

# Die Assistenzfunktion kooperativer Designflows

- verdeutlicht am Beispiel von CONCORD

## Norbert Ritter

Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049, 67653 Kaiserslautern  
Tel.: (0631) 205-3264, e-mail: ritter@informatik.uni-kl.de

## Bernhard Mitschang

Institut für Informatik, Technische Universität München, Arcisstr. 21, 80290 München  
Tel.: (089) 48095-184, e-mail: mitsch@informatik.tu-muenchen.de

**Zusammenfassung.** Viele der aus dem Bereich Workflow-Management bekannten Konzepte werden auch zur Unterstützung von Entwurfsanwendungen unter dem Stichwort Designflow-Management benötigt. Der Schwerpunkt einer Systemunterstützung liegt hier jedoch auf der *Assistenzfunktion*, d. h., Entwerfer sollen gemäß einer gewählten Entwurfsmethodik durch ihren Entwurfsauftrag geführt werden, ohne daß sie in ihrer Kreativität eingeschränkt werden. Der Assistenzaspekt verlangt nach einer variablen Unterstützung der folgenden Dimensionen: von lokalen, auftragsbezogenen Datenbeständen bis hin zu gemeinsamen Datenbeständen; von vollständiger Vorplanung der Abläufe bis hin zu vollkommen dynamischen Ablaufentscheidungen; von isolierter Auftragsbearbeitung bis hin zu freier Kooperation. Dieser Artikel diskutiert, welche Konzepte geeignet sind, den gesamten durch diese drei Dimensionen aufgespannten Raum flexibel zu unterstützen. Wir werden diese Konzepte am Beispiel des CONCORD-Systems erläutern.

**Schlüsselwörter:** Designflow-/Workflow-Management, Kooperation, Entwurfsanwendungen.

**Abstract.** Many concepts developed in the research area of workflow management can also be successfully applied in order to support design applications. Designflow management has to emphasize on an assistance function, i. e., designers must be guided through design tasks without restricting their creativity. The assistance aspect demands a flexible support of the following dimensions: from task-specific data pools to shared data pools; from completely predictable to completely dynamic flow executions; from isolated task processing to free cooperation. This paper discusses concepts well suited to flexibly support the entire space spanned by these dimensions. We will use the CONCORD system as a sample designflow-management system.

**Keywords:** Designflow/workflow management, cooperation, design applications.

**CR Subject Classification:** C3, H.1.2, H.2.4, H.2.8, H.5.3, J6, J7.

## 1 Einleitung

Entwurfsanwendungen werden durch sogenannte CAD-Frameworks [25, 5, 21] unterstützt. Ein CAD-Framework ist eine Software-Infrastruktur, die eine gemeinsame Anwendungsumgebung für CAD-Werkzeuge zur Verfügung stellt. In Anlehnung an [25, 1] hat ein *CAD-Framework* die in Abb. 1 gezeigte Grobarchitektur. Verschiedenartige Werkzeuge kommunizieren mit dem Framework-Kern über ein *Tool-Framework-Interface*. Der *Framework-Kern* umfaßt alle gemeinsamen Dienste. Die *Entwurfsdatenverwaltung* beispielsweise ist von solch zentraler Bedeutung. Sie muß die Verwaltung und konsistenzerhaltende Verarbeitung sowohl der in aller Regel versionierten, primären Entwurfsdaten (Entwurfsobjekte), als auch der sekundären, entwurfsprozeß-beschreibenden (Meta-)Daten unterstützen. Die Menge der Werkzeuge kann in drei Gruppen unterteilt werden. *Integrierte* Entwurfswerkzeuge sind solche, die mittels einer sog. White-Box-Integration verfügbar sind und daher direkt über das Tool-Framework-Interface von den zentralen Diensten im Kern Gebrauch machen können. *Gekapselte* Werkzeuge hingegen müssen, da bei der Integration in das Framework ihr Source-Code nicht verfügbar ist, mit eigens zur Verfügung gestellten Konvertierungsfunktionen, sogenannten *Wrappern*, den Entwurfsdatenaustausch regeln. Die dritte Gruppe sind die *Framework-Tools*, d.h. alle Werkzeuge und Dienste, über die die Benutzer (Entwerfer oder auch Administratoren) mit dem Framework in Verbindung treten können. Dies umfaßt auch Funktionen, die die Spezifikation von Designflows sowie die entsprechende Führung der Benutzer während des Entwurfsprozesses vornehmen und damit dem Benutzer in der Einhaltung einer gewählten Entwurfsmethodik assistieren. In Abb.1 haben wir diesen Teil des *Designflow-Management*, der mit dem Benutzer in Kontakt tritt, *Designflow-Interface* genannt. Das

Designflow-Interface arbeitet natürlich mit Ablaufsteuerungskomponenten zusammen, die Teil des Framework-Kerns sind und deren Funktionalität in diesem Artikel diskutiert wird.

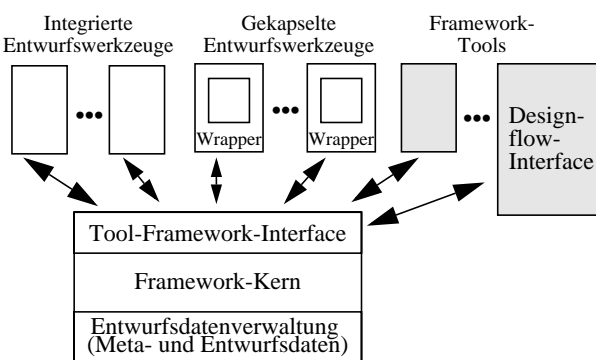


Abb. 1: Grobarchitektur eines CAD-Frameworks

Das Designflow-Management ist auf den ersten Blick eine dem Workflow-Management (Einführungen/Übersichten finden sich in [11, 10, 23, 3, 8] sowie in den Artikeln der Workflow-Management-Coalition, wie z. B. [24]) vergleichbare Aufgabe. Der angegebenen Literatur kann jedoch entnommen werden, daß für unterschiedliche Arten von Workflows (in der Regel werden administrative, Produktions-, kollaborative und Ad-hoc-Workflows unterschieden) spezifische Systemlösungen existieren. Wie wir im folgenden jedoch sehen werden, verlangen Entwurfsanwendungen nach der Integration von Systemlösungen für die Unterstützung von sowohl geregelten (vorplanbaren) als auch unregulierten (nicht im einzelnen vorhersehbaren) Abläufen, wobei letztere insbesondere auch die Unterstützung einer weitgehend freien Kooperation unter Konsistenzhaltung der Entwurfsdaten erfordern. Betrachten wir das volle Kontinuum der 'Geregeltheit', so wird intuitiv deutlich, was unter der Assistenzfunktion des Designflow-Management zu verstehen ist. Im Falle 'voller Geregeltheit' (voll vorgeplanter Ablauf) braucht dem Entwerfer im Treffen von Ablaufentscheidungen *nicht assistiert* zu werden, da alle Ablaufentscheidungen durch die Workflow-Spezifikation vorweggenommen wurden. Im Falle des anderen Extremes der 'vollen Ungeregeltheit' (keine Vorplanung möglich), kann *ebenfalls keine Assistenzfunktion* ausgeübt werden, da dem System keine Grundlagen für die Vorbereitung von Ablaufentscheidungen vorab bekanntgemacht werden können. Liegt jedoch eine *teilweise geregelte / teilweise unregelte Anwendung* vor, so sind dynamische Ablaufentscheidungen durch den Benutzer zu treffen, wobei das *System assistieren kann*, in dem es garantiert, daß die dynamischen Entscheidungen mit vorab gegebenen Spezifikationen bzgl. Korrektheit der Abläufe und Konsistenz der Entwurfsdaten konform sind, bzw. dem Benutzer im Treffen einer Entscheidung durch Anbieten von Lösungsmöglichkeiten hilft. Insgesamt müssen bei dem Versuch der Nutzung der aus den verschiedenen Bereichen des Workflow-Management bekannten Mechanismen folgende Eigenheiten von Entwurfsprozessen berücksichtigt werden:

