Informatik Forsch. Entw.
DOI 10.1007/s00450-008-0039-3

REGULÄRE BEITRÄGE

Unterstützung von Periodizität in Informationssystemen – Herausforderungen und Lösungsansätze*

Markus Kalb · Peter Dadam

Eingegangen: 21 März 2007 / Angenommen: 18 Dezember 2007

Zusammenfassung Die systemseitige Unterstützung von Periodizität bzw. periodischen Spezifikationen weist Anforderungen auf, die weit über die temporalen Fähigkeiten heutiger Informationssysteme hinausgehen. Im Allgemeinen charakterisieren periodische Spezifikationen Vorgänge, die aus regelmäßig wiederkehrenden Aktivitäten bestehen. Neben der Ausdrucksstärke ist die größte Herausforderung periodische Spezifikationen miteinander vergleichen zu können. Diese Vergleichbarkeit ist ein wichtiger Aspekt in einer Vielzahl von Anwendungen, etwa um vorausschauend sich eventuell ergebende potentielle Ressourcen- oder Terminkonflikte erkennen zu können. Erschwert wird dieses durch unterschiedliche (zeitliche) Granularitäten sowie Ausnahmen in entsprechenden Spezifikationen. Für den praktischen Einsatz ist es darüber hinaus unumgänglich, periodische Zusammenhänge auch im Kontext einer großen (umfangreichen) Menge periodischer Daten effizient verwalten und auswerten zu können. Der vorliegende Beitrag gibt einen Einblick in die Herausforderungen sowie einen Überblick zu in der aktuellen Literatur vorliegenden Lösungsansätzen einer systemseitigen Unterstützung von periodischen Spezifikationen.

* Diese Arbeit wurde gefördert von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Projekts „*Periodizität in Informationssystemen*".

M. Kalb · P. Dadam (✉)
Institut für Datenbanken und Informationssysteme, Fakultät für Ingenieurwissenschaften und Informatik, Universität Ulm,
James Franck Ring,
89081 Ulm, Deutschland
e-mail: Peter.Dadam@uni-ulm.de

M. Kalb
e-mail: Markus.Kalb@uni-ulm.de

Schlüsselwörter Periodizität · Zeitliche Granularitäten · Temporale Datenmodelle · Temporale Anfragen

Abstract The requirements of a system side support of periodicity or rather periodic specifications are far beyond the temporal capabilities of current information systems. Generally, periodic specifications characterize activities, which consist of regularly recurrent elements. Beside the expressiveness, the challenge is the ability to compare periodic specifications among each other. Such comparability is an important issue in many applications. Examples are the identification of (potential) upcoming conflicts between different periodic specifications, like conflicts in resource or time schedules. An aggravating factor here is the use of different temporal granularities as well as the occurrence of exceptions in periodic specifications. In addition, real world scenarios typically have to deal with a high number of periodic specifications. Thus, comparisons between periodic specifications must retain their efficiency also in case of large data sets. This paper provides a detailed overview of the challenges of an integrated support of periodic specifications as well as of the current approaches in literature.

Keywords Periodicity · Temporal Granularities · Temporal Data Modells · Temporal Queries

CR subject classification A.1 · H.1 · H.3 · I.1 · I.2.4 · F.4

1 Einleitung

Die systemseitige Unterstützung von temporalen Aspekten ist ein wichtiger Aspekt in vielen Applikationen. Durch immer anspruchsvollere Einsatzgebiete für Informationssysteme, wie etwa im klinischen Bereich, sind die Anforderung an eine temporale Unterstützung jedoch stark gestiegen. Besonders deutlich wird dies im Kontext von prozessorientierten Informationssystemen, wo zunehmend temporale Fähigkeiten gefordert werden, die bislang nur in Projektmanagement- bzw. speziellen Planungssystemen realisiert waren, wie z.B. Scheduling, die Überwachung von zeitlichen Minimal- und Maximalabständen zwischen Aktivitäten, die Berechnung von Pufferzeiten und die Überwachung von Deadlines.

In diesem Zusammenhang ist das Fehlen einer adäquaten systemseitigen Unterstützung periodischer Zusammenhänge (Periodizität) für viele Anwendungen ein großes Manko. Periodizität ist ein weit verbreitetes Phänomen der realen Welt und alltäglich in vielen unterschiedlichen Anwendungsgebieten (z.B. Finanzsektor, klinischen bzw. medizinischen Anwendungssektor). Periodizität bzw. periodische Spezifikationen charakterisieren einen Vorgang, der aus regelmäßig (periodisch) wiederkehrenden Aktivitäten besteht. So enthalten medizinisch-organisatorische Abläufe und Therapieprozesse häufig Aktivitäten oder Abfolgen von Aktivitäten, die periodisch wiederholt werden müssen, wie z.B. einfache Angaben „*Alle 4 Stunden Gabe von Medikament X*" oder komplexe Therapieschema „*Eine Chemotherapie besteht aus Zyklen zu je 28 Tagen, wobei vom 1.–14. Tag Medikament A sowie am 1. und 8. Tag Medikament B verabreicht wird. Nach der letzten Gabe von Medikament A findet eine 14-tägige Pause statt. Es finden insgesamt 6 Zyklen statt.*".

Periodische Angaben wie etwa „*jeden 3. Tag*" oder „*jede 4. Stunde*" gehören einer einfachen Klasse von periodischen Spezifikationen an. In dieser Klasse der *äquidistanten* periodischen Spezifikationen bleibt der Abstand zwischen den einzelnen wiederkehrenden Elementen stets gleich. Im Gegensatz dazu variiert in der Klasse der *nicht-äquidistanten* Spezifikationen der Abstand zwischen den Elementen, wie etwa in dem obigen Beispiel der Chemotherapie. Solche Fluktuationen können auch durch die Verwendung von Granularitäten wie Monat oder Jahr (z.B. „*jeden ersten Tag im Monat*") oder auch durch Ausnahmen in den periodischen Spezifikationen (z.B. „*Gabe von Medikament B entfällt* in *jedem 3. Zyklus*") hervorgerufen werden. Auch wenn entsprechende Fluktuationen sehr komplex werden können, so ist ihr grundsätzliches Verhalten wohl bekannt (z.B. Definition der Schaltjahresregel). Im Gegensatz dazu stehen unscharfe periodische Spezifikationen. Die Abstände zwischen den Elementen entsprechen hier einer Menge von möglichen Abständen. Beispielsweise führt die zusätzliche Bedingung „*Je nach Verfassung des Patienten wird zwischen jeden Zyklus eine Pause von 0–5 Tagen eingeräumt*." zu Unschärfen im Verlauf der Therapie.

In Hinblick auf die benötigten (erweiterten) temporalen Fähigkeiten heutiger Informationssysteme reicht die einfache Bereitstellung ausdrucksstarker Spezifikationsformalismen für periodische Zusammenhänge alleine jedoch nicht aus. Stattdessen wird gleichzeitig auch eine angemessene operationale Unterstützung benötigt, die einen effizienten Umgang mit periodischen Spezifikationen ermöglicht. Die größte Herausforderung in diesem Zusammenhang bilden generische Vergleichsoperationen, die es erlauben, das Zusammenwirken periodischer Spezifikationen untereinander zu bewerten. So müssen unter anderem Aussagen zu Kongruenz, Überlappung oder Disjunktheit periodischer Angaben getroffen werden können, um z.B. vorausschauend (potentielle) Ressourcen- oder Terminkonflikte zu erkennen. Beispielsweise kann die zusätzliche Bedingung (siehe obige Chemotherapie) „*Medikament B muss von einem Spezialisten gegeben werden, der aber jeden ersten Mittwoch im Monat dienstfrei hat*" zu einem Konflikt zwischen dem Dienstplan des Spezialisten und der Gabe von Medikament B führen. Das heißt, in Abhängigkeit vom Startpunkt der Therapie können Überlappungen zwischen den dienstfreien Tagen und den Medikamentengaben auftreten.

Für die zur Realisierung von Vergleichen verwendeten Formalismen bzw. logischen Modelle reicht die Eigenschaft der Vergleichbarkeit alleine jedoch nicht aus. So ermöglichen etwa heutige Kalender- und Zeitplanungssysteme bereits eine Unterstützung von (einfachen) periodischen Spezifikationen. Allerdings ist ihre grundsätzliche Vorgehensweise dadurch geprägt, dass ihre Einsatzszenarien meist nur eine geringe Anzahl an periodischen Spezifikationen aufweisen. Zur Konflikterkennung bedarf es somit lediglich einiger weniger Einzelvergleiche, wobei bestehende Konflikte meist interaktiv durch den Anwender selber aufgelöst werden können bzw. müssen.

Im Umfeld etwa von prozessorientierten Informationssystemen mit einer Vielzahl an Prozess-Typen und möglicherweise vielen tausend Prozess-Instanzen stößt die obige Vorgehensweise jedoch rasch an seine Grenzen. Im Zusammenhang etwa mit Resourcenzuweisungen und damit verbundenen (vorausschauenden) Konflikterkennungen müssen umfangreiche Mengen an periodischen Spezifikationen repräsentiert und miteinander verglichen werden können. In einem solchen Umfeld ist es darüber hinaus unumgänglich, dass bestehende Konflikte weitestgehend automatisch aufgelöst werden können. Hieraus erwächst die Forderung nach adäquaten Repräsentationsformen, die neben einer kompakten speichereffizienten Darstellung zusätzlich auch eine effiziente Operationsrealisierung ermöglichen.

In der Literatur ist die Bedeutung einer integrierten Unterstützung von Periodizität in den letzten Jahren zunehmend erkannt worden. So finden sich viele Ansätze mit interessanten, teilweise jedoch sehr unterschiedlichen logischen Modellen und Realisierungsansätzen für die operationale Unterstützung. Ziel dieses Beitrages ist, einen Überblick über aktuell in der Literatur vorliegende Forschungsansätze zu geben sowie diese bezüglich ihrer Ausdrucksstärke und operationalen Unterstützung zu untersuchen. Wir gehen im Folgenden von einem logisch zentralen System mit „einer“ Uhr bzw. von einem verteilten System aus, bei dem die lokalen Uhren hinsichtlich der betrachteten Granularität ausreichend gut synchronisiert sind.

Der Beitrag ist wie folgt gegliedert: In Abschn. 2 werden die allgemeinen Anforderungen und Problemstellung an die integrierte Unterstützung von Periodizität im Detail vorgestellt. Anschließend werden in Abschn. 3 vorliegende Formalismen zur logischen Beschreibung von periodischen Spezifikationen bezüglich ihrer Ausdrucksstärke untersucht. Im Anschluss daran werden in Abschn. 4 die verschiedenen Ansätze zur Realisierung der operationalen Unterstützung in ihren grundsätzlichen Eigenschaften vorgestellt und bewertet. Eine Zusammenfassung in Abschn. 5 schließt diesen Beitrag ab.

2 Anforderungen und Problemstellungen

Im folgenden Abschnitt werden zunächst grundlegende Begriffe definiert. Im Anschluss werden die konkreten Anforderungen an die Ausdrucksstärke und operationale Unterstützung vorgestellt.

2.1 Motivation und grundlegende Begriffe

Periodische Spezifikationen beschreiben Vorgänge oder Aktivitäten, die sich regelmäßig wiederholen. Ein typisches Beispiel für periodische Spezifikationen, etwa im klinischen Umfeld, stellen die bereits in der Einleitung erwähnten Chemotherapien dar. Ein anderes Beispiel bilden Fahrpläne für Züge oder Busse mit ihren stündlichen oder täglichen Ankunfts- bzw. Abfahrtszeiten. Sehr häufig variieren diese stündlichen oder täglichen Zyklen. So gelten etwa am Abend oder am Wochenende andere Abfahrtszeiten als während des Tages oder an Wochentagen. Kennzeichnend für solche *zyklen-orientierten* Angaben ist die Verwendung eines *Zyklus* bestimmter Länge, der sich aus *Zykluselementen* zusammensetzt, z.B. 1. und 8. Tag einer Chemotherapie oder die 15. und 45. Minute in einer Stunde bei Fahrplänen.

Neben zyklen-orientierten Angaben stellen *kalender-orientierte* Angaben eine weitere Form von periodischen Spezifikationen dar, wie etwa „*Jeder 1. Tag im Monat*“, „*Jeden Montag*“ oder die folgende Beschreibung von regelmäßigen Projekttreffen: „*Während des Projektes finden Treffen jeden Mittwoch statt sowie zusätzlich jeden ersten Montag im Monat. Falls entsprechende Treffen auf einen Feiertag fallen, werden diese auf den nächsten Arbeitstag verschoben*.“.

