auch im Modell zu erwähnen (VARIANTS).
- Mit Attribut-Varianten könnten zum Beispiel bei der Erarbeitung eines Gesetzes die verschiedenen Anträge festgehalten werden.

3.4.4.4 Ausnahmeregelungen

Beispiel

Nehmen wir an, dass es während der Bauzeit im Eingangsbereich eine provisorische Baute braucht, die nach der Fertigstellung des Anbaus wieder abgebrochen wird. Es wäre aber mühsam, wenn dann mit einem weiteren Daten-Zustand der erneut gültige Original-Zustand wieder definiert werden müsste.

Gedankenmodell

Updates und damit Daten-Zustände können als provisorisch gekennzeichnet werden. Sie sind nur relevant bis ein gegebenes Enddatum erreicht ist.

Modellskizze

```
UPDATE BauVorhaben STEPS Bearbeitung =
  Provisorisch_ab (ORDER): INTERLIS.XMLDateTime;
  Provisorisch_bis (PROVISIONAL): INTERLIS.XMLDateTime;
  Kommentar: TEXT;
END BauVorhaben;
```

3.4.5 Spezielle Attributtypen mit zeitlicher Relevanz

Ohne im Einzelnen unser Beispiel zu strapazieren, kann es Attribute geben, die zeitliche Angaben enthalten, ohne dass sie temporalen Charakter haben (z.B. Geburtsdatum einer Person). Zwischen nicht temporalen und temporalen Attributen gibt es sogar noch eine Zwischenstufe: Attribute deren Wert sich aus dem Zeitpunkt ergibt, für den die Abfrage gemacht wird.

3.4.5.1 Zeitregel

Ein bestimmtes Daten-Objekt (z.B. ein Fahrverbot) gilt nicht immer, sondern nur innerhalb bestimmter Zeitspannen (z.B. jeden Samstag).

Werden die in einem bestimmten Zeitpunkt relevanten Objekte abgefragt, wird die Zeitregel als Filterkriterium angewendet.

3.4.5.2 Zeitreihe

Immer öfter werden bestimmte Werte immer wieder (zu fixen Zeitpunkten oder bei Bedarf) erfasst (z.B. Temperatur, Stromverbrauch, Ein-/Ausschaltung).

Obwohl also über die Zeit unterschiedliche Werte bestehen, macht es keinen Sinn, dafür unterschiedliche Daten-Zustände zu definieren. Es genügt, dass zum jeweiligen Wert (Einzelwert oder Struktur) der Zeitpunkt festgehalten wird. Bei der Abfrage für einen bestimmten Zeitpunkt gilt dann entweder der letzte Wert vor diesem Zeitpunkt oder allenfalls ein Wert, der funktional aus Vorgänger- und Nachfolgerwerten gerechnet wird.

Modellskizze

```
CLASS Gebaeude TEMPORAL BY BauVorhaben =
  EGID: 1 .. 900000000;
  Geometrie (TEMPORAL):
              MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                            VERTEX GeometryCHLV95_V2.Coord2
                            WITHOUT OVERLAPS > 0.002;
  JahresStromVerbrauch (COURSE): 0..100000 [kWh];
END Gebaeude;
```

Hinweise

- Mit COURSE wird angezeigt, dass es sich um eine Zeitreihe handelt.

3.4.5.3 Zeitfunktion

Das einfachste Beispiel einer Zeitfunktion ist das Alter. Bei unserem Personenbeispiel macht es selbstverständlich keinen Sinn, das Alter als temporales Attribut zu führen. Man müsste es ja immer wieder im Rahmen von Daten-Zuständen anpassen.

Damit dennoch bei einer Abfrage das Alter geliefert werden kann, ohne dass dazu spezielle Abfrage-Massnahmen (mittels Views) definiert werden müssen, kann das Alter-Attribut als Zeitfunktion zum Geburtsdatum definiert werden.

Zeitfunktionen dürften aber auch nützlich sein, um z.B. bei Infrastuktur-Objekten den Revision-Zeitpunkt abfragen zu können.