- die unverzichtbare Kreativität der Entwerfer verlangt oft nach dynamischen Ablaufentscheidungen, die z. B. von der Güte des bisherigen Teilentwurfs abhängig sein können;
- Entwurfsabläufe werden oft zunächst nur grob spezifiziert und erst nachträglich (zur Laufzeit) bis hin zur Ebene des Werkzeugaufrufs verfeinert;
- die aufgrund der hohen Komplexität notwendige Zerlegung der Entwurfsaufgabe und entsprechende Delegation von Teilaufträgen an Entwerfer geschieht in der Regel dynamisch; die Rollenzuordnung ist nicht so statisch wie in klassischen Anwendungsgebieten des Workflow-Management, wie etwa der Geschäftsvorgangsbearbeitung; insbesondere sind dynamische Änderungen der Aufträge, des Ablaufs und der Organisation zu berücksichtigen;
- die zu unterstützende Kooperation zwischen Entwerfern erfordert eine im Vergleich zum Workflow-Management 'noch' reichhaltigere Fehlerbehandlung, da auch im Falle von Kooperationskonflikten eine nach 'vorne' gerichtete Konfliktauflösung angestrebt werden muß.

Dennoch können viele der im Bereich des Workflow-Management eingesetzten Mechanismen auch im Designflow-Management sinnvoll genutzt werden, wie in den folgenden Kapiteln deutlich werden wird. Insbesondere zählen hierzu grundlegende Koordinationsdienste [14], aber auch Techniken der Datenbankanbindung und Transaktionsunterstützung [7]. Letztere haben im Rahmen des Designflow-Management eine besonders hohe Bedeutung, da die Entwurfsdaten in einer logisch zentralen Entwurfsdatenhaltung (siehe entsprechende Framework-Komponente in Abb. 1) verwaltet werden. Diese Entwurfsdatenverwaltung muß eine kooperative Arbeitsweise der involvierten Entwerfer zulassen und einen anwendungsspezifischen Konsistenzbegriff (z. B. durch Entwurfsqualität bedingte Freigabestufen von Teilentwürfen) unterstützen. All diese unterstützenden Maßnahmen rechnen wir der Assistenzfunktion des Designflow-Management-Systems zu, so daß insbesondere die drei folgenden Aspekte von herausragender Wichtigkeit sind:

- *Datenbereiche (von lokalen, auftragsbezogenen Daten bis hin zu gemeinsamen Daten)*  
In frühen Entwurfsphasen werden häufig sehr viele Daten erzeugt, die als eher entwurfsauftrags-lokal zu betrachten sind und für die Bearbeitung anderer Entwurfsaufträge kaum Bedeutung haben. In späteren Phasen jedoch kann es sehr sinnvoll sein, bedeutungsvolle Entwurfsobjektzustände gezielt (vorab-) freizugeben bzw. auf gemeinsamen Datenbeständen zu arbeiten. Vom Designflow-Management-System ist die effiziente Unterstützung der damit verbundenen Zugriffsmöglichkeiten zu fordern.
- *Grad der Vorplanung (von vollständiger Vorplanung bis hin zu dynamischen Entscheidungen)*

Wir haben oben bereits den Aspekt der ‘Geregeltheit’ diskutiert. Daneben sind verschiedene Entwurfsmethodiken oft auf verschiedenen Abstraktionsebenen angesiedelt, d. h., sie beschreiben die zugehörige Vorgehensweise des Entwurfs mehr oder weniger detailliert. Daher muß das Designflow-Modell sowohl abstrakte als auch konkrete Spezifikationen erfassen können.

- *Kooperation (von isolierter Auftragsbearbeitung bis hin zu freier Kooperation)*

Durch eine vollständige Spezifikation der Ablaufschemata und Ziele von Entwurfsaufträgen kann in seltenen (außerordentlich überschaubaren) Fällen die Notwendigkeit der Aufeinanderabstimmung während der Bearbeitung der Entwurfsaufträge vermieden werden, so daß keine Kooperation während der Bearbeitung auftritt. In der Regel jedoch werden die (Ziele und Spezifikationen von) Entwurfsaufträge(n) während der Bearbeitung verfeinert, so daß eine Abstimmung zwischen den Entwurfsaufträgen notwendig wird. Entsprechend flexible Kooperationsmechanismen müssen bereitgestellt werden.

Die Idee, Workflow-Management-Fähigkeiten in ein Assistenzsystem zu integrieren, ist auch schon in anderen Anwendungsbereichen angedacht worden. In [2] bspw. werden entsprechende Konzepte zur Unterstützung klinischer Anwendungen vorgestellt. In diesem Bereich stehen jedoch mehr die Aspekte ‘Überwachung organisatorischer Abläufe’, ‘Terminvergabe’ und ‘Warnhinweise’ und nicht die drei vorgenannten Aspekte im Vordergrund, die wir in diesem Artikel betrachten wollen.

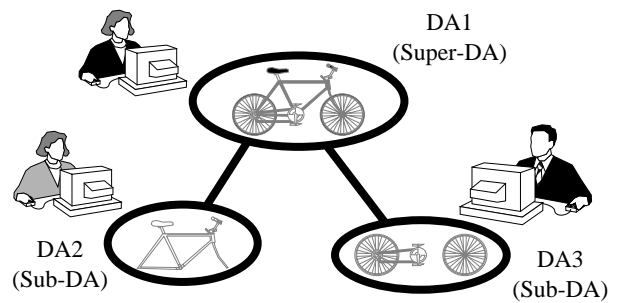
In die nachfolgenden, allgemeinen Betrachtungen zu den Grundlagen einer Assistenzfunktion fließen die Ergebnisse und Erfahrungen im Zusammenhang mit dem Prototypsystem CONCORD (CONtrolling COopeRation in Design environments [18, 19, 20]) ein. Das CONCORD-System realisiert ein datenbankorientiertes Entwurfsablaufmodell, das neben einer transaktionsgeschützten Verarbeitung von Entwurfsdaten [6] auch Mechanismen für Designflow-Management und Kooperationskontrolle [15] anbietet. Die beiden letzteren Aspekte sollen in diesem Artikel angesprochen und vorgestellt werden. Die Konzepte von CONCORD wurden darauf ausgerichtet, die verschiedenen Ausprägungen der Kriterien *Datenbereiche (getrennte vs. gemeinsame Datenbestände)*, *Grad der Vorplanung (Vorplanung vs. dynamische Ablaufentscheidungen)*, *Kooperation (isolierte vs. kooperierende Auftragsbearbeitung)* sowie mögliche Kombinationen daraus zu erfassen. Die flexible Unterstützung dieser Dimensionen ist von zentraler Bedeutung für die Assistenzfunktion und bildet somit den Fokus der weiteren Betrachtungen. Kap. 2 beschäftigt sich mit der Dekomposition von Entwurfsaufgaben und mit der Zuordnung von Datenbereichen zu Entwurfsaufträgen. Kap. 3 beschreibt ein-

fache, d. h. zunächst nicht kooperative Designflows und zeigt, wie dynamische (Ad-hoc-)Designflows integriert werden können. Kooperationsaspekte werden anschließend in Kap. 4 behandelt. Kap. 5 stellt die Bedeutung des Zusammenwirkens von Designflow-Management und Kooperationskontrolle hinsichtlich der Bereitstellung der geforderten Assistenzfunktion heraus und, Kap. 6 schließt die Betrachtungen mit einer Zusammenfassung ab.