Auch wenn die obigen Beispiele für periodische Spezifikationen aus sehr unterschiedlichen Anwendungsgebieten stammen, so weisen sie doch immer die gleichen „Komponenten“ auf. Zum einen ein **Periodizitätsschema**, welches das periodische Verhalten der Spezifikation festlegt und zum anderen einen **Instanziierungszeitpunkt**, der angibt, ab welchem Zeitpunkt dieses Schema gestartet bzw. instantiiert werden soll. In Abhängigkeit von der Anwendung kann die periodische Spezifikation zusätzlich noch in ihrem zeitlichen Gültigkeitsbereich bzw. in ihrer Wiederholungszahl eingeschränkt sein.

Das Periodizitätsschema beschreibt die gewünschte Abfolge von Zeitwerten in einem zugrunde liegenden Wertebereich. Zusammen mit der Angabe des Instanziierungszeitpunktes kann das Schema „ausgeführt“ bzw. angewendet werden und so eine Menge an konkreten Zeitwerten abgeleitet werden. Generell können zwei Formen von Periodizitätsschemas unterschieden werden: **äquidistante** und **nicht-äquidistante** Periodizitätsschemas. Im äquidistanten Fall bleibt der Abstand zwischen den einzelnen Zeitwerten stets konstant, wie etwa Angaben „*Jeden 2. Tag*“ oder „*Jede 3. Stunde*“. Im Gegensatz dazu stehen die nicht-äquidistanten Periodizitätsschemas, in denen der Abstand zwischen den einzelnen Zeitwerten variiert, wie beispielsweise in den obigen zyklen-orientierten Angaben oder in den Beispielen für kalender-orientierte Angaben.

Allerdings kann das nicht-äquidistante Verhalten eines Schemas unterschiedliche „Ursachen“ haben, was letztlich zu verschiedenen Formen von Periodizitätsschemas führt. Im folgenden Abschnitt werden diese Formen von nicht-äquidistanten Periodizitätsschemas im Detail vorgestellt und klassifiziert.

2.2 Nicht-äquidistante Periodizitätsschemas

Die einfachste Form von nicht-äquidistanten Periodizitätsschemas bilden Angaben, in denen die Reihenfolge der variierenden Abstände einfach vorgegeben sind. Beispielsweise führt in der Chemotherapie (Zykluslänge = 28 Tage) die Gabe von Medikament B am 1. und 8. Tag zu den variierenden Abständen von 8 bzw. 20 Tagen und somit zu einem nicht-äquidistanten Verhalten.

Komplexere Formen von nicht-äquidistantem Verhalten entstehen in Verbindung mit Ausnahmen sowie der Verwendung von Granularitäten unterschiedlicher Dauer in Periodizitätsschemas.

2.2.1 Berücksichtigung von Ausnahmen

Eine Ausnahme in einem Periodizitätsschema entspricht einer zeitweisen Abweichung bzw. Änderung von einem vorgegebenen Grundschema. Von der Änderung können sowohl eine *periodische* als auch *nicht-periodische* Auswahl an Elementen des Grundschemas betroffen sein. Diese Auswahl entspricht entweder einer Menge an Zyklen (*Zyklus-*

auswahl) oder einer Menge von Werten des Grundschemas unabhängig von ihrer Zugehörigkeit zu einem Zyklus (*Mengenauswahl*).

Die Angabe „*Jeden 2. Zyklus erfolgt am 10. Tag eine zusätzliche Gabe von Medikament B*“ oder die Angabe „*Jeden 2. Zyklus wird ein Bluttest nach der Gabe von Medikament A benötigt*“ entsprechen periodischen Ausnahmen, von der die Zyklusauswahl „*Jeden 2. Zyklus*“ betroffen ist. Die periodische Ausnahme „*Nach jeder 10. Gabe von Medikament B wird eine Laboruntersuchung benötigt*“ hingegen entspricht einer Mengenauswahl („*Jeder 10.Wert von Medikament B*“). Die einzelnen Medikamentengaben werden lediglich als Menge betrachtet.

Periodizitätsschemas weisen häufig nicht nur eine einzelne Ausnahme auf. Somit kann die Auswahl von Zyklen oder Elementen auch mehrfach pro Schema erfolgen. Bezüglich der Ausdrucksstärke der periodischen Auswahl bestehen die gleichen Anforderungen wie im allgemeinen Fall von Periodizitätsschemas. Beispielweise kann bei einer Chemotherapie eine Ausnahme auf Basis eines periodischen Feiertags definiert sein. Das Auswahlschema, d.h. das Periodizitätsschema für den Feiertag, kann wiederum nicht-äquidistantes Verhalten aufweisen und Ausnahmen beinhalten.

Im Gegensatz zu periodischen Ausnahmen entsprechen beispielsweise Projekttreffen, die an bestimmten Tagen annulliert werden oder auch einzelne Verlängerungen von Therapiezyklen, nicht-periodischen Ausnahmen. Bei solchen Ausnahmen ist lediglich eine endliche Menge an einzelnen Werten oder Zyklen von einer Änderung betroffen.

In Hinblick auf die konkrete Änderung an der periodischen oder nicht-periodischen Auswahl können drei Ausnahmeklassen unterschieden werden:

- **Löschung/Einfügung:** Das Löschen oder Einfügen von Zykluselementen in bestimmte Zyklen (bei Zyklusauswahl) oder von Werten in der Wertemenge (bei Mengenauswahl) sind Änderungen, die in diese Ausnahmeklasse fallen. Typische Beispiele hierfür sind etwa die vorherigen Angaben mit dem Einfügen von zusätzlichen Medikamentengaben oder weiteren Untersuchungen (Bluttest) sowie das Annullieren bzw. Löschen von Projekttreffen.
- **Selektive Verschiebung:** Das selektive Verschieben einzelner Zykluselemente oder einzelner Werte ohne Beeinflussung nachfolgender Werte fällt in diese Ausnahmeklasse. Beispiele hierfür bilden die Angabe „*Projekttreffen an Feiertagen werden auf den nächsten Arbeitstag verschoben*“ sowie das Verschieben der Gabe von Medikament B in bestimmten Zyklen.
- **Zusammenhängende Verschiebung:** Die Änderungen führen zu einem zusammenhängenden Verschieben aller nachfolgenden Zyklen oder Werte. Beispielsweise führt die Änderung von Zykluslängen zu einer Verschiebung aller nachfolgenden Zyklen inklusive ihrer Zykluselemente. Ein weiteres Beispiel bildet die Anpassung von Projekttreffen an die Sommer- bzw. Winterzeit. Die Berücksichtigung der Umstellung um eine Stunde führt zu einem zusammenhängenden Verschieben aller nachfolgenden Werte bzw. Projekttreffen. Die gleiche Situation ergibt sich bei der Berücksichtigung der Schaltjahresproblematik, d.h. die Änderung der Dauer des Monats Februars.

In den ersten beiden Klassen führen Ausnahmen lediglich zu „lokalen“ Auswirkungen in der Wertemenge eines Periodizitätsschemas. In der letzten Klasse hingegen führen Ausnahmen zu „globalen“ Auswirkungen in der Wertemenge, was den wesentlichen Unterschied zwischen den Ausnahmeklassen darstellt.

Die Berücksichtigung von Ausnahmen in Periodizitätsschemas führt, je nach Ausnahmeklasse bzw. Kombinationen der Klassen, zu komplexem nicht-äquidistantem Verhalten. Allerdings bilden Ausnahmen nur eine Ursache für nicht-äquidistante Periodizitätsschemas. Auch die Verwendung bestimmter Granularitäten kann diesen Effekt hervorrufen.

2.2.2 Verwendung von Granularitäten

Granularitäten (Zeiteinheiten) entsprechen unterschiedlichen Abstraktionsebenen zur Beschreibung von zeitlichen Zusammenhängen. So sind beispielsweise in Chemotherapien Angaben in Tagesabständen ausreichend, während andere Therapien stündliche Angaben benötigen. Im Finanzsektor wiederum sind monatliche oder jährliche periodische Aussagen typisch. Welche Granularität bzw. Granularitäten im Einzelnen verwendet werden, ist anwendungsabhängig. Allerdings müssen grundsätzlich zwei Verwendungsformen von Granularitäten innerhalb von Periodizitätsschemas unterschieden werden: *relative* und *absolute* Verwendung.

In der **relativen Verwendung** dienen Granularitäten lediglich als unterschiedliche Maßeinheiten für die Angabe von Abständen. Beispielsweise entspricht die Angabe der Länge einer Chemotherapie in *Tagen* oder die Angabe „*jede dritte Stunde*“ einer relativen Verwendung von Granularitäten.

In der **absoluten Verwendung** hingegen, wie etwa in dem Beispiel „*Jeden 1.Tag im Monat*“, übernehmen die Granules[1] der Granularität *Monate* die Rolle von Zyklen und die Granules der Granularität *Tage* die Rolle von Zykluselementen. Kennzeichnend ist hierbei, dass sich das periodische Verhalten der Granularitäten auf das Periodizitätsschema „überträgt“. Im gegebenen Beispiel etwa

[1] Ein Granule (engl.) bezeichnet die einzelnen Einheiten einer Granularität, d.h. ein einzelner *Tag* ist ein Granule der Granularität *Tage* bzw. der Monat *Januar* ein Granule der Granularität *Monate* [3].

müssen die jeweiligen Monatslängen sowie die Schaltjahresregel für Februar im Schema berücksichtigt werden. Nur so sind die korrekten Monatslängen und somit die Abstände zwischen den einzelnen Werten sichergestellt. Allerdings zeigt die nähere Betrachtung, dass es sich bei der Schaltjahresregel um das bereits beschriebene zusammenhängende Verschieben handelt. Das heißt, „*Jedes 4. Jahr*, *nicht jedes 100. Jahr und jedes 400. Jahr*" entspricht einem periodischen Auswahlschema für Februar zur Verlängerung der Monatslänge bzw. der Zyklen um einen Tag. In der absoluten Verwendung können Granularitäten somit als ein vom Kalender vorgegebenes Periodizitätsschema mit Ausnahmen verstanden werden. Die Ursache für das nicht-äquidistante Verhalten eines entsprechenden Periodizitätsschemas bildet somit der verwendete Kalender sowie die verwendete Granularität.

Die in Abschn. 2.1 vorgestellten kalender-orientierten Spezifikationen, wie etwa „*Jeden Montag*" oder „*Jeden ersten Mittwoch im Monat*", entsprechen in ihrer Semantik der absoluten Verwendung von Granularitäten. Die Berücksichtigung von Ausnahmen, wie zuvor dargestellt, ermöglicht daher eine einheitliche Unterstützung von Periodizitätsschemas sowohl für kalender-orientierte als auch für zyklen-orientierte Spezifikationen.

Allerdings gilt es, einen wesentlichen Unterschied zwischen beiden Spezifikationsformen zu beachten: Die Beschreibung der in kalender-orientierten Spezifikationen verwendeten Granularitäten erfolgt stets in Abhängigkeit vom Startpunkt des verwendeten Kalenders. Dieser Startpunkt ist für einen Kalender bzw. eine Granularität jedoch fest vorgegeben und darf nicht (beliebig) geändert werden. Für kalender-orientierte Spezifikationen bedeutet dies, dass diese zwar einen eigenen Instanziierungspunkt aufweisen können, „intern" jedoch stets den festen Startpunkt des Kalenders verwenden müssen. Im Gegensatz dazu wird bei zyklen-orientierten Angaben stets der vorgegebene Instanziierungspunkt verwendet.

Die vorgestellten verschiedenen Formen von Periodizitätsschemas geben einen ersten Überblick über die benötigte Ausdruckstärke von Beschreibungsformalismen und erhebt keinen Anspruch auf Vollständigkeit. Allerdings stellt die ausdrucksstarke Beschreibung von periodischen Spezifikationen lediglich einen, jedoch wichtigen Aspekt der systemseitigen Unterstützung dar. Der zweite Aspekt und größere Herausforderung ist die im Folgenden vorgestellte operationale Unterstützung von periodischen Spezifikationen.

2.3 Operationale Unterstützung

Die operationale Unterstützung umfasst eine Menge von Operationen, die den grundsätzlichen Umgang mit periodischen Spezifikationen ermöglichen. Hierzu zählen Operationen, die z.B. konkrete Zeitwerte für eine Spezifikation erzeugen oder auch überprüfen, ob eine Spezifikation einzelne Werte bzw. eine Wertemenge beinhaltet.