4 Umsetzung

4.1 Grundsätztliches

4.1.1 Zuwarten oder handeln

Das geschilderte Prinzip hat das Potenzial, den verschiedensten Anwendungsgebieten zu dienen, ohne dass für jeden Anwendungsfall neuer Programmcode geschrieben werden muss. Es genügt, die jeweiligen Datenmodelle gemäss den skizzierten Möglichkeiten zu ergänzen.

Dummerweise existieren unseres Wissens keine Systeme, die das Nötige leisten. Darauf zu warten, dass die Markt-Systeme das bringen, was man möchte, ist kaum realistisch oder mindestens mit unklarem Zeithorizont verbunden. In der Zwischenzeit wird es so weiter gehen wie bisher: heterogen, häufig mit verbilligenden Vereinfachungen, die sich schon bald rächen.

Handeln scheint uns also angezeigt. Dies ist nicht trivial und darum mit einem gewissen Aufwand und entsprechenden Kosten verbunden.

Mit dem vorgeschlagen Vorgehen hat man also einmal einen grösseren Aufwand, mit dem bisherigen Vorgehen vielfach einen (eventuell) etwas kleineren, jedoch in der Summe damit letztlich einen grösseren.

4.1.2 Methodenfreiheit

Methodenfreiheit ist ein altes – unseres Erachtens aber immer noch richtiges – Postulat. Es besagt, dass man den Systemen nicht vorschreiben soll, wie sie etwas tun sollen. Man muss aber versuchen, die Ziele (vgl. Kap 3) zu erreichen – auch wenn die Markt-Systeme nicht einfach das leisten, was man eigentlich möchte.

Es gilt aber auch die Umkehrung: Gesucht ist darum eine Vorgehensweise, welche die aktuellen Markt-Systeme nutzt, sie aber gezielt ergänzt, damit unsere Ziele erreicht werden.

4.2 Nötige Massnahmen

4.2.1 Erweiterung von INTERLIS

Datenmodelle werden primär so formuliert wird, dass sie für jeden beliebigen Zeitpunkt gelten (inkl. Konsistenzbedingungen und Kardinalitäten). Sekundär wird das Modell aber um folgende Elemente ergänzt:

- Definition von Klassen für Updates mit den aus fachlicher Sicht nötigen Attributen und der Angabe, welche Objekt-Klassen mit welchen Daten-Zustands-Klassen verändert werden können.
- Definition von Klassen für Bearbeitungsschritte mit den aus fachlicher Sicht nötigen Attributen (z.B. bearbeitende Stelle und Person) und Verwendung in den Klassen für Daten-Zustände.
- Angabe, welche Objekt-Klassen und Attribute temporal sind und welche Update-Klassen dafür zuständig sind.

Damit Modelle die zusätzlichen Angaben enthalten können, wird es voraussichtlich einerseits eine Sprach-erweiterung und andererseits neue Basisklassen (im INTERLIS-Modell) brauchen.

Nötige Arbeiten:

- Entwurf der Sprach-Erweiterungen
- Entwurf von zusätzlichen Datentransfer-Möglichkeiten
- Entsprechende Anpassung des Referenzhandbuchs
- Anpassung des Meta-Modells
- Anpassung des INTERLIS-Compilers
- Anpassung des UML-Editors

4.2.2 Massnahmen für einfache Nutzbarkeit

Mit der Erweiterung von INTERLIS wird der Umgang mit temporalen Daten in den einzelnen Fachgebieten noch nicht erleichtert. Dafür braucht es Zusätze, die zwar generisch (also unabhängig vom Fachgebiet), aber nicht unabhängig von konkreten Systemen realisiert werden können.

4.2.2.1 Funktionalitäten für temporale Datenbanken

Nötige Funktionalitäten:

- Ableitung des logischen Modells gemäss temporalem konzeptionellem Modell.
- Unterstützung der Bearbeitung im Sinne von Kapitel 3.3.
- Unterstützung von Abfragen im Sinne von Kapitel 3.4.