## 2 Designflows zur Abwicklung von Entwurfsaufgaben

Während in der klassischen Geschäftsvorfallbearbeitung die Zerlegung der Aufgabe im wesentlichen statisch durch organisatorische Strukturen (betriebliche Funktionsträger, für deren Interaktion vorab starre Muster existieren) vorgegeben ist, muß eine Entwurfsaufgabe aufgrund der meist sehr hohen Komplexität unter ausschließlicher Betrachtung des eigentlichen Problems zerlegt werden, wobei die Problemzerlegung auch dynamisch im Laufe des Entwurfsprozesses fortgesetzt werden kann. Assoziiert man mit den durch die Zerlegung entstehenden Teilaufgaben ausführende Entwerfer, so wird deutlich, daß flexible Kooperationsprimitive benötigt werden.

(a) Ausgangssituation: delegierte Teilaufträge



(b) Zielsituation: fertige Teilaufträge, gelöste Entwurfsaufgabe

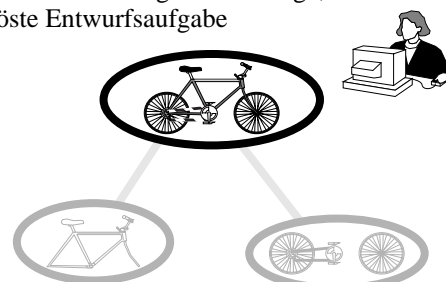


Abb. 2: Delegation in CONCORD

In CONCORD werden Entwurfsaufträge system-seitig explizit durch entsprechende Ablaufeinheiten, Entwurfsaktivitäten (engl. Design Activity oder kurz DA) genannt, repräsentiert. Während des Entwurfsprozesses kann dynamisch eine Hierarchie von DAs erzeugt und verändert werden. Der Entwurfsprozeß startet demnach mit der Erzeugung einer (Top-Level-)DA, die den übergeordneten Entwurfsauftrag repräsentiert und

während der Verarbeitungsphase Teilaufträge durch Erzeugung von Sub-DAs auslagern kann. Den einzelnen DAs sind Entwerfer(-Teams) zugeordnet, die während der Entwurfs-Sitzungen durch Werkzeuganwendungen auf die Erfüllung des entsprechenden Auftrages hinarbeiten, wobei ihnen das Designflow-Management-System in der Einhaltung der gewählten Entwurfsmethodik assistiert (was dies im einzelnen bedeutet, wird in den nachfolgenden Kapiteln erläutert). Abb. 2 (a) zeigt in einem einfachen, intuitiven Beispiel das Prinzip der Delegation von Teilaufträgen. Das Entwurfsproblem wird entlang der Entwurfsobjektstruktur zerlegt: der Entwurf des Fahrrades (DA1) wird zunächst aufgespalten in den Entwurf des Rahmens (DA2) und den des Antriebs (DA3). Diese beiden Teilaufgaben können parallel von (kooperierenden) Entwerfern ausgeführt werden.

Für die Erzeugung einer DA sind folgende Angaben erforderlich:

- **Designflow-Spezifikation**  
Die Designflow-Spezifikation enthält eine Beschreibung der zu verfolgenden Vorgehensweise. Damit ist der von uns verwendete Begriff 'Designflow-Spezifikation' mit den in der Workflow-Literatur oft benutzten Begriffen wie 'Workflow-Typ' oder 'Workflow-Schema' vergleichbar. Die Einhaltung dieser Vorgaben wird während der Laufzeit der DA durch das Designflow-Management-System erzwungen. In diesem Sinne beschreibt der Begriff 'Designflow' (als Ausprägung der der DA zugeordneten Designflow-Spezifikation) den Laufzeitaspekt der DA. Kap. 3 und 4 werden in die Möglichkeiten der Erstellung von Designflow-Spezifikationen einführen.
- **Entwurfsziel-Spezifikation**  
Das Entwurfsobjekt, das als Ergebnis der Auftragsbearbeitung erwartet wird, soll in aller Regel eine bestimmte Menge von Eigenschaften erfüllen. Diese Eigenschaften können entweder als durch das DBS überprüfbar (evtl. komplexe, d. h. komplexe Objektstrukturen betreffende) Prädikate oder durch explizite Prüfroutinen, bspw. Finite-Element-Analysen zur Gewährleistung von Belastungseigenschaften, repräsentiert werden.
- **Entwurfsobjekttyp**  
Einem Entwurfsauftrag kann ein Entwurfsobjekttyp zugeordnet werden, der eine schematische Beschreibung des (als Ergebnis erwarteten) Entwurfsobjekts darstellt. Der Entwurfsobjekttyp ist nichts anderes als ein Schemaelement der Entwurfsdatenbank (bspw. ein Konfigurations- oder Komplex-Objekt-Typ). So wird z. B. jede VLSI-Entwurfsdatenbank ein Schemaelement 'Chip' oder 'Zelle' beinhalten und eine Produktdatenbank bspw. für Fahrräder eine Stücklistenstruktur.

- **Initialer Datenbereich**  
Der Datenbereich einer DA umfaßt die Menge der Entwurfsdaten<sup>1</sup>, die für die innerhalb dieser DA auszuführenden Entwurfswerkzeuge zugreifbar sind. Es kann nun einerseits aus anwendungsspezifischen Gesichtspunkten sinnvoll sein, einer DA einen lokalen Datenbereich zuzuweisen; andererseits gibt es jedoch auch Anwendungsfälle, in denen es geeigneter ist, mehrere DAs auf gemeinsamen Entwurfsdaten arbeiten zu lassen. Wir werden insbesondere in der Diskussion des Kooperationsaspektes (siehe Kap. 4) näher auf gemeinsame Entwurfsdatenbereiche eingehen. Ob eine neu erzeugte DA auf einem mit anderen DAs gemeinsamen Datenbereich arbeitet oder ob ihr zunächst ein lokaler Datenbereich zuzuordnen ist<sup>2</sup>, wird mit der Spezifikation des initialen Datenbereiches festgelegt. Zu diesem Zweck stellt CONCORD entsprechende (deskriptive) Datenbankankweisungen zur Verfügung. Ein lokaler Datenbereich ist dann sinnvoll, wenn dem zugehörigen Entwurfsauftrag bestimmte Daten eindeutig zugeordnet werden können und es zu erwarten ist, daß während der Bearbeitung des neuen Entwurfsauftrages viele, nur lokal relevante Daten erzeugt werden. Gemeinsame Datenbereiche sind sinnvoll, wenn die (meisten) Zwischenergebnisse der DAs auch für die jeweils anderen bedeutungsvoll sind.
- **Verantwortliche(r) Entwerfer**  
Der verantwortliche Entwerfer (bzw. das verantwortliche Team) muß alle dynamisch zu treffenden Entwurfsentscheidungen liefern und interaktive Werkzeuge (im Sinne des Entwurfsauftrages) bedienen. Er wird demnach von dem Designflow-Management-System zur Laufzeit aufgefordert, alle Informationen bereitzustellen, die in der vordefinierten Designflow-Spezifikation nicht enthalten sind.