Die größte Herausforderung bilden Operationen zum Vergleich periodischer Spezifikationen. In vielen Anwendungen ist es unumgänglich, Aussagen zu Überlappungen, Kongruenz oder Disjunktheit von Werten zwischen periodischen Spezifikationen zu treffen, etwa in Hinblick auf eine vorausschauende Konflikterkennung.

Im Allgemeinen können die folgenden drei Klassen von Operationen unterschieden werden:

- **Ableitungsoperationen** dienen zur Ableitung konkreter Zeitwerte einer periodischen Spezifikation. Diese Operationen können sowohl Formen von „Zählern" (z.B. *3. Wert einer Spezifikation* oder *alle Werte des 4. Zyklus*) aufweisen als auch Semantiken der Form „*nächster Wert der Spezifikation*" bzgl. eines gegebenen Zeitpunktes (mit vorgegebener Granularität).
- **Überprüfungsoperationen** dienen dem Testen von gegebenen Werten bzw. Mengen von Werten, ob diese in einer periodischen Spezifikation enthalten sind oder durch diese erzeugt werden können. Beispielsweise wird dies benötigt, um (vorausschauend) festzustellen, ob bestimme Zeitpunkte bzw. feste Termine in einer Spezifikation enthalten sind. Darüber hinaus dienen diese auch dazu, die Ausführungshistorie einer Spezifikationen nachträglich auf ihre Korrektheit zu testen bzw. diese zu bestätigen.
- **Vergleichsoperationen** dienen der Bestimmung von gemeinsamen oder verschiedenen Werten von periodischen Spezifikationen mit gleichen oder verschiedenen Periodizitätsschema. Diese Operationen werden benötigt, um Aussagen zu Konflikten in entsprechenden Anwendungen treffen zu können, wie etwa im einleitenden Beispiel der Chemotherapie und dem Dienstplan bzw. dienstfrei des Arztes.

Wie erwähnt, stellt die Realisierung von Vergleichsoperationen die größere Herausforderung dar und wird deshalb einen Schwerpunkt in diesem Beitrag bilden. Im Folgenden werden hierzu weitere Anforderungen an diese Operationsklasse aufgezeigt.

Vergleichsoperationen zwischen periodischen Spezifikationen realisieren – sofern keine Schranken explizit angegeben werden – einen Vergleich zwischen zwei (theoretisch unendlich großen) Wertemengen. Das heißt es werden bestimmte Mengenbeziehungen zwischen den Spezifikationen überprüft. Beispielsweise entspricht die Teilmengenbeziehung *Dienstfrei* $\subset$ *Chemotherapie* überlappenden Wertemengen zwischen den beiden Spezifikationen, d.h. einige Termine der Chemotherapie stehen mit dienstfreien Tagen des Arztes in Konflikt.

Das *Auswertungsergebnis* eines Vergleichs bildet einen wichtigen Aspekt entsprechender Operationen. Im einfachsten Fall ist dies lediglich eine *boolesche Auswertung*, die überprüft, ob entsprechende Beziehungen vorliegen oder nicht. Allerdings trifft eine solche Auswertung keine Aussage, welche konkreten Werte die entsprechende Mengenbeziehung erfüllen bzw. nicht erfüllen. Bezogen auf das obige Beispiel bedeutet das, dass die konkreten Ereignisse, die den Konflikt verursachen, nicht lokalisiert werden können. Im Allgemeinen ist eine solche Lokalisierung jedoch unabdingbar, etwa um im Konfliktfall alternative Personen bzw. Ressourcen zu finden und zuteilen zu können. Eine *quantitative Vergleichsauswertung*, welche die konkreten erfüllenden bzw. nicht erfüllenden Werte einer Mengenbeziehung identifiziert, stellt das in der Regel benötigte Auswertungsergebnis dar.

Neben dem Auswertungsergebnis ist auch der *Auswertungszeitraum* von Vergleichen ein weiterer wichtiger Aspekt. Wie bereits erwähnt, werden bei einem Vergleich zwei möglicherweise unendlich große Wertemengen miteinander verglichen. Somit können entsprechende Konflikte zwischen periodischen Spezifikationen erst „sehr spät" in deren Verlauf auftreten. In der Praxis sind solche „späten" Konfliktaussagen meist nicht relevant. Beispielsweise ist die Erkennung eines Konflikts in einer Chemotherapie, der sich etwa erst in 10 Jahren etabliert, im Normalfall uninteressant. Im Gegensatz zu einer solchen *unbeschränkten Auswertung* ist es häufig ausreichend und sinnvoller, den Auswertungszeitraum eines Vergleichs einzuschränken. Grundlage für eine solche *eingeschränkte Auswertung* bildet ein *Ereignishorizont*, der anwendungsabhängig den Gültigkeitszeitraum einer Konfliktaussage festlegt. Der Ereignishorizont kann konstant sein oder von Vergleich zu Vergleich variieren.

Es reicht allerdings nicht aus, lediglich die prinzipielle Realisierbarkeit von Vergleichsoperationen bezogen auf einen bestimmten Spezifikationsformalismus zu betrachten. Auch die Effizienz der resultierenden Implementierungen ist sehr wichtig und diese ist wiederum sehr stark abhängig von der physischen Repräsentation von periodischen Spezifikationen. Neben einem möglichst *kompakten Speicherverhalten*, muss diese auch eine möglichst *effiziente Ausführung der Operationen* (Speicher- und Laufzeitverhalten) gewährleisten können. Insbesondere gilt es, aufwendige Transformationen zum Vergleichs- bzw. Anfragezeitpunkt zu vermeiden.

3 Beschreibung von Periodizitätsschemas

In der Literatur findet sich eine Vielzahl an Ansätzen zur logischen Beschreibung von Periodizitätsschemas. Im Folgenden nicht vertieft werden Ansätze, die ausschließlich qualitative periodische Zusammenhänge berücksichtigen, wie etwa „*Aktivität A regelmäßig <u>vor</u> Aktivität B ausführen*" (z.B. [13, 14, 27]).

Im Folgenden liegt der Schwerpunkt auf Ansätzen mit quantitativem Charakter, d.h. Ansätzen, die periodische Zusammenhänge über Wertemengen betrachten. Allerdings werden „unscharfe" quantitiative Periodizitätsschemas, wie etwa „*an einem beliebigen Montag innerhalb von jedem Monat*", nicht berücksichtigt. Hierzu zählen auch Periodizitätschemas, in denen Aktivitäten zusätzlich bestimmte Durchführungsdauern aufweisen. Die alleinige Angabe einer Durchführungsdauer etwa bei der Chemotherapie „*Die Gabe von Medikament B an jedem 1. und 8. Tag eines Zyklus benötigt 6 Stunden*", ohne Aussage, wann genau diese Aktivität innerhalb der einzelnen Tage staffinden wird, führt zu einer Unschärfe. Allerdings weisen entsprechende Periodizitätschemas ohne Berücksichtigung der Durchführungsdauern wieder ein exaktes quantitatives periodisches Verhalten auf, was im Folgenden im Vordergrund stehen wird.

Viele der im Folgenden erwähnten Ansätze stammen aus dem Umfeld der Beschreibung von Granularitäten bzw. Kalendern. Dies ist dadurch bedingt, wie auch in Abschn. 2.2 gezeigt, dass Granularitäten und Periodizitätsschemas, logisch betrachtet, lediglich zwei Seiten derselben Medaille darstellen. Auch wenn an periodische Spezifikationen operationale und interne Anforderungen bestehen, die weit über die von Granularitäten hinausgehen, können sie auf Ebene der formalen Beschreibung als gleich betrachtet werden.

Bezüglich des verwendeten Beschreibungsparadigmas können *algebraische* und *constraint-basierte* Ansätze unterschieden werden. Im Folgenden steht hierbei die generelle Vorgehensweise entsprechender Ansätze im Vordergrund.

3.1 Algebraische Ansätze

Kennzeichnend für diese Form des Beschreibungsparadigmas sind formale Operationen, deren sukzessive Anwendung die periodischen Eigenschaften eines Periodizitätsschemas schrittweise konstruieren. Vertreter dieser Kategorie von Ansätzen sind z.B. der *Collection-Formalismus* [22] (bzw. erweiterte oder angepasste Varianten [6, 8, 9, 12, 18, 29]), der *Slice-Formalismus* [2, 20, 21, 24] sowie die *Calendar-Algebra* [3, 25].

Im Allgemeinen können bei diesen Ansätzen zwei Grundformen von Operationen unterschieden werden: Gruppierungs- und Selektionsoperationen. In Bezug auf Granularitäten dienen Gruppierungsoperationen der Bildung neuer Granularitäten durch „Zusammenfassen" bzw. „Gruppieren" einer bestimmten Anzahl von aufeinanderfolgenden Granules aus einer bestehenden Granularität. Der Ausdruck $Tage = Group_{24}(Stunden)$ beispielsweise beschreibt die Granularität *Tage* durch Gruppieren von jeweils 24 *Stunden*. Im Zusammenhang mit Periodizitätsschemas dienen diese Operationen der Beschreibung von Zykluslängen, wie etwa bei der Chemotherapie durch das Gruppieren von 28 *Tagen* zu einem Zyklus, d.h. $Chemo = Group_{28}(Tage)$.

Für die Unterstützung von nicht-äquidistanten Granularitäten bzw. Zyklen finden sich spezielle Gruppierungsoperationen. Beispielsweise werden bei der Operation $Alter^{m}_{p_1 \ldots p_n,k}(G, G_2)$ die Granules von G jeweils in Gruppen zu je m Granules (logisch) zusammengefasst und die Granules an Position $p_1, \ldots, p_n$ um den Wert k von G_2 verkleinert bzw. vergrößert [3]. Das nachfolgende Beispiel zeigt eine mögliche Beschreibung der Granularität *Monate*. Hierbei werden in M_1 zunächst Granules durch Gruppieren von jeweils 30 Tagen gebildet und anschließend die Granules bzw. Monate mit 31 Tagen mit der Operation *Alter* entsprechend verlängert. Im Anschluss wird in M_2 der Monat Februar auf 28 Tage verkürzt. Im letzten Schritt wird in der Granularität *Monate* basierend auf einer vereinfachten Schaltjahrregel („*jedes 4.Jahr*") die Länge der Februare in Schaltjahren angepasst, d.h auf 29 Tage verlängert.

$$
\begin{aligned}
M_1 &= Alter^{12}_{(1,3,5,7,8,10,12),+1}\,(Group_{30}(Tage), Tage)\\
M_2 &= Alter^{12}_{2,-2}\,(M_1, Tage)\\
Monate &= Alter^{12*4}_{2*4,+1}\,(M_2, Tage)
\end{aligned}
$$

Die zweite Grundform von Operationen, die Selektionsoperationen, erzeugen neue Granularitäten durch „Selektieren" bestimmter Granules aus bestehenden Granularitäten. Die Operation $Select_{p_1,\ldots,p_2}(G, G_2)$ etwa, wählt die Granules an Positionen $p_1, \ldots, p_2$ der Granularität G_2 aus, die in einem Granule der Granularität G zusammengefasst sind. So beschreibt zum Beispiel der Ausdruck $Select_{1-5}(Woche, Tage)$ die Granularität *Arbeitswoche* durch Selektieren der Granules *Montag (1.)–Freitag (5.)* aus der Granularität *Woche*. In Bezug auf Periodizitätsschemas ermöglichen Selektionsoperationen die Auswahl bestimmter Zykluselemente aus einem Zyklus. So entspricht etwa die Gabe von Medikament B einer Chemotherapie ($Chemo = Group_{28}(Tage)$) am 1. und 8. Tag eines jeden Zyklus dem Ausdruck $Select_{1,8}(Chemo, Tage)$ oder die Angabe „*Jeden 1. Tag im Monat*" dem Ausdruck $Select_1(Monate, Tage)$.

In den Arbeiten zur algebraischen Beschreibung finden sich die oben dargestellen Grundoperationen Selektion und Gruppierung in unterschiedlichster Form wieder. In der *Calendar-Algebra* entsprechen diese Operationen den oben dargestellten Grundoperationen [3, 25]. Zusätzlich finden sich weitere spezialisierte Gruppierungs- und Selektierungsoperationen, wie etwa zur Berücksichtigung von Zeitverschiebungen.