- Extraktor, der einen bestimmten Daten-Zustand aus der Datenbank ausgibt.
- Integrator, der externe Daten zu einem bestimmten Daten-Zustand in die Datenbank integriert

Die Komponenten für Bearbeitungs- und Abfrage-Unterstützung müssen mindestens teilweise spezifisch für die jeweiligen DB-Systeme erstellt werden. Wenn diese mit einer einheitlichen, systemunabhängigen Schnittstelle ausgestattet werden, könnten Integrator und Bearbeitungsoberfläche systemunabhängig realisiert werden.

4.2.2.2 Konkrete Werkzeuge

Werkzeuge müssen entsprechend erweitert werden:

- ili2db
- ilivalidator
- ig/Check
- ili2fme
- ModelBaker
- ….

5 Weiteres Vorgehen

Gemäss den im Kapitel 3 skizzierten Prinzipien können verschiedenste Anwendungsgebiete temporal beschrieben werden. Es stellen sich primär folgende Fragen:

- Genügen die skizzierten Möglichkeiten?
- Gibt es Beispiele, die damit nicht gelöst werden können?
- Welche Änderungen an den Prinzipien ergeben sich daraus?

Natürlich wird man sich auch die Frage stellen, zu welchen Kosten man die nötige Technik realisieren kann. Dafür ist es nötig, dass man insbesondere die Komponenten gemäss Kap. 4.2.2.1 etwas genauer unter die Lupe nimmt (verfügbare Module, Architektur, grobe Funktionalität, Kostenschätzung).

6 Kompaktes Gedankenmodell für technisch Interessierte

6.1 Vorbemerkung

Da die Umsetzung der Lösung im Zusammenhang mit INTERLIS steht, werden zwei Begriffe ersetzt, um in die INTERLIS-Welt zu passen:

- Daten-Zustand -> Basket-Zustand
  INTERLIS-Basket sollen in temporaler Hinsicht autonom sein. Aus technischer Sicht fällt der Basket-Zustand mit dem Update zusammen (bzw. eine fachliche Update-Klasse ist eine Subklasse des Basket-Zustandes).

- Attribut-Zustand -> Eigenschafts-Zustand
  Nicht nur Attribute sonder auch Referenzen zu anderen Objekten (bei Referenz-Attributen und Rollen von Beziehungen) können verschiedene Zustände haben.

Das im Folgenden kompakt dargestellte Gedankenmodell erhebt im jetzigen Zeitpunkt noch nicht den Anspruch auf Vollständigkeit und Richtigkeit. Diese werden im Zusammenhang mit den INTERLIS-Zusätzen erarbeitet.

6.2 Gedankenmodell

6.2.1 Grundprinzip

Ein temporaler Basket enthält sämtliche Daten zur Vergangenheit, Gegenwart und Zukunft (soweit sie erfasst wurden). Abfolge, zeitliche Wirkung und Änderungsdokumentation ergeben sich aus weitgehend unabhängigen Elementen:

Zustände

Basket-Zustände gliedern den Basket in aufeinanderfolgende Zusammenfassungen, die im Sinne des Topics konsistent sind. Ein Basket-Zustand enthält alle Sachobjekte, die für diesen Zustand relevant sind je mit den jeweiligen Objekt-Zuständen. Da die Basket-Zustände sequentiell geordnet sind (vgl. auch 6.2.2.1), ergibt sich ein Objekt-Zustand pro Eigenschaft auf dem Eigenschafts-Zustand des am nächsten vorangehenden Basket-Zustandes.

Bearbeitungsschritte

Änderungen an den Daten (Erzeugung/Löschung von Sachobjekten, Änderung von Eigenschaften) erfolgen immer im Rahmen eines Bearbeitungsschrittes zu einem bestimmten Basket-Zustand. Sie sind im Rahmen des Basket-Zustandes sequentiell geordnet. Der letzte Bearbeitungsschritt kann über längere Zeit offen bleiben. Sobald er abgeschlossen wird, können die zugehörigen Objekt- und Wert-Zustände erst wieder im Rahmen eines weiteren Bearbeitungsschrittes verändert werden.