Insgesamt läuft die Entwurfsarbeit innerhalb einer DA folgendermaßen ab: Der verantwortliche Entwerfer arbeitet mit Unterstützung des Designflow-Management-Systems, das sich wiederum an der Designflow-Spezifikation orientiert, auf das Entwurfsziel hin. Bei Erreichen des Entwurfszieles (dies erfordert unter anderem, daß die dem Datenbanksystem in Form von Prädikaten bekanntgemachten Teile der Entwurfsziel-Spezifikation als gültig evaluiert wurden), kann die DA beendet und damit die Delegationsbeziehung entfernt werden.

---

1. Im allgemeinen unterscheiden wir allgemeine Entwurfsdaten von der Gesamtmenge der den aktiven DAs zugeordneten Entwurfsdaten. Allgemeine Entwurfsdaten beinhalten allgemeine Regeln, Standards etc., sind für alle DAs zugreifbar und werden vorwiegend lesend zugegriffen. Die Diskussionen dieses Artikels beziehen sich jedoch ausschließlich auf die den DAs zugeordneten (auftragsbezogenen) Entwurfsdaten.

2. Die Zuweisung eines lokalen Datenbereiches bedeutet, daß nur auf explizit vorgeplanten Kooperationswegen Entwurfsdaten ausgetauscht werden können (siehe Kap. 4).

In Abb. 2 (b) ist die erfolgreiche Beendigung der beiden Sub-DAs und das Ergebnis der Entwurfsaufgabe unseres Beispiels angedeutet. In nachfolgenden Bearbeitungsphasen könnten von DA1 aus dynamisch weitere Teilaufträge durch Erzeugen von Sub-DAs ausgelagert werden. Mit der Terminierung einer Sub-DA gehen die lokalen Daten (falls der Sub-DA ein lokaler Datenbereich zugeordnet war) in den Datenbereich der Vater-DA über. Dabei kann eine unterschiedliche Behandlung von Ergebnisdaten (vgl. Punkt 'Entwurfsobjekttyp' oben) und Hilfsdaten vorgenommen werden, die jedoch von dem Einzelfall abhängig ist.

### 3 Spezifikation und Kontrolle 'einfacher' Designflows

Wie im vorangegangenen Kapitel verdeutlicht, wird einer DA bei ihrer Erzeugung ein initialer Entwurfsdatenstatus zugeordnet. Durch Aktivitäten (Entwurfswerkzeuganwendungen) werden Entwurfsdatenstatus fortgeschrieben, bis letzten Endes ein solcher Zustand vorliegt, der das Entwurfsziel erfüllt. Diese Aktivitäten können mittels einer Designflow-Spezifikation bis zu einem gewünschten Grad vorbestimmt werden. Wir wollen hier klären, auf welchen Elementen die Spezifikation beruht, und kurz ansprechen, wie diese vom Designflow-Management-System abgearbeitet werden kann.

#### 3.1 Spezifikation 'einfacher' Designflows

Ähnlich wie bei vielen Workflow-Systemen [11] besteht eine Designflow-Spezifikation aus Zustands- und Transitionsspezifikationen. Zustände werden implizit spezifiziert. Eine Transition besteht neben dem Transitionsnamen aus einer (durch das darunterliegende Datenbanksystem) auf dem jeweiligen Entwurfsdatenstatus evaluierbaren Vor- und einer entsprechenden Nachbedingung sowie aus einem Aktionsteil. Letzterer kann Aktivitätsspezifikationen der folgenden Abstraktionsstufen beinhalten. Eine detaillierte Beschreibung dieser Spezifikationsmöglichkeiten findet sich in [19].

- **Werkzeugaufruf**  
Ein konkreter Werkzeugaufruf wird durch Angabe des Werkzeugnamens und durch die Bereitstellung von Eingabeparameterbelegungen vorgeplant.
- **Abstrakte Aktivität**  
Eine abstrakte Aktivität räumt dem Entwerfer zusätzliche Freiheiten ein; zur Laufzeit wird es ihm überlassen, welche Aktivitäten zur Erreichung der Nachbedingung der entsprechenden Transition durchzuführen sind. Sinnvolle Einschränkungen können in Form von Regeln hinzugefügt werden; bspw. könnte es notwendig sein zu spezifizieren, daß, falls Werkzeug1 aufgerufen wird, unmittelbar danach Werkzeug2 aufgerufen werden muß. Nach Aktivierung einer DA verbleiben dem zugeordneten Entwerfer folgende Möglichkeiten. Einerseits kann

er die abstrakte Aktivität durch einen die angegebenen Regeln erfüllenden Sub-Designflow ersetzen (Verfeinerung). Die andere Möglichkeit besteht darin, innerhalb der abstrakten Phase jeweils dynamisch zu entscheiden, welches Werkzeug als nächstes aufgerufen werden soll. Letzteres bedeutet allerdings, daß vor Ausführung eines Werkzeugs jeweils vom Designflow-Management-System sichergestellt werden muß, daß die Ausführung nicht gegen eine der spezifizierten Regeln verstoßen wird.

- **Aufruf eines Sub-Designflows**  
Der Aktionsteil einer Transitions-Spezifikation kann auch einen weiteren Designflow-Namen enthalten, wodurch eine Schachtelung von Designflows erreicht wird. Dies unterstützt eine modulare Erstellung und sinnvolle Wiederverwendung von Designflow-Spezifikationen. Ein Beispiel dazu ist der Sub-Designflow 'Abstimmung', der in Abb. 3 gezeigt und in Kap. 4.1 besprochen wird.
- **Komplexe Aktivität**  
Neben den üblichen Kontrollflußmöglichkeiten (Sequenz, Parallelausführung, Alternativverzweigung) auf der Designflow-Ebene können komplexe Transitionen dazu genutzt werden, um bspw. Iterationen zu realisieren. So besteht z. B. die Möglichkeit, einen Sub-Designflow so lange zu iterieren, bis die Nachbedingung der Transition erfüllt ist. Auf eine vollständige Beschreibung der Möglichkeiten zur Definition von komplexen Transitionen wird an dieser Stelle aus Platzgründen verzichtet.

Für die notwendigen Datenflußspezifikationen wird ein ähnlicher Mechanismus wie die im FlowMark-System [13] benutzten Input-/Output-Container verwendet. Zusätzlich zu den Eingabe-/Ausgabeparametern von Werkzeugen können für Designflows sowie für abstrakte Aktivitäten Parameter deklariert werden. Datenflußspezifikationen können auch weggelassen werden; dies führt zur Laufzeit dazu, daß der Entwerfer durch das Designflow-Management-System veranlaßt wird, dynamisch die entsprechenden Bindungen anzugeben.

Durch die geeignete Kombination der genannten Spezifikationsmöglichkeiten können sehr mächtige Ablaufbeschreibungen erzeugt werden, die dem ausführenden Entwerfer einerseits beliebige Freiheitsgrade einräumen können (Unterstützung sog. unregelter Entwurfsaktivitäten), es auf der anderen Seite aber auch erlauben, sehr strikte Vorgaben auszudrücken (geregelter Aktivitäten). Die Integration dieser beiden Aspekte unterscheidet den hier beschriebenen Ansatz von herkömmlichen Workflow-Modellen.

#### 3.2 Ablaufkontrolle und Ausnahmebehandlung

Wie bereits angesprochen, wird die Einhaltung der einer DA zugeordneten Designflow-Spezifikation zur Laufzeit durch das Designflow-Management-System erzwungen. Nur bei abstrakten Transitionen (es handelt

sich hierbei um Transitionen, deren Aktivitätsspezifikation eine abstrakte Aktivität beschreibt; vgl. Kap. 3.1), bei fehlenden Datenflussspezifikationen oder durch interaktive Werkzeuge wird Entwerferinteraktion erforderlich und daher auch vom System veranlaßt.