Im *Slice-Formalismus* (z.B. [6, 24]) wiederum finden sich die Grundoperationen in Form eines normierten Operationsausdruck über mehrere Granularitätsebenen wieder. Ein solcher Ausdruck, d.h. ein Slice, hat die Form $O_n.G_n + \ldots + O_1.G_1$ wobei der einzelne Ausdruck $O_i.G_i$ jeweils eine Selektionsoperation für die Granularität G_i beschreibt, d.h. $O_i.G_i \equiv select_{O_i}(G_{i+1}, G_i)$. Es werden jeweils alle Granules der Granularität G_i selektiert, die in Bezug auf die Granularität G_{i+1} an den relativen, in O_i angegebenen Positionen stehen. Eine Granularität G_i wird als Menge von Intervallen verstanden, die durch verschiedene Gruppierungsoperationen gebildet werden können. Diese sind in ihrer Semantik identisch zu den Gruppierungsoperationen der Calendar-Algebra [3, 6]. Die Granularitäten des Ausdrucks sind geordnet und jede Granularität G_i besteht jeweils aus ganzzahligen Granules der Granularität G_{i-1}, wobei die Anzahl variieren kann (siehe *Monate*). Die einzelnen Ausdrücke sind hierbei stets „Und"-verknüpft. Der Ausdruck $*.Jahre + \{1, 2\}.Monate + \{1\}.Tage$ etwa entspricht der Angabe „*an jedem 1. Tag des 1. <u>und</u> 2. Monats (Jan, Feb) eines jeden Jahres*". Einen umfassenden Überblick sowie Vergleich der Ausdruckstärke des Slice-Formalismus als auch des Collection-Formalismus gegenüber der Calendar-Algebra (welche die beiden anderen subsumiert) findet sich in Bettini et al. [3].

Grundsätzlich ermöglicht das auf Gruppierungs- und Selektierungsoperationen basierende Vorgehen äquidistante und nicht-äquidistante Granularitäten bzw. Periodizitätsschemas zu beschreiben. Allerdings weisen die vorliegenden Vorschläge Schwächen im Umgang mit Ausnahmen in Periodizitätsschemas auf. So entspricht etwa die oben verwendete Operation *Alter*() lediglich Ausnahmen mit äquidistanter Elementauswahl. Ausnahmen mit nicht-äquidistanten Auswahlschema oder Zyklusauswahl werden jedoch nicht berücksichtigt, wie etwa bei der Angabe „*An jedem 3. Zyklus erfolgt am letzten Tag sowie an den ersten drei Tagen des darauffolgenden Zyklus eine zusätzliche Vitamingabe*". Dies stellt jedoch kein Problem des einzelnen Ansatzes dar, sondern ist eine inhärente Schwäche aller vorliegenden algebraischen Ansätze.

Bei der Erweiterung entsprechender Formalismen um weitere, nicht unterstützte Formen von Ausnahmen, erweist sich darüber hinaus die dargestellte Spezifikationslogik als hinderlich. Kennzeichnend für diese Logik ist, dass bei Ausnahmen sowohl die Art und Form des Auswahlschemas als auch die konkrete Änderungsoperation in einer einzelnen Operation „verschmolzen" wird, wie etwa bei *Alter*() mit äquidistanter Elementauswahl und Vergrößern/Verkleinern von Granules. Das allgemeine Vorgehen bei Ausnahmen, auf Basis eines Auswahlschemas bestimmte Elemente bzw. Zyklen zu „markieren" und gezielt mit Änderungsoperationen zu manipulieren, ist lediglich implizit in der Semantik der Operation hinterlegt. Eine Erweiterung entsprechender Formalismen unter Beibehaltung dieser „versteckten" Spezifikationslogik führt durch die möglichen Kombinationen von Änderungsoperationen, Auswahlschema und Auswahlart zu einer Vielzahl an Einzeloperationen. Neben dem Problem der Benutzerfreundlichkeit erweist sich die Vielzahl an Einzeloperationen, insbesondere bei einigen relevanten Realisierungsstrategien für Operationen, als hinderlich (siehe später). Eine explizite und generische Unterstützung von Ausnahmen wäre deshalb wünschenswert.

3.2 Constraint-basierte Ansätze

Die Beschreibung periodischer Zusammenhänge erfolgt bei dieser Klasse von Ansätzen durch logisch verknüpfte Terme (Prädikate, Constraints), die deklarativ die Eigenschaften eines Periodizitätsschemas festlegen. In der Literatur finden sich constraint-basierte Ansätze auf Basis Logiken erster Ordnung (z.B. [19, 33, 34]), temporaler Logiken (z.B. [10, 11, 17, 23]) und deduktiver Logiken (z.B. [35, 37]).

In einfachen Ansätzen beziehen sich die Terme unmittelbar auf einen zugrunde liegenden Wertebereich einer Granularität bzw. eines Periodizitätsschemas. Beispielsweise beschreibt das einstellige Prädikat $periodic_{m,p}(x)$ bzw. der Ausdruck $periodic_{28,1}(x) \vee periodic_{28,8}(x)$ die Gabe von Medikament B der Chemotherapie am 1. und 8. Tag eines Zyklus. Die Semantik solcher Constraints ist mit einer Menge linearer Geradengleichungen $m * x + p$ identisch. Das obige Beispiel kann somit als die Gleichungen $x = 28 * i + 1$ und $x = 28 * i + 8,\ i = 0, 1, 2, \ldots$ verstanden werden. Die erfüllenden Belegungen des Prädikats bzw. der Variable x – in diesem Fall $\{1, 8, 29, 36, \ldots\}$ – bilden die entsprechende Wertemenge der Granularität oder des Schemas. Kennzeichnend für diese Ansätze ist die ausschließliche Verwendung einer einzigen Granularität, nämlich die, welche mit dem zugrunde liegenden Wertebereich korrespondiert. Die Verwendung unterschiedlicher Granularitäten ist nur durch das explizite „Ausrechnen" der Längen von (äquidistanten) Granularitäten möglich. Die Gabe von Medikament B etwa kann in einem mit *Stunden* korrespondierenden Wertebereich durch den Ausdruck beschrieben werden: $periodic_{(24Stunden*28),1} \vee periodic_{(24Stunden*28),8}$.

Zur impliziten Beschreibung von Granularitäten sowie deren hierarchischen Abhängigkeiten werden Prädikate in Bezug auf eine zugrunde liegende abstrakte Struktur definiert. Die gewünschte Granularität entspricht genau der Struktur, die diese Prädikate erfüllt. Beispielsweise bildet eine solche Struktur die beiden (logischen) Mengen von Start- und Endpunkten der Granules einer Granularität [11]. Auf Basis der Prädikate werden implizit Eigenschaften zwischen Start- und Endpunkten bzw. Abhängigkeiten zwischen Granules unterschiedlicher Strukturen beschrieben. So beschreibt etwa das Prädikat $Group_p(G_1, G_2)$ eine Granularität G_2, in der jedes Granule aus einer Menge aufeinander folgender Granules aus der Granularität G_1 besteht. Das Prädikat entspricht lediglich einer Abkürzung für die konkreten logischen Formeln, die dessen formale Semantik auf Basis einer geeigneten Logik festlegen. Die Granularität *Tage* kann etwa durch den Ausdruck $Tage(\mathbf{x}) = Group_{24}(Stunden(\mathbf{y}), \mathbf{x})$ und die Granularität *Woche* mit $Woche(\mathbf{z}) = Group_7(Tage(\mathbf{x}), \mathbf{z})$ beschrieben werden. Der erste Ausdruck $Tage(x)$ ist ein Prädikat, das genau dann erfüllt ist, wenn eine Belegung der Variable x das Prädikat $Group_{24}(y, x)$ mit einer zweiten Struktur y erfüllt. Diese Struktur y wiederum muss zusätzlich das Prädikat $Stunden(y)$ erfüllen, d.h. y entspricht der Granularität *Stunden*. Die gleiche Semantik findet sich bei dem zweiten Audruck zur Beschreibung der Granularität *Woche*. Die erfüllenden Belegungen für die Variablen x und z bzw. die jeweiligen Strukturen entsprechen den Granularitäten *Tage* und *Woche*.

Die Semantik eines solchen Prädikats ist vergleichbar mit der Semantik der zuvor vorgestellten Gruppierungsoperation in algebraischen Ansätzen. Ein anderes Prädikat wiederum, nämlich $Group_pSkip_q(G_1, G_2)$, ist mit Selektionsoperationen vergleichbar. Es ist genau dann erfüllt, falls jedes Granule von G_2 aus genau p aufeinander folgenden Granules von G_1 besteht und bis zum nächsten Granule jeweils q Granules von G_1 ausgelassen werden. Die Granularität *Arbeitswoche* entspricht einer erfüllenden Belegung des Ausdrucks $Arbeitswoche(\mathbf{x}) = Group_5Skip_2(Tage(\mathbf{y}), \mathbf{x})$ bzw. die Gabe von Medikament A in einer Chemotherapie, d.h. in den ersten 14 Tagen eines jeden Zyklus, des Ausdruck $Chemo(\mathbf{x}) = Group_{14}Skip_{14}(Tage(\mathbf{y}), \mathbf{x})$.

Allerdings weisen vorliegende Vorschläge constraint-basierte Beschreibungen Schwächen im Umgang mit nicht-äquidistanten Granularitäten, wie etwa *Monate* bzw. bei Ausnahmen in Periodizitätsschemas, auf. Insbesondere die Anzahl der notwendigen Prädikate im Falle komplexer periodischer Spezifikationen bildet hier ein Problem. Zum Beispiel wird der Schaltmonat Februar durch explizite Auflistung aller Monatslängen innerhalb der Periode von 400 Jahren beschrieben [11]. Hierzu werden etwa im obigen Beispiel die Monatslängen, d.h. die Abstände zwischen den Start- und Endpunkten der Granules durch das Prädikat $M_n(x)$ beschrieben. Somit entspricht die Kette von 400 Prädikaten, d.h. $Monate(\mathbf{x}) = M_{31}(M_{28} \ldots (M_{31}(M_{29} \ldots (\mathbf{x}))))$, der Granularität *Monate*.

Diese Vorgehensweise zur Beschreibung nicht-äquidistanter Granularitäten ist kein Einzelfall, sondern findet sich in ähnlicher Form in allen vorliegenden constraint-basierten Ansätzen. Im Zusammenhang mit Ausnahmen in Periodizitätsschemas führt dieses Vorgehen dazu, dass das „Resultat" einer Ausnahme, d.h. die Menge der variierenden Abstände einer Periode, zunächst explizit „ausgerechnet" und dann deklarativ mit Prädikate beschrieben werden muss. Ein kompaktes Vorgehen im Sinne einer deklarativen Beschreibung der Änderungen für „ausgewählte" Zyklen, ohne eine damit verbundene explizite Auflistung des

Resultats bzw. entsprechender Prädikate, wird bisher nicht unterstützt.

Zusammenfassend betrachtet sind die vorliegenden algebraischen und constraint-basierten Vorschläge zur Beschreibung von Periodizitätsschemas nur sehr eingeschränkt tauglich. In keinem der Vorschläge findet sich eine adäquate und generische Unterstützung von Ausnahmen in Periodizitätsschemas. Das heißt, es fehlt sowohl eine einheitliche Beschreibung von äquidistanten und nicht-äquidistanten Auswahlschemas, die Unterstützung von Zyklen- bzw. Elementauswahl sowie die Bereitstellung generischer Änderungsoperationen.

Allerdings bildet die Erweiterung dieser Vorschläge, etwa um neue Operationen oder adäquate Prädikate, nicht das Hauptproblem. Das wahre Problem ist eine geeignete operationale Unterstützung für Periodizitätsschemas. Es werden Algorithmen benötigt, die effizient algebraische bzw. logische Ausdrücke evaluieren können. Entsprechende Algorithmen auch bei vollständiger Ausdrucksstärke von Schemas zur Verfügung zu stellen, entspricht jedoch keiner trivialen Erweiterung, wie im nachfolgenden Kapitel noch gezeigt wird.

4 Realisierungsstrategien für die operationale Unterstützung

Die Realisierung einer geeigneten operationalen Unterstützung für periodische Spezifikationen bildet eine der größten Herausforderungen. Wie in Abschn. 2.3 gezeigt, können die drei wesentlichen Klassen *Ableitungs-*, *Überprüfungs-* und *Vergleichsoperationen* unterschieden werden. Im folgenden Abschnitt werden verschiedene generelle Möglichkeiten der Realisierung dieser Operationsklassen vorgestellt sowie Vor- und Nachteile diskutiert.