6.2.2 Präzisierungen und Zusätze

6.2.2.1 Reihenfolge der Basket-Zustände

Die Reihenfolge kann geändert werden. Dabei gibt es allerdings Einschränkungen:

- Enthält ein Basket-Zustand die Erzeugung eines Objektes, kann der Basket -Zustand nicht nach einen Basket-Zustand verlegt werden, der eine Änderung dieses Objektes enthält.
- Enthält ein Basket-Zustand die Löschung eines Objektes, kann der Basket -Zustand nicht vor einen Basket -Zustand verlegt werden, der eine Änderung dieses Objektes enthält.
- Enthält ein Basket-Zustand die Änderung an einer bestimmten Eigenschaft eines Objektes, kann dieselbe Eigenschaft anderer Objekt-Zustände als Folge der geänderten Reiehnfolge falsch sein und muss darum manuell überprüft werden.

6.2.2.2 Status

Der Status beschreibt, wie sich ein bestimmter Basket-Zustand oder Objekt-Zustand im Laufe der Zeit entwickelt (z.B. Entwurf, genehmigt, ausgeführt). Beschrieben wird ein solcher Status durch einen Aufzählwert. Dieser kann als Zeitreihe (Zeitpunkt, ab dem ein bestimmter Wert gilt) oder als funktional abgeleitetes Attribut (insbesondere ab Zeitpunkten des Basket-Zustandes) modelliert sein.

6.2.2.3 Basket-Zustand und Bearbeitungsschritt aus fachlicher Sicht

Sowohl Basket-Zustand wie Bearbeitungsschritt können im Modell fachspezifisch subklassiert werden und dabei eigene Eigenschaften aufweisen.

Basket-Zustands-Eigenschaften können im Rahmen von Bearbeitungsschritten wie Sachobjekt-Eigenschaften verändert werden.

Eigenschaften von Bearbeitungsschritten (z.B. die bearbeitende Person) müssen im Rahmen der Bearbeitung definiert werden und sind nicht mehr veränderbar.

6.2.2.4 Vollständige Dokumentation von Änderungen

Objekt-Eigenschaften (temporal oder nicht) können mehrmals innerhalb desselben Basket-Zustandes geändert werden. Erfolgen die Änderungen im Rahmen desselben Bearbeitungsschrittes, entsteht nur ein Eigenschafts-Zustand gemäss der letzten Änderung. Erfolgen die Änderungen in verschiedenen Bearbeitungsschritten, ergeben sich verschiedene Eigenschafts-Zustände, sofern dies im Modell verlangt wird. Selbstverständlich gilt aber nur der jeweils neuste.

6.2.2.5 Varianten

Wird ein neuer sachlicher Zustand projektiert, ergibt sich oft das Bedürfnis nach Varianten. Varianten äussern sich in unterschiedlichen Zuständen, von denen nur einer relevant sein kann, d.h. nur einer in eine konkrete Auswertung einfliessen kann. Dies kann für ganze Basket-Zustände oder für Eigenschafts-Zustände Sinn machen.

6.2.2.6 Aufteilung/Zusammenfassung von Basket-Zuständen (Teilvollzug)

Insbesondere bei grösseren Projekten wird zunächst das Projekt als Ganzes bearbeitet. Im Laufe der Zeit erweist es sich, dass die Umsetzung in mehreren Phasen erfolgen muss.

Objekt-Zustände werden als Folge nicht mehr gleichzeitig real.

Ein solcher Basket-Zustand kann deshalb als Grundlage markiert werden. Die zugehörigen Eigenschaft-Zustände können dann nur real werden, wenn sie zu einem Folge-Basket-Zustand gehören, welcher real wird. In diesem Folge-Basket-Zustand können auch weitere Bearbeitungen vorgenommen werden.

6.2.2.7 Basket-Zustand ist nur vorübergehend relevant

Ein Basket-Zustand kann eine vorübergehende Regelung beschreiben (Provisorium). Erfolgt eine Abfrage für einen Zeitpunkt nach dem Abschluss des Provisoriums gilt der Basket-Zustand als nicht relevant, wird also in der Abfrage nicht berücksichtigt.