Neben den angesprochenen Freiheitsgraden, die bereits in die Designflow-Spezifikation eingebaut werden können, werden flexible Konzepte angeboten, die es dem Benutzer ermöglichen, bei Strategiekonflikten (Werkzeugabbrüche, nicht erfüllte Vorbedingungen von Transitionen, etc.), in der Designflow-Historie zurückzupositionieren und einen alternativen Weg oder den gleichen Weg mit anderer Parameterbelegung bzw. anderer Werkzeugbedienung erneut zu durchlaufen mit dem Ziel, die Entwurfsobjektqualität zu steigern. Zu diesem Zweck führt das Designflow-Management-System ein *Designflow-Protokoll*. Dieses besteht im wesentlichen aus Einträgen, die die Aufrufstruktur der bisher durchgeführten Werkzeugläufe und Informationen über die für den zurückliegenden Datenfluß relevanten Objekte, Versionen und Konfigurationen enthält. Zeigt der Entwerfer dem Designflow-Management-System an, daß er zu einem bestimmten Zustand zurückspringen möchte, so werden im Gegensatz zu vielen anderen Systemen keine Kompensationen der betroffenen Transaktionen bzw. Aktivitäten eingeleitet, da eine Bereitstellung von Kompensationstransaktionen für die in interaktiven Entwurfswerkzeugen vorgenommenen Datenmanipulationen kaum möglich ist. In vielen Fällen erlaubt die Nutzung von Versionierungs- und Konfigurationsdiensten ein 'reibungsloses' Wiederaufsetzen auf einem früheren Zustand. Eine detaillierte Beschreibung der beim Zurückpositionieren (systemseitig) durchzuführenden Maßnahmen findet sich in [19].

## 4 Spezifikation und Kontrolle kooperativer Designflows

Nachdem wir in den beiden vorangegangenen Kapiteln die Aspekte *Datenbereiche* und *Grad der Vorplanung* diskutiert haben, möchten wir hier auf den *Kooperationsaspekt* eingehen. Dabei soll deutlich gemacht werden, daß die drei im Vordergrund stehenden Aspekte nicht nur in ihren Extremformen zu betrachten sind, sondern daß es das volle Kontinuum und die vielfältigen Kombinationsmöglichkeiten auszuschöpfen gilt, um eine Anwendung adäquat zu unterstützen.

Selbstverständlich sind die Möglichkeiten der Kooperationskontrolle durch die bislang diskutierten Aspekte teilweise vorgegeben, so daß es sich nicht um vollständig orthogonale Aspekte handelt. So ergibt sich die Notwendigkeit der Kooperation aus der gewählten Dekomposition der Entwurfsaufgabe, und die Form der Kooperationskontrolle muß sowohl zu dem Grad der Vorplanung der kooperierenden DAs als auch zur Organisation der zugehörigen Datenbereiche passen. Bei-

spielsweise muß eine eher unregelmäßige Anwendung mit hohem Kooperationsbedarf durch geeignete Primitive für eine freie Kooperation unterstützt werden.

### 4.1 Spezifikation kooperativer Designflows

Die beiden in den vorangegangenen Kapiteln diskutierten Aspekte (unterschiedliche Arten von Datenbereichen und unterschiedliche Grade der Vorplanung) verlangen auch hinsichtlich der Kooperationskontrolle angemessen flexible Konzepte. Im allgemeinen unterscheiden wir zwei Arten der Kooperation: *vorgeplante* und *dynamische* Kooperation. Im Falle einer geregelten Anwendung können sowohl die innerhalb der benötigten DAs stattfindenden Abläufe als auch die kooperativen Abstimmungsaktionen zwischen DAs mit Kooperationsbedarf *vorgeplant* werden. Im Falle einer unregelmäßigen Anwendung kann bei der Aufgabenzerlegung lediglich festgestellt werden, welche DAs aufgrund ihrer jeweiligen Entwurfsziele einen Kooperationsbedarf aufweisen. Die Abwicklung der (zur Laufzeit stattfindenden) kooperativen Abstimmungsaktionen kann jedoch nicht vorgeplant werden, so daß eine freie, *dynamische* Kooperation unterstützt werden muß.

Welche Mechanismen zur Kooperationskontrolle herangezogen werden müssen, hängt weiter davon ab, ob die kooperierenden DAs auf einem gemeinsamen Datenbereich oder auf getrennten Datenbereichen arbeiten (vgl. Diskussion des Aspektes *Datenbereiche* in Kap. 2). Betrachten wir zunächst den Fall der getrennten Datenbereiche. Mit Hilfe der folgenden Kooperationsoperationen können zielgerichtet bedeutungsvolle Entwurfsobjektzustände (vorab und selektiv) freigegeben bzw. zugreifbar gemacht werden.

- *Permit*: Mit der Ausführung der Permit-Operation erlaubt die ausführende DA der kooperierenden DA, auf einen spezifizierten Entwurfsobjektzustand (in der Regel eine Version oder Konfiguration) zuzugreifen. Es kann Leserecht, Schreibrecht und im Hinblick auf Versionen Ableitungsrecht eingeräumt werden.
- *Revoke*: Mit Revoke können eingeräumte Rechte wieder entzogen werden.
- *Transmit*: Transmit überträgt die angegebenen Daten vollständig (Besitzrecht) von dem lokalen Datenbereich der ausführenden DA in den lokalen Datenbereich der angegebenen DA.

Je nach Kooperationsbedarf und Vorplanungsgrad der lokalen Designflows können DAs einerseits die Erlaubnis erhalten, die angesprochenen Kooperationsoperationen dynamisch auszuführen (dynamische Kooperation bei getrennten Datenbereichen), oder die Ausführung von Kooperationsoperationen kann in den entsprechenden Designflows vorgeplant werden (vorgeplante Kooperation bei getrennten Datenbereichen). Bei dynamischer Nutzung kann zusätzlich eine *Request*-Operation verwendet werden, mit der der Zugriff

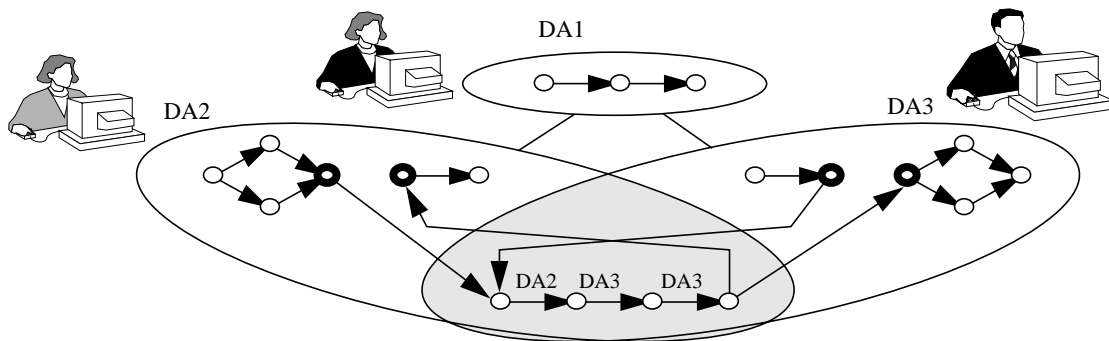


Abb. 3: Kooperativer Sub-Designflow

auf einen Entwurfsobjektzustand (mit einer bestimmten Entwurfsqualität) der kooperierenden DA 'beantragt' werden kann. Die Request-Operation dient einerseits der systemgestützten Kommunikation zwischen den Entwerfern, andererseits impliziert sie eine Notifikationsfunktion. Letzteres bedeutet, daß ein Entwerfer vom Designflow-Management-System benachrichtigt wird, wenn er einen Entwurfszustand erreicht und der zugehörige Entwurfsobjektzustand von einer kooperierenden DA mittels der Request-Operation vorher angefragt worden war.