4.1 Teilmaterialisierung

Eine Basisstrategie zur Realisierung von *Überprüfungs- und Vergleichsoperationen* stellt die Teilmaterialisierung einer periodischen Spezifikationen dar. Für einen vorgegeben begrenzten Zeitbereich, etwa dem Ereignishorizont bei Vergleichsoperationen (siehe Abschn. 2.3), werden hierzu konkrete zeitliche Werte der Spezifikation erzeugt. Diese expliziten Wertemengen werden dann gemäß der Operationssemantik entweder auf einzelne Werte überprüft oder auf Gleichheit, Ungleichheit oder sonstige Eigenschaften verglichen. Vorraussetzung für eine solche Strategie ist eine adäquate Realisierung von *Ableitungsoperationen* zur Erzeugung der Wertemengen ausgehend von der logischen Beschreibung (siehe später). Die Teilmaterialisierung entspricht im wesentlichen dem Vorgehen wie es etwa in heutigen Kalender- bzw. Zeitplanungssystemen überwiegend anzutreffen ist.

Bei einer Teilmaterialisierung ist es allerdings nicht sinnvoll, speziell bei großer Anzahl an Spezifikationen, für jeden einzelnen Vergleich eine „on-demand" Materialisierung zum Vergleichszeitpunkt vorzunehmen. Ein solches Vorgehen würde zu einem sehr hohen Aufwand pro Vergleich führen und erweist sich darüber hinaus für später noch benötigte Zugriffs- und Indexstrukturen als hinderlich. Stattdessen gilt es, die für einen begrenzten Zeitbereich relevanten Spezifikationen „vorzumaterialisieren". Zum Vergleichszeitpunkt müssen dann lediglich die vormaterialisierten Wertemengen verglichen werden. Grundvoraussetzung für diese Teilmaterialisierungstrategie ist aber eine (zeitliche) Synchronisation des aktuell gültigen (begrenzten) Zeitbereichs mit den im Zeitbereich relevanten periodischen Spezifikationen und ihren Wertemengen. Allerdings bedarf es aufgrund des Fortschreitens der Zeit und die damit verbundene „Verschiebung" des (begrenzten) Zeitbereichs einer kontinuierlichen Synchronisation. Das heißt, es müssen kontinuierlich „Nachmaterialisierung" und Löschungen von Werten vorgenommen werden, um die entsprechenden Wertemengen der (relevanten) periodischen Spezifikationen auf dem aktuellen Stand zu halten.

Die Teilmaterialisierung kann grundsätzlich die Realisierung von Überprüfungs- und Vergleichsoperationen gewährleisten. Allerdings weist die Teilmaterialisierung a priori verschiedene Nachteile und Einschränkungen auf:

1. Der bei einer kontinuierlichen Synchronisation anfallende Verwaltungsaufwand, entweder auf Seiten des Informationssystems oder in den zugrunde liegenden Datenbanken ist sehr hoch und aufgrund des Fortschreitens der Zeit beständig vorhanden. Ähnlich verhält es sich auch bei nachträglichen Änderungen an entsprechenden Spezifikationen (Updateproblem). Insbesondere im Zusammenhang mit einer umfangreichen Menge an periodischen Spezifikationen, wie etwa in prozessorientierten Informationssystemen, stellt dieser Verwaltungsaufwand eine kritische Größe dar.
2. Eine Teilmaterialisierung erlaubt es kaum, vorkommenden individuellen Abweichungen in der Größe des Zeitbereichs bzw. Ereignishorizonts zu begegnen. Einzelne über den Horizont hinausreichende Konfliktprüfungen, etwa bei wichtigen „VIP-Prozessen", können somit lediglich durch eine Nachmaterialisierung realisiert werden. Dies führt erneut zu einem sehr hohen Aufwand pro Vergleich.
3. Das Laufzeit- und Speicherverhalten dieser Realisierungsstrategie ist ummittelbar von der Größe der zeitlichen Begrenzung sowie der Anzahl enthaltener Werte abhängig. Je nach Anwendung, Art der periodischen Spezifikation sowie Größe der Begrenzung kann dieses Verhalten bereits schon für die einzelne Spezifikation,

insbesondere jedoch für die Gesamtheit an periodischen Spezifikationen kritische Werte erreichen.

Im Gegensatz zur Teilmaterialisierung, d.h. die zeitabhängige Repräsentation der periodischen Spezifikation durch dynamische Wertemengen, stellt eine „statische“ Repräsentation eine wünschenswertere Alternative dar. Das heißt die periodischen Spezifikationen werden zeitunabhängig repräsentiert, können jedoch zeitabhängig bezüglich eines Vergleichs abgefragt werden. Bei dieser Repräsentationsalternative entfällt somit der hohe Verwaltungsaufwand für dynamische Wertemengen und die weiteren damit verbundenen Probleme.

Im Folgenden werden zwei Ansatzklassen von zeitunabhängigen Repräsentationsformen und entsprechende Operationsrealisierungen vorgestellt: die Verwendung der formalen Beschreibung und die Verwendung von Wiederholungsmustern.

4.2 Verwendung der formalen Beschreibung

Die Grundlage der folgenden Realisierungsstrategien bildet die logische Beschreibung der periodischen Spezifikationen, d.h. die temporalen Operationen beziehen sich direkt auf die algebraischen Operationsketten oder logischen Formeln eines Periodizitätsschemas. Allerdings bedarf es für die algebraischen und constraint-basierten Beschreibungen geeigneter Evaluationsstrategien, so dass die verschiedenen Operationsklassen auch effizient realisiert werden können.

4.2.1 Algebra- bzw. symbolbasierte Realisierung

Diese Realisierung basiert auf der Auswertung von Operationsketten der algebraischen Beschreibung von Periodizitätsschemas.

Ableitungs- und Überprüfungsoperationen: Die Grundidee ist, „Umrechnungsoperationen“ für jede einzelne Beschreibungsoperation bereitzustellen. Diese definieren explizit, wie Angaben von Zyklenelementen des „Ausgabeschemas“ in Werte des zugrunde liegenden „Eingabeschemas“ einer Beschreibungsoperation umgerechnet werden können und umgekehrt. Diese Umrechnungen der einzelnen Beschreibungsoperationen bilden die Grundlage entsprechender Operationen über dem gesamten algebraischen Ausdruck eines Periodizitätsschemas.

Bei Ableitungsoperationen gilt es zum Beispiel für Gruppierungsoperation wie etwa $G = Group_m(H)$, das Intervall von Werten aus H zu ermitteln, aus dem sich ein gegebener Wert in G zusammensetzt. Bei der „Umrechnung“ $G(i) = [H_{\text{first}}, H_{\text{last}}]$, d.h. von einem Wert i aus G, ergibt sich für das Intervall in H als erster Wert $H_{\text{first}} = i * m$ bzw. als letzter Wert $H_{last} = H_{\text{first}} + (m-1)$. So entspricht der 3.Tag der Granularität $Tage = Group_{24}(Stunden)$ dem Intervall $Tage(3) = [72, 95]$ in *Stunden*. Für die weiteren Gruppierungs- und Selektierungsoperationen können vergleichbare, wenn auch komplexere Umrechnungen gefunden werden. Ausgehend von einer konkreten Angabe zu einem Zykluslelement bzw. Granule können entsprechende Umrechnungen rekursiv für alle Beschreibungsoperationen durchgeführt werden, solange bis die „tiefste“ mit dem Wertebereich korrespondierende Ebene erreicht ist. Auf dieser Ebene, d.h. auf Basis der Umrechnung der „tiefsten“ bzw. „ersten“ Beschreibungsoperation, können konkrete Werte aus dem zeitlichen Wertebereich abgeleitet werden. Im Fall von *Stunden* als tiefste Ebene ergibt sich für das Beispiel $Woche(3)$ mit $Woche = Group_7(Tage)$ das Intervall $[Tage(21), Tage(27)]$ auf Stundenebene bzw. im Wertebereich der Intervall $[504, 671]$.

Bei Überprüfungsoperationen wird das gleiche Vorgehen verwendet, jedoch in Gegenrichtung, d.h. ausgehend von Angaben des „Eingabeschemas“ in Angaben des „Ausgabeschemas“ einer Beschreibungsoperation. Allerdings ist bei Periodizitätsschemas bzw. bei Granularitäten mit „Löchern“ nicht jeder Wert des Wertebereichs einem Zykluselement bzw. Granule der Granularität zuordenbar. Beispielsweise sind zeitliche Werte, die mit *Samstag* oder *Sonntag* korrelieren, nicht in der Granularität *Arbeitswoche* enthalten. In solchen Fällen schlägt die Umwandlung an einer Stelle in der algebraischen Operationskette fehl. Nur wenn bis zur „obersten Ebene“ bzw. letzten Beschreibungsoperation stets eine Umwandlung des Wertes gefunden werden kann, bedeutet dies, dass der Wert in dem Periodizitätsschema bzw. Granularität enthalten und die Überprüfungsoperation erfolgreich ist.

Die symbol-basierte Realisierung bildet eine adäquate Strategie für Ableitungs- und Überprüfungsoperationen von periodischen Spezifikationen. Auch wenn diese ausschließlich auf dem algebraisch beschriebenen Periodizitätsschema basiert, können die zusätzlichen Aspekte einer Spezifikation, d.h. Gültigkeitszeitraum und individuelle Instanziierungszeitpunkte, durch entsprechende Vor- bzw. Nachbereitungsschritte berücksichtigt werden.

Allerdings sind bei einer symbol-basierten Realisierung im Fall großer Mengen periodischer Spezifikationen möglichst effiziente Einzelumrechnungen zwingend notwendig. Des Weiteren müssen hierbei auch eventuell bestehende äquivalente oder redundante Umrechnungen in Bezug auf den gesamten algebraischen Ausdruck erkannt und eingespart werden können. Die in der Literatur vorliegenden Vorschläge zur symbol-basierten Realisierung von algebraischen Beschreibungen (z.B. [3, 25, 26]) weisen in diesem Zusammenhang Schwächen auf. Aufgrund der „semantisch reichhaltigen“ Beschreibungsoperationen (siehe Abschn. 3.1), d.h. der „implizit“ in der Operation angenommenen Schritte zur Bildung des Ausgabeschemas, ergeben sich ebenfalls komplexe Umrechnungen. So basiert etwa in der Operation *Alter*() die Umrechnung auf einem Näherungsverfahren, das zwar korrekt, jedoch sehr aufwendig in seiner Ausführung ist (z.B. [3, 25]). Darüber hinaus liegt häufig eine Vielzahl an einzelnen Beschreibungsoperationen mit vergleichbarer bzw. ähnlicher Semantik, jedoch jeweils eigenen Einzelumrechnungen, vor. In Bezug auf den gesamten algebraischen Ausdrucks eines Periodizitätsschemas ermöglichen einzelne komplexe Umrechungen oder eine Vielzahl separater Einzelumrechnungen es kaum, äquivalente oder redundante Schritte bzw. Umrechnungen zu erkennen und einzusparen.

Vergleichsoperationen: Ein algebraischer Realisierungsansatz für Vergleichs- bzw. Mengenoperationen ist vergleichbar mit dem vorherigen Vorgehen bei Ableitungs- und Überprüfungsoperationen (z.B. [29, 32]). Es werden für jede Beschreibungsoperation formale qualitative Beziehungen zwischen dem „Ein-" und „Ausgabeschema" angegeben. So führt die Selektionsoperation *Select* zu einer Teilmengenbeziehung des Ein- und Ausgabeschemas, wie beispielsweise der Ausdruck $Arbeitswoche = Select_{1-5}(Woche, Tage)$ zur Beziehung $Arbeitswoche \subseteq Woche$. Ein weiteres Beispiel bilden die Angaben „*Jeden Montag*" ($Mo = select_1(Woche, Tage)$) und „*Jeden 2. Montag*" ($2tenMo = select_1(Group_2(Mo), Tage)$) mit der Beziehung $2tenMo \subseteq Mo$.

Bei einer Mengenoperation zwischen zwei Schemas werden deren unmittelbar oder transitiv ableitbaren Beziehungen ermittelt. Für jede Mengenoperation sind Regeln hinterlegt, wie basierend auf den qualitativen Beziehungen der Operanden das Ergebnis der Operation unmittelbar abgeleitet werden kann. Beispielsweise kann bei einer Schnittoperation und einer Teilmengenbeziehung beider Operanden G_1, G_2 das Ergebnis durch die Regel $G_1 \subseteq G_2 \Rightarrow G_1 \cap G_2 = G_2$ abgeleitet werden bzw. bei einer Vereinigungsoperation durch $G_1 \cup G_2 = G_2$. So führt im oberen Beispiel eine Mengenoperation zwischen „*Jeden Montag*" und „*Jeden 2. Montag*" aufgrund der Teilmengenbeziehung $2tenMo \subseteq Mo$ zu dem Ergebnis $Mo \cap 2tenMo = 2tenMo$ bzw. $Mo \cup 2tenMo = Mo$.