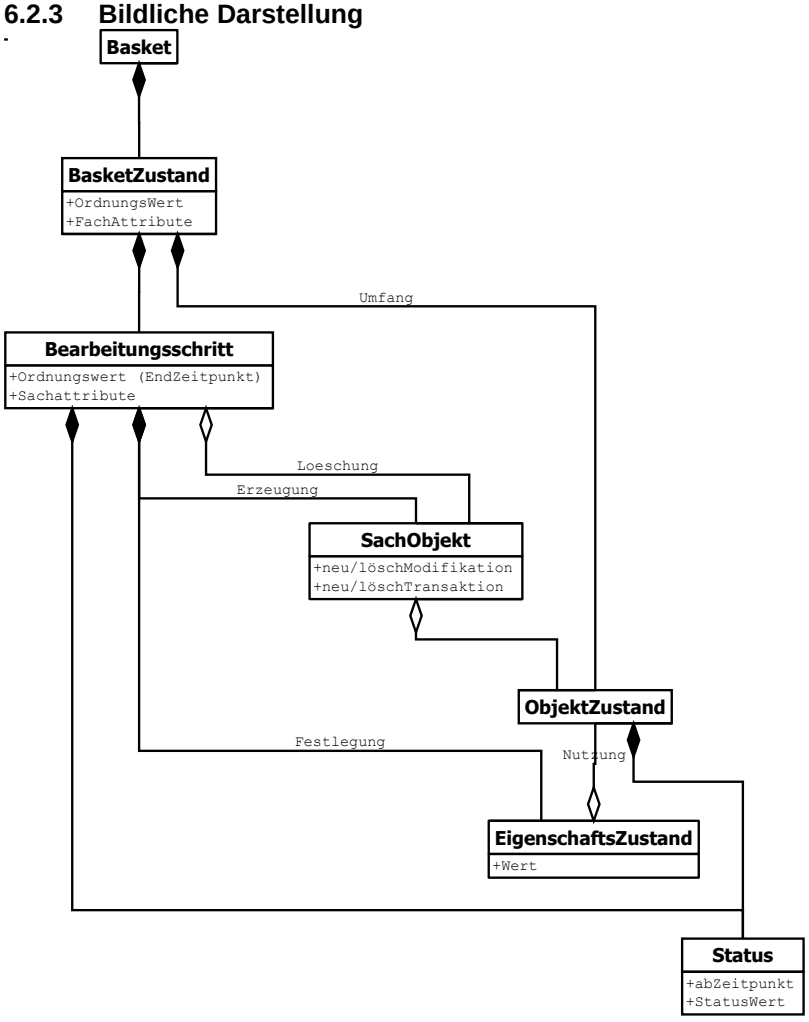
6.2.2.8 Eigenschaften mit zeitlich unterschiedlichen Werten

Typische Beispiele:

- Fahrverbot von 22-6 Uhr
- Zeitpunkt, zu welchem ein bestimmter Alterungszustand erreicht wird.

Solche Attribut-Werte ergeben sich funktional mittels Zeitfunktionen (mit dem Abfrage-Zeitpunkt als Parameter).

6.2.3 Bildliche Darstellung



6.3 Erläuterndes Beispiel

6.3.1 Modellskizze

```
STEP Bearbeitung =
  Person: TEXT;
END Bearbeitung;


  DOMAIN Status = (projektiert, bewilligt, geplant);


UPDATE BauVorhaben STEPS Bearbeitung =
  Gueltig_ab (ORDER): INTERLIS.XMLDateTime;
  Kommentar: TEXT;
  Status (STATUS): Status;
END BauVorhaben;
```

```
CLASS Gebaeude TEMPORAL BY BauVorhaben =
  EGID: 1 .. 900000000;
  Grundriss (TEMPORAL, CORRECTIONS):
              MANDATORY SURFACE WITH (STRAIGHTS, ARCS)
                                VERTEX GeometryCHLV95_V2.Coord2
                                WITHOUT OVERLAPS > 0.002;
  HeizungsArt(TEMPORAL, CORRECTIONS, VARIANTS): (Oel, Gas, LuftWP, ErdWP);
  BauStatus (STATUS): Status := Grundriss.BauVorhaben.Status;
  HeizungsStatus (STATUS): Status := HeizungsArt.BauVorhaben.Status;
END Gebaeude;
```