Auf die angesprochene Möglichkeit der Vorplanung von kooperativen Aktionen soll noch etwas näher eingegangen werden. Die einfachste Form der vorgeplanten Kooperation besteht nun in der Spezifikation einer der oben angegebenen Kooperationsoperationen im Aktionsteil einer Transitionspezifikation, d. h., wir erweitern die in Kap. 3.1 gegebene Liste von möglichen Aktivitätsspezifikationen um den Punkt 'Aufruf einer Kooperationsoperation'. Der Nachteil dieser Art der vorgeplanten Kooperation ist jedoch oft, daß keine gemeinsamen Kooperationsphasen spezifiziert werden können, die der 'bilateralen' Abstimmung dienen. Daher führen wir sogenannte *kooperative Sub-Designflows* ein. Zur Erläuterung dient die Illustration in Abb. 3. Sie zeigt eine Top-Level-DA mit zwei Sub-DAs, für die ein Kooperationsbedarf angenommen wird. Die die Kooperationsphase betreffende Vorplanung äußert sich hier in einem gemeinsamen Sub-Designflow (in Abb. 3 grau hinterlegt). 'Gemeinsam' bedeutet, daß die beiden DAs beim Eintritt in die gemeinsame Phase synchronisiert werden und dann denselben Designflow gemeinsam durchlaufen. Jede der Einzelaktivitäten des gemeinsamen Sub-Designflows wird im Verantwortungsbereich einer der beteiligten DAs durchgeführt, wie in Abb. 3 angedeutet. So könnte beispielsweise folgende Vorgehensweise vorgeplant werden: ein innerhalb von DA2 erarbeiteter Vorentwurf wird im ersten Schritt für DA3 zugreifbar gemacht; im zweiten Schritt wird eine Verifikation des Vorentwurfes durch DA3 vorgenommen; im dritten Schritt wird das Verifikationsergebnis zurückgegeben.

Auch im Falle eines gemeinsamen Datenbereiches der kooperierenden DAs kann dynamische und vorgeplante Kooperation unterstützt werden. Hier liegt der

Schwerpunkt jedoch nicht in der kontrollierten 'Vorab-Freigabe' von Entwurfsinformation, sondern in der Organisation der notwendigen Schritte zum Aufeinanderabstimmen der durch die kooperierenden DAs zu erstellenden Entwürfe. Auch im Falle eines gemeinsamen Datenbereiches kann ein kooperativer Sub-Designflow spezifiziert werden, der der Koordination von Arbeitsschritten dient, die allen beteiligten DAs sowie dem übergeordneten Entwurfsziel zuträglich sind (vorgeplante Kooperation bei gemeinsamen Datenbereichen). Hier macht es natürlich keinen Sinn, den Aufruf von Kooperationsoperation vorzuplanen, da die beteiligten DAs von vornherein auf demselben Datenbereich arbeiten. Die Nutzung eines kooperativen Sub-Designflows ist bei eher geregelten Anwendungen sinnvoll. Im Falle der eher unregelmäßigen Anwendung kann die sogenannte *abstrakte Kooperationsphase* vorgesehen werden (dynamische Kooperation bei gemeinsamen Datenbereichen). Auch hier wird eine gemeinsame Phase vereinbart, jedoch beinhaltet der kooperative Sub-Designflow lediglich eine abstrakte Transition, in der die kooperierenden DAs gemeinsam die entsprechenden Ziele der Kooperationsphase erreichen müssen. Innerhalb dieser Phase darf in der Regel nicht uneingeschränkt gearbeitet werden, es müssen vielmehr applikationsspezifische Korrektheitskriterien eingehalten werden. Applikationsspezifische Korrektheitskriterien sind im wesentlichen dazu notwendig, um bestimmte, erwünschte Kooperationsmuster, d. h. Folgen von Werkzeugaufrufen, zu erzwingen. Für diesen Zweck ist der in [17] gewählte, auf endlichen, deterministischen Automaten basierende Ansatz geeignet. Da dieser Ansatz in [17] jedoch zur Steuerung von Folgen von Datenbankoperationen, d. h. zur Synchronisation auf Datenbankebene, benutzt wird, mußte der Mechanismus an die Gegebenheiten des Designflow-Management angepaßt werden. Beispielsweise könnte durch einen Automaten ausgedrückt werden, daß DA3 immer dann einen Verifikationsschritt ausführen muß, nachdem DA2 ein bestimmtes, das Entwurfsobjekt in einer bestimmten Weise änderndes Werkzeug aufgerufen hat. Man beachte hierbei, daß es sich bei den durch die Automaten repräsentierten Kooperationsmustern um deskriptive, nicht um präskriptive Vorgaben handelt. Dies bedeutet, daß nicht wie im Falle des in Abb. 3 illustrierten kooperativen Sub-Designflows ein vorge-

planter Ablaufplan zur Durchführung der Abstimmungsaktionen abgearbeitet wird, sondern daß bei jedem von einem der beteiligten Entwerfer beabsichtigten Werkzeugaufruf (dynamische Ablaufentscheidung) anhand der aktiven Automaten geprüft wird, ob dieser Werkzeugaufruf mit den definierten Kooperationsmustern verträglich ist. Die genaue Adaption des Automatenansatzes zur Kooperationskontrolle in CONCORD kann in [19] nachgelesen werden.

## 4.2 Ablaufkontrolle und Ausnahmebehandlung

Die Kontrolle der Kooperationsphasen hängt wesentlich von der gewählten Form der Kooperation ab.

- **Vorgeplante Kooperation**

Handelt es sich um einen gemeinsamen Sub-Designflow, so muß der gemeinsame Eintritt der beteiligten DAs in die Kooperationsphase überwacht werden. Außerdem wird sichergestellt, daß die Kooperationsphase nur verlassen wird, wenn die lokalen Nachbedingungen der zugehörigen Transitionen erfüllt sind.

- **Dynamische Kooperation**

Bei der dynamischen Nutzung der Kooperationsoperationen (getrennte Datenbereiche) muß geprüft werden, ob den beteiligten DAs eine dynamische Kooperation in den gegenwärtigen Phasen des Entwurfes erlaubt ist. Weiterhin wird zugesichert, daß durch die Ausführung der Kooperationsoperationen und die damit verbundene Änderung der Datenbereiche der beteiligten DAs keine auf den lokalen Datenbereichen definierten Integritätsbedingungen verletzt werden.

Eine besondere Bedeutung kommt der Wartung von Requests (siehe Beschreibung der Request-Operation in Kap. 4.1) durch entsprechende Notifikationen zu. Wird z. B. ein Entwurfsobjektzustand angefragt, der momentan noch nicht existiert, so wird der Entwerfer der angesprochenen DA benachrichtigt, sobald er den angefragten Entwurfsobjektzustand hergestellt hat; er kann dann entscheiden, ob er der kooperierenden DA die angefragten Rechte erteilt oder nicht. Der Entwerfer der anfragenden DA wird natürlich auch benachrichtigt, sobald angefragte Daten zugreifbar gemacht wurden. Eine Weiterführung dieses Prinzips erlaubt es, alle weiteren Verbesserungen eines zur Verfügung gestellten Entwurfsobjektzustandes ebenfalls automatisch zugreifbar zu machen und die nutznießende DA entsprechend jedesmal zu informieren.