Allerdings ist dieses Vorgehen nicht in jedem Fall anwendbar, da sich nicht in jedem Fall geeignete Beziehungen zwischen algebraischen Ausdrücken herleiten lassen. Insbesondere gilt dies bei Ausdrücken mit teilweise „überlappenden" Wertemengen, wie etwa zwischen der Angabe „*Jeden Montag*" und „*Jeden 1. Tag im Monat*". In solchen Fällen reichen qualitative „Vereinfachungsregeln" nicht aus, um das konkrete Ergebnis einer Mengeoperation der beiden überlappenden Wertemengen anzugeben. Genauer gesagt, es müsste jede mögliche Überlappung zweier Mengen durch eine entsprechende qualitative Beziehung beschrieben werden, was jedoch kaum realisierbar ist.

Darüber hinaus erlaubt diese Realisierungsmöglichkeit lediglich kalender-orientierte Spezifikationen zu vergleichen. Aufgrund der individuellen Instanziierungszeitpunkte zyklen-orientierter Spezifikationen können identische Operationsketten sowohl gleiche, disjunkte oder überlappende Wertemengen bilden. Dies jedoch alleine anhand der Operationsketten erkennen zu können, ist nicht möglich. Für zyklen-orientierte Spezifikationen ist eine solche Vorgehensweise deshalb nicht anwendbar.

Eine alternative Möglichkeit der algebraischen Realisierung von Vergleichsoperationen basiert auf dem Slice-Formalismus (siehe Abschn. 3.1). Ein Slice ist ein Ausdruck der Form $O_n.G_n + ... + O_1.G_1$ wobei $O_i.G_i$ jeweils eine Selektionsoperation für die Granularität G_i beschreibt. Der Ausdruck $*.Jahre + \{1, 2\}.Monate + \{1\}.Tage$ etwa entspricht der Angabe „*an jeden 1. Tag des 1. <u>und</u> 2. Monats (Jan, Feb) eines jeden Jahres*".

Die Realisierung von Mengenoperationen zwischen zwei Sliceausdrücken basiert auf Mengenoperationen zwischen den Mengen O_i, O_k identischer Teilausdrücke $O_i.G_j$ und $O_k.G_j$. Das Ergebnis einer Schnittoperation etwa zwischen $*.Monate + \{1, 8, 10\}.Tage$ und $*.Monate + \{3, 8, 9\}.Tage$ ist der Ausdruck $*.Monate + \{8\}.Tage$. Bei Ausdrücken in denen die jeweiligen Granularitätsreihenfolgen nicht zueinander „kompatibel" sind, bedarf es zusätzlicher Umwandlungen, wie zum Beispiel bei den Ausdrücken $Monate + 5.Tage$ und $Monate + Woche + 1.Tage$. Der erste Ausdruck beschreibt die Angabe „*jeden 5. Tag eines Monat*" und der zweite Ausdruck die Angabe „*jeden Montag*". Die Wertemengen beider Ausdrücke überlappen sich teilweise. Zur Erkennung der gemeinsamen Werte muss die Beziehung zwischen *Wochen* und *Monaten* explizit durch eine entsprechende Umwandlung bzw. Erweiterung von einem der beiden Ausdrücke realisiert werden. Im obigen Beispiel kann der zweite Ausdruck umgewandelt werden in $Monate + \{\}.Tage$ und innerhalb der Periode von *Wochen* und *Monaten* (28 Jahre) erweitert bzw. explizit aufgeführt. Es wird eine Menge von Ausdrücken der Form: $1.Monate + 3.Tage \cup 2.Monate + 6.Tage \cup \cdots \cup 336.Monate + 6.Tage$ erzeugt. Dieser erweiterte Ausdruck wird dann mit dem ersten Ausdruck $Monate + 5.Tage$ geschnitten.

Die Realisierung von Vergleichsoperationen basierend auf normierten Ausdrücken für Selektionsoperationen ist bei kalender-orientierten Spezifikationen grundsätzlich möglich. Allerdings kann dabei (wie oben gezeigt) eine sehr umfangreiche Anzahl an Ausdrücken entstehen, was sich bezüglich der Effizienz des Verfahrens als großer Nachteil erweist. Allerdings bildet das Kernproblem dieser Vorgehensweise erneut die Tatsache, dass lediglich kalender-orientierte Spezifikationen unterstützt werden können. Die individuellen Instanziierungszeitpunkte von zyklen-orientierten Spezifikationen bzw. die resultierenden Probleme (siehe oben) können auch mit dieser algebraischen Realisierungsmöglichkeit nicht begegnet werden. Die praktische

Anwendbarkeit der vorliegenden Vorschläge (z.B. [6, 24]) ist daher stark beschränkt.

4.2.2 Constraint-basierte Realisierung

Eine Realisierung von Operationen auf Basis der constraint-basierten Beschreibung von periodischen Spezifikationen basiert im Wesentlichen auf der Überprüfung der Erfüllbarkeit entsprechender logischer Ausdrücke.

Überprüfungsoperationen: Zur Realisierung dieser Operation gilt es für einen gegebenen zeitlichen Wert zu überprüfen, ob dieser eine erfüllende Belegung für den logischen Ausdruck der periodischen Spezifikation darstellt. Es hängt von der Art der logischen Terme bzw. der verwendeten Logik ab, ob eine solche Überprüfung grundsätzlich effizient erfolgen kann.

Bei der Verwendung einfacher logischer Terme kann die Überprüfung auf Basis der semantischen Äquivalenz zu linearen Geradengleichungen erfolgen. Ein gegebener Wert x für den Ausdruck $periodic_{28,1}(x) \vee periodic_{28,8}(x)$ muss eine Lösung der Gleichungen $x = 28 * i + 1$ oder $x = 28 * i + 8$ darstellen. Bei entsprechender Umstellung der Gleichungen ist dies effizient überprüfbar. Darüber hinaus ermöglichen die Gleichungen es auch, konkrete Werte einer periodischen Spezifikation abzuleiten. Im Gegensatz zur allgemeinen Realisierung von Ableitungsoperationen (siehe unten) ist dies bei einfachen logischen Ausdrücken somit grundsätzlich möglich.

Allerdings, wie bereits in Abschn. 3.2 erwähnt, reichen einfache logische Ausdrücke nicht aus, unterschiedliche Granularitäten adäquat zu unterstützen. Zur (impliziten) Unterstützung unterschiedlicher Granularitäten werden erweiterte Logiken verwendet. Im Gegensatz zu den oberen „einfachen“ Logiken, weist jedoch die Überprüfung erfüllender Belegungen in den erweiterten Logiken grundsätzlich nicht triviale Komplexitäten auf, wie z.B. PSPACE [11].

Eine Realisierung von Überprüfungsoperationen ist lediglich für „einfache“ constraint-basierte Beschreibungen, d.h. Ansätze mit expliziter Unterstützung von Granularitäten, grundsätzlich möglich (z.B. [33, 34]). Erweiterte Logiken, d.h. Ansätze mit impliziter Granularitätsunterstützung, sind aufgrund der Komplexität der Realisierung nicht geeignet (z.B. [10, 11, 17]).

Ableitungs- und Vergleichsoperationen: Bei diesen Operationen basiert die Realisierung auf dem Finden erfüllender Belegungen für die logischen Ausdrücke periodischer Spezifikationen. So entspricht beispielsweise die Vergleichs- bzw. Schnittoperation (Konjunktion) der „*Gabe von Medikament B*“ und „*Jede 5. Tag*“ dem Finden von Werten x, welche den logischen Ausdruck erfüllen: $(periodic_{28,1}(x) \vee periodic_{28,8}(x)) \wedge periodic_{5,1}(x)$.

Im Allgemeinen sind für das Finden erfüllender Belegungen (Erfüllbarkeitsproblem) lediglich Verfahren mit nicht-polynomialen Komplexitäten bekannt, die für den praktischen Einsatz nicht geeignet sind. Jedoch ermöglichen spezielle Verfahren wie etwa *Temporal Constraint Networks* (TCN) für bestimmte Formen von (zeitlichen) Constraints, die Erfüllbarkeit effektiver zu entscheiden sowie eine Lösungsmenge anzugeben. In diesem Zusammenhang finden sich in der Literatur Vorschläge zur Auswertung contraint-basierter Beschreibungen von periodischen Spezifikationen bzw. Granularitäten auf Basis von TCN (z.B. [1, 3, 28, 30–32]).

Allerdings weist die Vorgehensweise bei allen vorliegenden Vorschlägen ein grundsätzliches Problem auf. In diesen Arbeiten werden die „periodischen“ Constraints wie etwa $periodic_{28,1}(x)$ nicht unmittelbar ausgewertet. Stattdessen werden solche Constraints lediglich für einen begrenzten Zeitbereich in „klassische“ Zeitconstraints (Abstandsconstraints) „umgewandelt“ bzw. expandiert. Dieses Vorgehen entspricht jedoch dem Vorgehen der Strategie der Teilmaterialisierung von Werten. Anstelle konkreter Werte wird eine Menge an Abstandsconstraints in einem bestimmten Zeitbereich materialisiert. Somit weist ein solches Vorgehen die gleichen Probleme wie eine Teilmaterialisierung auf.

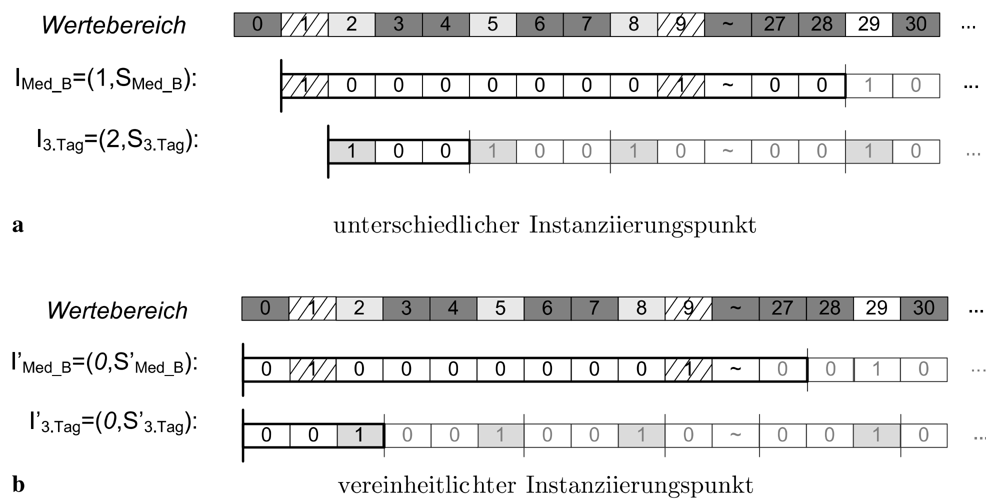
4.3 Musterbasierte Realisierungsformen

Die Grundlage dieser Realisierungsform bilden Wiederholungsmuster, die auf Basis der logischen Beschreibung von periodischen Spezifikationen erzeugt werden (z.B. in [4, 5] am Beispiel der Calender Algebra [25] gezeigt). Ein Wiederholungsmuster repräsentiert die kleinste Periode einer Spezifikation, d.h. den kleinsten Zeitraum der Spezifikation, nach dem sich die möglichen Abstände zwischen Werten sowie deren Reihenfolgen wiederholen.

Ein Wiederholungsmuster „verläuft“ über einen (diskreten) Wertebereich, wobei die Positionen des Musters mit den Werten des Wertebereichs korrespondieren. Die konkreten Belegungen der Positionen, d.h. die Stützstellen des Musters, treffen Aussagen, ob der zur Position korrespondierende Wert in der Spezifikation enthalten ist (oder nicht). Die Länge des Musters, d.h. die Periodenlänge, entspricht der kleinsten Periode der Spezifikation. Beispielsweise ist in Abb. 1a eine Bitvektordarstellung der Wiederholungsmuster für die Beispiele „*Gabe vom Medikament B* (S_{Med_B})“ (Periodenlänge = 28, Instanziierungspkt. = 1) und „*Jeden 3. Wert* ($S_{3.Tag}$)“ (Periodenlänge = 3, Instanziierungspkt. = 2) dargestellt.