6.3.2 Basket-Zustände und Bearbeitungsschritte

Basket-Zustand	**Bearb-Schritt**	**Bearb-Zeitpt.**	**Beschreibung**
Initial		0	Gebäude 101 mit Gasheizung (G), 102 mit Ölheizung (Ö), 103 mit Ölheizung (Ö)
A	A1	1	Gebäude 101 mit Anbau vorn
	A2	4	Bewilligung vorgesehen für Zeit 7
	A3	11	Real zu Zeit 11
B	B1	2	Gebäude 101 mit Luftwärmepumpe (L)
	B2	5	Gebäude 101 mit Erdwärmepumpe (E) als Variante
	B3	8	Gebäude 101 mit Variantenentscheid Erdwärmepumpe, Gebäude 102 mit Erdwärmepumpe
	B4	12	Heizprojekt real zu Zeit 12
C	C1	3	Gebäude 103 soll abgebrochen werden
	C2	10	Abbruch bewiligt zu Zeit 10
	C3	13	Abbruch real zu Zeit 14
D	D1	6	Gebäude 101 mit Anbau hinten
	D2	7	Bewilligung vorgesehen für Zeit 9
	D3	14	Real zu Zeit 13 (verspätet eingetragen)

Die Basket-Zustände sind in der aufgeführten Reihenfolge geordnet. Die einzelnen Bearbeitungsschritte sind (selbstverständlich) pro Basket-Zustand geordnet, werden aber in der Reihenfolge der erwähnten Bearbeitungszeitpunkte ausgeführt.

6.3.3 Tabellarische Darstellung der Objekt- bzw. Eigenschafts-Zustände

Hinweise zur Darstellung:

- Kolonne BearbSchritt: Zeitpunkt der Ausführung, Name und Zeitpunkt gemäss Tabelle 6.3.2
- Löschungen werden mittels durchgestrichener ID angezeigt.
- Basket-Zustand mit Status projektiert (p), bewilligt (b), real (r).
- Der Wirkungszeitpunkt (Status ab) kann durchaus wesentlich nach der definierenden Transaktion liegen.
- Heizungsart (Ö, G, L, E): wenn mehrere Varianten: ! = relevant, ? = alternativ
- Der Objekt-Status ergibt sich aus den Basket-Zuständen für BauVorhaben, welches den Grundriss bzw. die Heizungsart veränderte (Grundriss/Heizungsart).

Bedeutung der fettgedruckten Angaben in der Tabelle:

- Basket-Zustand: Basket-Zustand neu erzeugt
- Status: geänderter Wert.
- Status-Ab: Zeitpunkt, ab dem der Status gilt
- ID: Aufnahme in den Basket-Zustand
- Grundriss, Heizung: geänderter Wert.

BearbSchritt		Situation Basket-Zustand			Situation Objekt-Zustand			
Zeit	Name	Name	Status	Status-Ab	ID	Grundriss	Heizung	Status
	Initial (0)	initial	r	0	101		G	r/r
	initial	initial	r	0	102		Ö	r/r
	initial	initial	r	0	103		Ö	r/r
1	A1	**A**	**p**	**1**				
	A1	A			**101**		G	p/r
2	B1	**B**	**p**	**2**				
	B1	B			**101**		**L**	p/p
3	C1	**C**	**p**	**3**	~~**103**~~			p/-
4	A2	A	**b**	**7**				
5	B2	B			101		**!L?E**	p/p
6	D1	**D**	**p**	**5**				
	D1	D			**101**		!L?E	p/p
7	D2	D	**b**	**9**				
		A			101		!L?E	b/p
8	B3	B			101		**!E?L**	b/p
	B3	B			102		**E**	r/p
		D			101		!E?L	p/p
9		D			101		!E?L	b/p
10	C2	C	**b**	**10**	~~103~~			b/-
11	A3	A	**r**	**11**				
		A			101		!L?E	r/p
12	B4	B	**r**	**12**				
		B			101		!E?L	r/r
		B			102		E	r/r
		D			101		!E?L	b/r
13	C3	C	**r**	**14**				
		D			101		!E?L	r/r
14	D4	D	**r**	**13**				
		C			~~103~~			r/-