Bei dem Entzug von Rechten wird ebenfalls benachrichtigt. Gleiches geschieht bei der 'logischen' Invalidierung von vorab freigegebenen Entwurfsobjektzuständen (etwa durch Zurückschreiten in der Designflow-Historie). Es bleibt hier dem betroffenen kooperierenden Entwerfer überlassen, wie auf einen Rechteentzug bzw. eine Invalidierung reagiert wird.

In abstrakten Kooperationsphasen (gemeinsame Datenbereiche) muß die Einhaltung der durch die Automaten spezifizierten Korrektheitskriterien überwacht werden. Folglich läuft die Verarbeitung nach folgendem Prinzip ab: möchte ein für eine der kooperierenden DAs zuständiger Entwerfer ein Werkzeug aufrufen, so muß zunächst die Zulässigkeit der Werkzeugausführung durch die Automaten geprüft werden (vgl. Kap. 4.1).

Hinsichtlich der Kooperationskontrolle wird der Benutzer insbesondere in die Konfliktauflösung miteinbezogen [20], solange es sich um mit dem mentalen Benutzermodell verträgliche Strategie- bzw. Kooperationskonflikte handelt. Andere Konflikte, wie z. B. interne Zugriffskonflikte, werden durch automatisierte Protokolle vermieden oder ggf. aufgelöst. So sorgt auch ein speziell auf die Verwaltung versionierter, komplex strukturierter Entwurfsdaten zugeschnittenes und Isolation gewährleistendes Sperrprotokoll [20] für die korrekte Verarbeitung der Entwurfsdaten innerhalb nebenläufiger Werkzeuganwendungen. Eine ausführliche Beschreibung dieses Synchronisationsverfahrens für die transaktionsgeschützte Verarbeitung von Entwurfsdaten in CONCORD ist in [6] zu finden. An dieser Stelle möchten wir lediglich noch einmal herausstellen, daß die in den Kap. 2, 3 und 4 diskutierten Konzepte der Koordination von Entwurfsaktivitäten dienen und daher zunächst unabhängig vom konkreten Zugriff auf die gemeinsamen Daten sind. Die sich durch die beschriebenen Kooperationsmechanismen ergebenden gemeinsamen Datenbereiche bzw. Überlappungen von Datenbereichen betreffen lediglich die prinzipielle Sichtbarkeit der Daten. Konkrete Zugriffe auf diese Daten durch nebenläufige Werkzeuge werden (mit Hilfe des angesprochenen Sperrprotokolls) im Sinne von ACID-Transaktionen [4] gegeneinander geschützt, was eine notwendige 'Basiskorrektheit' aller Abläufe garantiert.

## 5 Zusammenwirken von Designflow- und Kooperationskontrolle

In den vorangegangenen Kapiteln wurden verschiedene Möglichkeiten hinsichtlich der Wahl der Datenbereiche, dem Grad der Vorplanung von Designflows sowie der Art der Kooperationskontrolle aufgezeigt. Wir wollen nun kurz zusammenfassen, wie all diese Konzepte sinnvollerweise kombiniert werden können, um eine ausreichende Basis für die notwendige Assistenzfunktion durch kooperative Designflows zu bekommen.

Allgemein ergibt sich der geeignete Mechanismus zur Kooperationskontrolle aus dem Grad der Vorplanung der DAs mit Kooperationsbedarf. Liegen hochgradig geregelte Abläufe innerhalb der DAs vor, so können die notwendigen Abstimmungsschritte in der Regel auch exakt vorgeplant werden. Andererseits wird es



sich bei unregelmäßigen DAs wohl als sehr schwierig erweisen, eine vollständige Vorplanung der Kooperationsaktionen vorzunehmen. Dabei steht 'geregelt Kooperationsaktion' dafür, daß ein Vorwissen über den Ablauf der der Abstimmung dienenden Arbeitsphasen existiert und die Kooperation entsprechend vorgeplant werden kann. Fehlt dieses Wissen ('ungeregelte Kooperation'), so muß eine weitgehend freie Kooperation unterstützt werden. Neben diesen Beobachtungen lassen sich die folgenden Regeln ableiten; nach dem Pfeil wird jeweils die geeignete Form der Kooperationskontrolle angegeben:

- Geregelt Anwendung  $\wedge$  getrennte Datenbereiche  $\rightarrow$  vorgeplante Kooperation: gemeinsame Sub-Designflows unter Verwendung der expliziten Kooperationsoperationen (wie in Abb. 3 beispielhaft gezeigt);
- Ungeregelte Anwendung  $\wedge$  getrennte Datenbereiche  $\rightarrow$  dynamische Nutzung der expliziten Kooperationsoperationen;
- Geregelt Anwendung  $\wedge$  gemeinsame Datenbereiche  $\rightarrow$  vorgeplante Kooperation: gemeinsame Sub-Designflows ohne Nutzung der expliziten Kooperationsoperationen;
- Ungeregelte Anwendung  $\wedge$  gemeinsame Datenbereiche  $\rightarrow$  abstrakte Kooperation.

Bei der letzten Regel ist zu beachten, daß der Automatenansatz sich auch zur vollständigen Regelung der Kooperation anwenden läßt. Aufgrund der durch ihn realisierten Zulässigkeitsprüfung<sup>1</sup> eignet er sich jedoch mehr für die Kontrolle *ehrer* unregelmäßiger Aktivitäten.

## 6 Resümee

In diesem Artikel haben wir anhand des in CONCORD realisierten Konzeptes der kooperativen Designflows ausgeführt, inwieweit die drei folgenden Dimensionen berücksichtigt werden müssen, um die erwünschte Assistenzfunktion unterstützen zu können:

- von disjunkten Datenbereichen für Entwurfsaufträge über gezielte Bereitstellung bedeutungsvoller Entwurfsobjektzustände bis hin zu gemeinsamen Datenbereichen;
- von vollständig vorgeplanten Designflows bis hin zu vollkommen offenen (ad-hoc) Flows;
- von isolierter Auftragsbearbeitung bis hin zu (kontrollierter) Kooperation.

Diese Mechanismen stellen die *Basisfunktionalität* für die erforderliche *Assistenzfunktion* bereit, mit der ein Entwerfer entsprechend einer gegebenen Entwurfsmethodik durch seine Entwurfsaufgabe geführt werden kann, ohne unnötige Einschränkungen seiner Kreativität und Entwurfsfreiheit einzugehen. Es zeigt sich, daß

abhängig von der Organisation der Entwurfsdatenbereiche und von dem Grad der Vorplanung der Designflows jeweils eine sinnvolle und passende Kooperationskontrolle bereitgestellt werden kann. Damit werden hinreichende Mechanismen zur Vorplanung und Abarbeitung kooperativer Designflows angeboten, womit sich ein sehr breites Spektrum von Entwurfsanwendungen bedienen läßt.

Gerade das Konzept der Assistenzfunktion bzw. deren Basisfunktionalität (hinsichtlich der hier diskutierten Dimensionen Datenbereiche, Grad an Vorplanung, Grad an Kooperation) macht die Unterscheidung zwischen herkömmlichen (hauptsächlich betriebswirtschaftlich orientierten) Workflows und der Klasse der Designflows offensichtlich. Damit stellt sich natürlich auch gleich die Frage, ob die einschlägigen Implementierungsansätze bzw. Workflow-Management-Systeme auch für das Einsatzszenario der Designflows geeignet und verwendbar sind. Der Beantwortung dieser Frage messen wir eine hohe Wichtigkeit zu, und ausführliche Untersuchungen dazu stehen daher auch im Brennpunkt weiterer Arbeiten.