Ein Vorteil muster-basierter Realisierungen, insbesondere in Hinblick auf Vergleichsoperationen, bildet die Möglichkeit unterschiedliche Instanziierungszeitpunkte der Spezifikationen „herauszurechnen“ bzw. zu vereinheitlichen. Ermöglicht wird dies durch eine Verschiebung auf

Abb. 1 Bitvektordarstellung zweier Wiederholungsmuster



einen gewünschten einheitlichen Instanziierungszeitpunkt bei gleichzeitiger zyklischen Rotation des Musters (jeder Wert der rechts „heraus fällt", wird links wieder eingefügt). Die Rotation ist werteäquivalent, d.h. die Wertemenge des rotierten Musters ausgehend von dem vereinheitlichten Instanziierungspunkt ist identisch zur ursprünglichen Wertemenge. Der ursprüngliche Instanziierungszeitpunkt dient später lediglich als untere Schranke beispielsweise für Ableitungsoperationen. In Abb. 1b ist die Vereinheitlichung auf den Instanziierungspkt = 0 für die vorherigen Beispiele aus Abb. 1a dargestellt.

Allerdings können lediglich Spezifikationen mit periodischen Ausnahmen durch Wiederholungsmuster repräsentiert werden. Spezifikationen mit nicht-periodischen Ausnahmen, d.h. Ausnahmen, bei denen nur eine bestimmte endliche Menge an Zyklen oder Elemente betroffen sind, können mit Wiederholungsmustern nicht dargestellt werden. Entsprechende Spezifikationen weisen keine kleinste Periode auf, d.h. es kann kein konstantes einzelnes Muster angeben werden, das sich über den gesamte Zeitbereich wiederholt. Nicht-periodische Ausnahmen erfordern in diesem Fall eine andere Repräsentation bzw. eine separate Behandlung.

Die Realisierung von Operationen ist abhängig von der konkreten Repräsentation des Musters. In der Literatur finden sich zwei grundsätzliche Repräsentationsformen, die im Folgenden vorgestellt werden.

4.3.1 Lineare Musterrepräsentation

Bei dieser Repräsentationsform wird ein Wiederholungsmuster explizit durch seine erste Periode dargestellt. So finden sich Vorschläge zur Repräsentation des Wiederholungsmuster als einfache (äquidistante) lineare Geradengleichungen (z.B. [2, 19–21, 26, 33]) oder als Bit- bzw. Stringvektor (z.B. [4, 5, 36]).

Ableitungs- und Überprüfungsoperationen: Unabhängig von der konkreten Repräsentation im Einzelnen, kann aufgrund der Äquivalenz zu einfachen mathematischen Strukturen die Realisierung solcher Operationen stets als „Berechnung" zum Beispiel einer Menge linearer Gleichungen verstanden werden. So kann das Wiederholungsmuster I'_{Med_B} aus Abb. 1b durch die linearen Gleichungen $g_1 : x = 28 * i + 2$ und $g_2 : x = 28 * i + 9$ repräsentiert werden (siehe Abschn. 4.2.2). Neben der Ableitungsoperation auf Basis der Indexvariable i kann bei entsprechender Umstellung der Gleichungen auch die Überprüfungsoperation effizient realisiert werden.

Vergleichsoperationen: Generell gilt es beim Vergleich von Wiederholungsmustern zwei Fälle zu unterscheiden: Mustern mit identischer und mit unterschiedlicher Periodenlänge.

Bei identischer bzw. vielfacher Periodenlänge können die beiden bezüglich des Instanziierungszeitpunkts vereinheitlichten Wiederholungsmuster unmittelbar miteinander verglichen werden. Das heißt die jeweiligen Stützstellen des Wiederholungsmusters werden gemäß der Semantik des Vergleichs direkt miteinander verglichen. Hierbei kann ebenfalls leicht ein Ergebnismuster für die quantitative Vergleichsauswertung angegeben werden.

Für Wiederholungsmuster mit unterschiedlicher Periodenlänge findet sich, unabhängig von der konkreten Repräsentation, stets das gleiche Vorgehen. Es wird die *kleinste gemeinsame Periodenlänge* (least common period – LCP) der beiden Wiederholungsmuster gebildet. Das LCP entspricht dem kleinsten gemeinsamen Vielfachen der Periodenlängen der Muster und ist der Zeitbereich, den beide Muster nach ganzzahliger Anzahl von Wiederholungen durchlaufen haben. Die Muster werden innerhalb des LCP erweitert bzw. explizit aufgeführt. Ziel des Vorgehens ist es, die unterschiedlichen Periodenlängen so zu verlängern, dass erneut identische Periodenlängen entstehen. Nach der

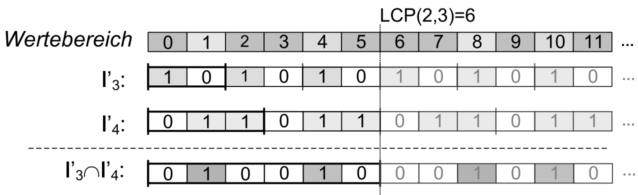


Abb. 2 Schnittoperation von Wiederholungsmustern mit unterschiedlicher Periodenlänge

Verlängerung kann die gleiche Realisierung wie im ersten trivialen Fall verwendet werden. In Abb. 2b sind zwei Muster mit Periodenlänge 3 bzw. Länge 2 dargestellt, sowie das Ergebnismuster der Schnittoperation mit Periodenlänge $6 = \mathrm{LCP}(2, 3)$.

Das Vorgehen linearer Musterrepräsentationen gewährleistet es grundsätzlich, sowohl Ableitungs- und Überprüfungsoperationen als auch Vergleiche zu realisieren.

Allerdings hängt es von der Art der periodischen Spezifikation ab, ob die Realisierung im Einzelfall effizient erfolgen kann. Bei linearen Musterrepräsentationen ist die Effizienz der operationalen Unterstützung unmittelbar von der Größe des Musters abhängig, d.h. von der Periodenlänge und der Anzahl an Stützstellen. Bereits bei einfachen jedoch praxisrelevanten Beispielen ergeben sich Größenordnungen, die eine effiziente Auswertung kaum ermöglichen. Beispielsweise beträgt die Periodenlänge der Angabe „*Jeden 1. Tag im Monat*“ 400 *Jahre* (146097 *Tage*). Die Anzahl konkret belegter Stützstellen beträgt 4800 Werte (12 pro Jahr $\times$ 400 Jahre). Diese Einzelangabe entspricht einer Menge von 4800 linearen Gleichungen bzw. einem Bitvektor der Länge 146097 Bits ($\approx$ 4500 Integerwerte mit 32 Bit). Bei einem Vergleich der oberen Angabe mit einer Chemotherapie der Länge von 28 Tagen beträgt das $\mathrm{LCP}(28, 146097) = 584.388$ *Tage*, d.h. das Ergebnismuster weist eine Periodenlänge von ca. 2000 Jahren auf.

4.3.2 Automaten-basierte Repräsentation

Die Realisierung von Operationen wird in diesen Ansätzen auf die Theorien formaler Sprachen und Automaten zurückgeführt. Ein Muster wird als ein Wort einer Sprache verstanden, welche durch den Automaten repräsentiert wird. Die Automatendarstellung des Musters bildet die Grundlage für die Operationsrealisierung.

In der Literatur finden sich Ansätze, die ein Wiederholungsmuster bzw. jede Position des Muster durch einen entsprechenden Zustand im Automaten repräsentieren (z.B. [7, 15]). Die Anzahl der Zustände korreliert hierbei mit der Periodenlänge des Musters. Die Symbole des Musters bilden die Zustandsübergangsfunktion.

Allerdings finden sich auch Vorschläge, die eine kompaktere Darstellung des Wiederholungsmusters ermöglichen (z.B. [15, 16]). Hierbei werden unterschiedlich priorisierte Zustandsübergangsfunktionen verwendet, die Regelmäßigkeiten innerhalb des Wiederholungsmuster ausnutzen und deshalb mit wenigeren Zuständen auskommen. Die Semantik dieser Übergangsfunktionen ist mit Zählern der Art „*5 mal Symbol 0; 2 mal Symbol 1*“ vergleichbar, d.h. entspricht einem automaten-basierten „*run length encoding*“ des Musters.

Ableitungs- und Überprüfungsoperationen: Die Realisierung solcher Operationen basiert auf dem Simulieren des Automaten. Ausgehend von dem Instanziierungspunkt einer periodischen Spezifikation wird das Muster auf Basis des Automaten „durchlaufen“. Hierbei werden entweder konkrete Werte abgeleitet oder überprüft, ob sich ein entsprechender Wert ergibt.

Grundsätzlich ermöglicht eine Simulation des Automaten die Realisierung entsprechender Operationen. Allerdings handelt es sich um ein sehr aufwendiges Vorgehen, das bereits bei einer einzelnen Angabe Effizienzprobleme aufweist. Darüber hinaus entsprechen nicht-kompakte Automatendarstellungen einer Form der linearen Musterrepräsentationen und weisen die gleichen Probleme wie im vorherigen Abschnitt besprochen auf.

Vergleichsoperationen: Im Fall von nicht-kompakten Automatendarstellung und ihrer Äquivalenz zu endlichen Automaten können grundsätzlich Algorithmen zur Realisierung von Vergleichs- bzw. Mengenoperationen angeben werden. Zur Konstruktion des Ergebnisautomaten für den Schnitt bzw. die Vereinigung zweier Automaten wird zunächst der Kreuzproduktautomat gebildet, d.h. das Kreuzprodukt der beiden Zustandsmengen. Anschließend werden, je nach Semantik der Operation, die Endzustände angepasst. Allerdings fehlen entsprechende Betrachtungen bzw. Aussagen zur Realisierbarkeit solcher Operationen in Bezug auf kompakte Automatendarstellungen.

Die automaten-basierte Realisierung von Vergleichsoperationen ist mit existierenden Vorschlägen nur im Fall nicht-kompakter Automatendarstellung möglich. Allerdings sind der Umfang der Darstellung sowie das Vorgehen bei Vergleichsoperationen vergleichbar mit linearen Musterrepräsentationen. So entspricht die Bildung des Kreuzproduktautomaten im Kern der Bildung des LCP zweier Muster. Das heißt, die resultierende Anzahl von Zuständen des Produktautomaten ist identisch mit dem Produkt bzw. LCP der beiden Periodenlängen des Musters. Jede der Aussagen zu linearen Musterrepräsentationen gilt deshalb mindestens genauso stark für eine entsprechende Automatendarstellung. Aufgrund der aufwendigeren Graphdarstellung der Automaten sind entsprechende Aussagen eher noch schwerwiegender.

Die vollständige operationale Unterstützung von periodischen Spezifikationen kann durch eine musterbasierte Realisierung grundsätzlich gewährleistet werden. Allerdings fehlt es an adäquaten Ansätzen im Zusammenhang mit Wiederholungsmustern mit sehr langen Periodenlängen. Aufgrund des Repräsentationsumfangs und den damit verbundenen operationalen Aufwand sind vorliegende Vorschläge nicht geeignet.

In Hinblick auf die Praxis erweist sich dieses Ergebnis als kritisch, da Muster mit sehr langen Periodenlängen im Allgemeinen nicht die Ausnahme sondern vielmehr die Regel bilden. Periodische Angaben auf Basis der Granularität *Monat* sind häufig und überaus gebräuchlich in der Praxis. Darüber hinaus können Muster mit langen Periodenlängen relativ leicht auch durch die Berücksichtigung von Ausnahmen in eigentlich „kurzen" Mustern wie der Chemotherapie entstehen. Das Vorgehen zur Bildung des Wiederholungsmusters bei periodischen Spezifikationen mit Ausnahmen ist vergleichbar mit der Bildung des LCP bei Vergleichsoperationen. Beispielsweise führt die Ausnahme „*Zusätzlicher Labortest nach jedem 3. Zyklus*" zu einem Wiederholungsmuster der Länge 84 (3×28 Tage $= 84$ Tage). Eine weitere zusätzliche Ausnahme, wie etwa das „*Einfügen einer Pause von 5 Tagen nach jedem 2. Zyklus*", führt bereits zu einem Muster der Länge 178 ($2 \times 84 + 2 \times 5$ Pause $= 178$ Tage). Weitere Ausnahmen an anderen Zyklen oder komplexere Auswahlschemas führen zu weiteren Verlängerungen der Periodenlänge, so dass auch in diesem Fall extrem lange Periodenlängen entstehen können.