Obwohl gegenwärtig das Prototyp-System CONCORD fertiggestellt und auch teilweise schon eingesetzt wird, sollen weiterhin systematisch einzelne Implementierungsansätze bzw. ganze Workflow-Management-Systeme auf ihre Verwendung in Designflow-Management-Umgebungen hin überprüft werden. Hierbei sind natürlich sowohl die am Markt schon etablierten Systeme wie z. B. FlowMark [13] oder WorkParty [26], als auch Laborprototypen bzw. Forschungsprojekte wie MOBILE [9], Mentor [27], Meteor [22] oder EXOTICA [16, 12] (um jeweils beispielhaft nur einige aus der Vielzahl bestehender zu nennen) zu berücksichtigen. Dabei ergeben sich u. a. die folgenden interessanten Implementierungsansätze:

- Realisierung der Ablaufkomponente (engl. workflow engine), etwa im Rahmen eines skriptbasierten, regelbasierten, petrinetzbasierten oder automatengesteuerten Ansatzes;
- Verwaltung der Abläufe und Ablaufbeschreibungen, etwa mittels relationalen oder objektorientierten Datenbanksystemen;
- Behandlung von Fehlerfällen, z. B. unter Verwendung des Transaktionskonzeptes oder Sicherungspunkten.

Sämtliche Implementierungsbewertungen und auch Implementierungsentscheidungen müssen hinsichtlich der Aspekte Erweiterbarkeit, Leistung und Verteilung geprüft werden, da dies insbesondere für einen Framework-Ansatz bzw. eine Entwurfsumgebung wie CONCORD von zentraler Wichtigkeit ist.

1. Beabsichtigte Werkzeugaufrufe müssen zuerst mit Hilfe des Automaten-Mechanismus auf Korrektheit geprüft werden.

**Danksagung.** Wir danken T. Härder, A. Frank sowie den anonymen Gutachtern für die Durchsicht des Manuskripts und die konstruktive Kritik.

## Literatur

1. ten Bosch, O.: Design Flow Management in CAD Frameworks. PhD Thesis, Delft University of Technology, The Netherlands, 1995
2. Dadam, P., Kuhn, K., Reichert, M., Beuter, T., Nathe, M.: ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen. Proc. GI-SI Jahrestagung, Zürich, Schweiz, 1995
3. Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow Management From Process Modeling to Infrastructure for Automation. Journal on Distributed and Parallel Database Systems, 3(2), April 1995
4. Gray, J., Reuter, A.: Transaction Processing: Concepts and Techniques. San Mateo, CA: Morgan Kaufmann 1993
5. Harrison, D., Newton, R., Spickelmier, R., Barnes, T.: Electronic CAD Frameworks. Proc. of the IEEE, Vol. 78, No. 2, Febr. 1990, pp. 393-417
6. Härder, T., Ritter, N.: Transaction-based Design Data Processing in the PRIMA Framework. In: Rammig, F.J. Wagner, F.R. (eds.), Electronic Design Automation Frameworks, Volume 4, Chapman & Hall 1995, pp. 3-12
7. Hsu, M. (special issue editor): Special Issue on Workflow and Extended Transaction Systems. IEEE Data Engineering Bulletin, 16 (2), June 1993
8. Hsu, M. (special issue editor): Special Issue on Workflow Systems. IEEE Data Engineering Bulletin, 18 (1), March 1995
9. Jablonski, S.: MOBILE: A Modular Workflow Model and Architecture. Proc. 4th Int. Working Conference on Dynamic Modeling and Information Systems, Noordwijkerhout, NL, September, 1994
10. Jablonski, S.: Workflow-Management-Systeme - Modellierung und Architektur. Bonn: Thompson 1995
11. Jablonski, S., Bussler, C.: Workflow Management - Modeling Concepts, Architecture and Implementation. Bonn: Thompson 1996
12. Kamath, M., Alonso, G., Günthör, R., Mohan, C.: Providing High Availability in Very Large Workflow Management Systems. Proc. 5th Int. Conference on Extending Database Technology (EDBT), Avignon, Frankreich, März, 1996, S. 427-442
13. Leymann, F.: Supporting Business Transactions via Partial Backward Recovery in Workflow Management Systems. In: Lausen, G. (Hrsg.) Datenbanksysteme in Büro, Technik und Wissenschaft, Berlin, Heidelberg, New York: Springer 1995
14. Malone, T., Crowston, K.: What is Coordination and how can it help design Cooperative Work Systems? Proc. ACM 1990 Conference on Computer Supported Cooperative Work, Los Angeles, 1990, pp. 357-370
15. Mitschang, B., Härder, T., Ritter, N.: Design Management in CONCORD: Combining Transaction Management, Workflow Management, and Cooperation Control. Proc. 6th Int. Workshop on Research Issues in Data Engineering: Interoperability of Nontraditional Database Systems (RIDE-NDS), New Orleans, 1996, pp. 160-168
16. Mohan, C., Alonso, G., Günthör, R., Kamath, M., Reinwald, B.: An Overview of the EXOTICA Research Project on Workflow Management Systems. Proc. 6th Int. Workshop on High Performance Transaction Systems, Asilomar, September, 1995
17. Nodine, M.H.; Zdonik, B.: Cooperative Transaction Hierarchies: Transaction Support for Design Applications. VLDB Journal 1, pp. 41-80, 1992
18. Ritter, N., Mitschang, B., Härder, T., Gesmann, M., Schönig, H.: Capturing Design Dynamics - The CONCORD Approach. Proc. 10th Int. IEEE Data Engineering Conference, Houston, Texas, Feb. 1994, pp. 440-451
19. Ritter, N.: DB-gestützte Kooperationsdienste für technische Entwurfsanwendungen. Forschungsbericht, Universität Kaiserslautern, 1997, in Vorbereitung
20. Ritter, N., Mitschang, B., Härder, T.: Conflict Management in CONCORD. Proc. 6th Int. Conf. on Data and Knowledge Bases for Manufacturing and Engineering, Tempe, Arizona, October, 1996
21. Rammig, F. J., Steinmüller, B.: Frameworks und Entwurfsumgebungen. Informatik-Spektrum, 15, 33-43 (1992)
22. Rusinkiewicz, M., Sheth, A.: Specification and Execution of Transactional Workflows. In: Kim, W. (ed): Modern Database Systems: The Object Model, Interoperability, and Beyond. New York: Addison-Wesley 1994
23. Vossen, G., Becker, J. (Hrsg.): Geschäftsprozeßmodellierung und Workflow-Management - Modelle, Methoden, Werkzeuge. Bonn: Thompson 1996
24. The Workflow Reference Model. Version 0.6, Workflow-Management Coalition, 1993
25. van der Wolf, P.: CAD Frameworks - Principles and Architecture. Boston: Kluwer Academic 1994

26. WorkParty, Siemens Nixdorf Informationssysteme AG, Benutzerhandbuch, Version 2.0, 1995
27. Wodtke, D., Weißenfels, J., Weikum, G., Kotz Dittrich, A.: The Mentor Project: Steps Towards Enterprise-wide Workflow Management. Proc. 12th Int. IEEE Data Engineering Conference, New Orleans, 1996