5 Zusammenfassung

Wir haben in diesem Beitrag einen detaillierten Einblick in die Herausforderungen sowie einen Überblick über die in der aktuellen Literatur vorliegenden Lösungsansätze einer systemseitigen Unterstützung von periodischen Spezifikationen gegeben.

In Hinblick auf die benötigte Ausdrucksstärke zeigte sich, dass die generische Unterstützung von Ausnahmen in periodischen Spezifikationen eine der wichtigsten Anforderungen darstellt. Erst die umfassende Beschreibung periodischer oder nicht-periodischer Abweichungen bzw. Änderungen von einem vorgegebenen Grundschema ermöglicht es, praxisrelevante Anforderungen zu erfüllen. Darüber hinaus bilden Ausnahmen die Grundvorraussetzung für die Verwendung unterschiedlicher Granularitäten und damit verbundene Kalenderphänomene (z.B. Sommer-/Winterzeit bei Fahrplänen).

Allerdings stellt die adäquate und generische Unterstützung von Ausnahmen ein großes Manko vorliegender Vorschläge für Beschreibungsformalismen dar. In keinem der in der Literatur vorliegenden Ansätze finden sich adäquate Möglichkeiten, einheitlich periodische oder nicht-periodische Ausnahmen zu beschreiben. Entweder sind in den entsprechenden Formalismen Ausnahmen grundsätzlich nicht berücksichtigt oder aber nur in sehr einfachen Ausprägungen.

Ein weiterer wichtiger Aspekt und der Schwerpunkt des Beitrages bildete die operationale Unterstützung von periodischen Spezifikationen. Neben den Ableitungs- und Überprüfungsoperationen erwiesen sich hier insbesondere die Vergleichsoperationen als eine der größten Herausforderungen. Neben der Basisstrategie der Teilmaterialisierung von periodischen Spezifikationen wurden zwei alternative Ansatzklassen von Realisierungsstrategien für Operationen vorgestellt. Im Gegensatz zur Teilmaterialisierung sind die alternativen Ansatzklassen a priori nicht bzw. nicht im gleichen Maße durch einen hohen Verwaltungsaufwand aufgrund umfangreicher dynamischer Wertemengen oder bei Updates gekennzeichnet.

Die alternative Ansatzklasse, basierend auf der logischen Beschreibung von Spezifikationen, konnte allerdings nur bei Ableitungs- und Überprüfungsoperationen den Anforderungen der operationalen Unterstützung begegnen. Für Vergleichsoperationen hingegen stellt diese Ansatzklasse aufgrund ihrer sehr eingeschränkten Unterstützung einiger weniger Formen von periodischen Spezifikationen keine praktisch anwendbare Realisierungsalternative dar. Lediglich die dritte Ansatzklasse, d.h. die musterbasierte Realisierung, konnte den Anforderungen der operationalen Unterstützung grundsätzlich begegnen. Die Verwendung von Wiederholungsmustern als Repräsentant der (theoretisch) unendlichen Wertemenge einer periodischen Spezifikation stellte eine alternative Möglichkeit für die Realisierung sowohl von Ableitungs- und Überprüfungsoperationen als auch von Vergleichsoperationen dar.

Die prinzipielle Realisierbarkeit von Operationen alleine reicht jedoch nicht aus. Insbesondere im Fall einer großen Menge an periodischen Spezifikationen muss auch die Effizienz der resultierenden Implementierung berücksichtigt werden. Wie gezeigt wurde, kann hier weder eine Teilmaterialisierung noch die vorliegenden Vorschläge zur musterbasierten Realisierung den Anforderungen im vollen Umfang begegnen. Das Laufzeit- und Speicherverhalten einer Teilmaterialisierung kann je nach Anwendung, Art der periodischen Spezifikation sowie Größe der Begrenzung sowohl für die Einzelne als auch für die Gesamtheit an Spezifikationen kritische Werte aufweisen. Im Gegensatz dazu fehlt es bei der musterbasierten Realisierung an adäquaten Ansätzen für Wiederholungsmustern mit sehr langen Periodenlängen. Aufgrund des Repräsentationsumfangs und dem damit verbundenen operationalen Aufwand sind die vorliegenden Vorschläge nicht geeignet. Kompaktere Repräsentationen – unter Beibehaltung der Vergleichbarkeit – sind sowohl bezüglich des Speicherbedarfs des Einzelmusters als auch bezüglich der Effizienz der operationalen Unterstützung für den praktischen Einsatz unverzichtbar.

In der Literatur ist die Bedeutung einer integrierten Unterstützung von Periodizität in Informationssystemen grundsätzlich erkannt worden. Allerdings bedarf es noch weiterer intensiver Forschungsarbeit, bei der sowohl umfassende Beschreibungsmöglichkeiten als auch die Effizienz der operationalen Unterstützung wichtige offene Punkte darstellen. Erst die Addressierung dieser entscheidenden Punkte ermöglicht es, periodische Spezifikationen grundsätzlich aber auch im Kontext einer großen Menge an Daten effizient verwalten und auswerten zu können.

Danksagung Wir möchten uns an dieser Stelle bei den anonymen Gutachtern für die sehr wertvollen Hinweise zur Verbesserung dieses Beitrages sehr herzlich bedanken.

Literatur

1. Anselma L (2004) Recursive representation of periodicity and temporal reasoning. In: Proc. TIME'04, 11th Int. Symposium on Temporal Representation and Reasoning, Tatihou, France, July 2004, pp 52–59
2. Bertino E, Bettini C, Ferrari E, Samarati P (1998) An acess control model supporting periodicity constraints and temporal reasoning. IEEE Trans Database Syst 23(3):231–285
3. Bettini C, Jajodia S, Wang S (2000) Time Granularities in Databases, Data Mining, and Temporal Reasoning. Springer-Verlag, Berlin Heidelberg New York
4. Bettini C, Mascetti S (2005) An efficient algorithm for minimizing time granularity periodical representations. In: Proc. TIME'05, 12th Int. Symposium on Temporal Representation and Reasoning, Burlington, Vermont, USA, June 2005, pp 20–25
5. Bettini C, Mascetti S, Sean Wang X (2004) Mapping calendar expressions into periodical granularities. In: Proc. 11th TIME'04, Int. Symposium on Temporal Representation and Reasoning, Tatihou Island ,France, July 2004, pp 96–102
6. Bettini C, de Sibi R (2000) Symbolic representation of user-defined time granularities. Ann Math Artif Intell 30(1–4):53–92
7. Bresolin D, Montanari A, Puppis G (2004) Time granularities and ultimately periodic automata. In: Proc. JELIA 2004, 9th European Conference on Logics in Artificial Intelligence, volume LNAI 3229, Lisbon, Portugal, September 2004, pp 513–525
8. Casati F, Ceri S, Paraboschi S, Pozzi G (1999) Database Support for Workflow Management: The WIDE Project, chapter 7. Kluwer Academic Publishers, Boston, pp 141–168
9. Chandra R, Segev A, Stonebraker M (1994) Implementing calendars and temporal rules in next generation databases. In: Proc. Int'l Conf. on Data Engineering, Houston, USA, February 1994, pp 264–273
10. Combi C, Franceschet M, Peron A (2002) A logical approach to represent and reason about calendars. In: Proc. TIME'02, 9th Intl. Symposium on Temporal Representation and Reasoning, Manchester, UK, July 2002, pp 134–140
11. Combi C, Franceschet M, Peron A (2004) Representing and reasoning about temporal granularities. J Logic Comput 4(1):51–77
12. Cukierman D, Delgrande J (1998) Expressing time intervals and repetition within a formalization of calendars. J Comput Intell 14(4):563–597
13. Cukierman D, Delgrande J (2000) A formalization of structured temporal objects and repetition. In: Proc. TIME 2000, 7th Int'l Workshop on Temporal Reasoning and Representation, Cape Breton, Canada, July 2000, pp 83–87
14. Cukierman D, Delgrande J (2004) The sql time theory: A formalization of structured temporal objects and repetition. In: Proc. 11th TIME'04, Int. Symposium on Temporal Representation and Reasoning, Tatihou, France, July 2004, pp 28–35
15. Dal Lago U, Montanari A (2001) Calendars, time granularities, and automata. In: Proc. SSTD 2001, 7th Int. Symposium on Spatial and Temporal Databases, volume LNCS 2121, Los Angeles, US, July 2001, pp 279–298
16. Dal Lago U, Montanari A, Puppis G (2003) Towards compact and tractable automaton-based representation of time granularity. In: Proc. ICTCS'03, 8th Italien Conference on Theoretical Computer Science, volume LNCS 2841, Bertinoro, Italia, October 2003, pp 72–85
17. Demri S (2004) LTL over Integer Periodicity Constraints (extended abstract). In: Proc. of 7th Int. Conf. on Foundations of Software Science and Computation Structures (FOSSACS), volume 2987 of LNCS, Barcelona, Spain, March 2004, pp 121–135
18. Egidi L, Terenziani P (2004) A lattice of classes of user-defined symbolic periodicities. In: Proc. TIME'04, 11th Int. Symposium on Temporal Representation and Reasoning, Tatihou, France, July 2004, pp 13–20
19. Egidi L, Terenziani P (2004) A mathematical framework for the semantics of symbolic languages representing periodic time. In: Proc. TIME'04, 11th Int. Symposium on Temporal Representation and Reasoning, Tatihou, France, July 2004, pp 21–27
20. Kabanza F, Stévenne J-M, Wolper P (1990) Handling infinite temporal data. In: Proc. 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Nashville, US, April 1990, pp 392–403
21. Kabanza F, Stévenne J-M, Wolper P (1995) Handling infinite temporal data. J Comput Syst Sci 51(1):3–17
22. Leban B, McDonald D, Forster D (1986) A representation for collections of temporal intervals. In Proc. 5th National Conference on Artificial Intelligence, Philadelphia, US, August 1986, pp 367–371
23. Montanari A, Peron A, Policriti A (1999) Decidable theories of omega-layered metric temporal structures. Log J IGPL 7(1): 79–102
24. Niezette M, Stevenne J (1992) An efficient symbolic representation of periodic time. In: Proc. First Int. Conference on Information and Knowledge Management, Baltimore, US, November 1992, pp 161–168
25. Ning P, Wang XS, Jajodia S (2002) An algebraic representation of calendars. Ann Math Artif Intell 36(1–2):5–38
26. Ohlbach HJ (2004) The role of labeled partitionings for modeling periodic temporal notions. In: Proc. TIME'04, 11th Int. Symposium on Temporal Representation and Reasoning, Tatihou, France, July 2004, pp 60–63
27. Terenziani P (1997) Integrating calendar dates and qualitative temporal constraints in the treatment of periodic events. IEEE Trans Knowl Data Eng 9(5):763–783
28. Terenziani P (2002) Temporal reasoning with class and instances of events. In: Proc. TIME'02, 9th Int'l Symp. on Temporal Representation and Reasoning, Manchester, UK, July 2002, pp 100–107
29. Terenziani P (2003) Symbolic user-defined periodicity in temporal relational databases. IEEE Trans Knowl Data Eng 15(2): 489–509
30. Terenziani P, Anselma L (2006) Temporal reasoning about composite and/or periodic events. J Exp Theor Artif Intell 18(1): 87–115
31. Terenziani P, Carlini C, Montani S (2002) Towards a comprehensive treatment of temporal constraints in clinical guidelines. In: Proc. TIME'02, 9th Intl. Symposium on Temporal Representation and Reasoning, Manchester, UK, July 2002, pp 20–27
32. Terenziani P, Anselma L, Montani S (2005) Periodicity-based temporal constraints. In AI*IA 2005: Advances in Artificial In-

telligence 9th Congress of the Italian Association for Artificial Intelligence, number 3673 in LNCS, Milan, Italy, September 2005, pp 62–65
33. Toman D, Chomicki J, Rogers DS (1994) Datalog with integer periodicity constraints. In: Proc. of the 1994 Int. Symposium Logic Programming, Melbourne, Australia, November 1994, pp 189–203
34. Tuzhilin A, Clifford J (1995) On periodicity in temporal databases. Inf Syst 20(8):619–639
35. Vardi MY (1988) A temporal fixpoint calculus. In: Conference Record of the Conference on Principles of Programming Languages, San Diego, California, United States, January 1988, pp 250–259
36. Wijsen J (2000) A string-based model for infinite granularities. In: Proc. of the AAAI-2000 Workshop on Spatial and Temporal Granularity, Austin, US, July 2000, pp 9–16
37. Wolper P (1983) Temporal logic can be more expressive. Inf Control 56(1/2):72–99