

# Objektorientierte Dokumenttypen & Prototypen zur Unterstützung kooperativer Tätigkeiten

**Guido Gryczan**

Universität Hamburg,  
AB Softwaretechnik  
Vogt-Kölln-Str. 30  
22527 Hamburg

gryczan@informatik.uni-hamburg.de

**Falk Wiegand**

Südwest LB, Stuttgart Projektgruppe  
Komunalkreditsoftware

**Heinz Züllighoven**

Universität Hamburg  
AB Softwaretechnik

## Zusammenfassung

Wir berichten über ein seit Juli 1996 laufendes Projekt bei der Südwest LB, Stuttgart. Ziel des Projektes ist die Entwicklung eines Arbeitsplatzsystems für die Vergabe von Krediten an Kommunen. Die speziellen Anforderungen des Projektes liegen im Bereich der Analyse, Visualisierung und Unterstützung von Kooperations- und Koordinationsprozessen zwischen den beteiligten qualifizierten Kreditspezialisten. Wir berichten über die konzeptionellen Grundlagen und die praktischen Erfahrungen mit der von uns verwendeten objektorientierten Methode, dem sog. Werkzeug & Material-Ansatz (vgl. [BGZ95, BGK+97]). Dabei stellen wir in den Vordergrund, wie der Bau von Prototypen mit anwendungsorientierten Dokumenttypen kombiniert wurde (vgl. [BKG+96]). Es zeigt sich, daß qualifizierte Anwender wesentlich bei der Analyse der Arbeitssituation und dem Entwurf des zukünftigen Systems beteiligt werden können. Wesentlich ist auch, daß Prototypen von ihrem fachlichen Kern nach „außen“ und nicht von der Oberfläche her gestaltet werden sollten. Dabei spielen Leitbilder und Entwurfsmetaphern eine große Rolle. Objektorientierung erweist sich in der Umsetzung dieser Konzepte als geeignete Konstruktionstechnik.

Im Artikel geben wir einen kurzen Überblick über den von uns verwendeten objektorientierten Werkzeug & Material-Ansatz. Wir schildern den Projektverlauf mit Bezug auf die verwendeten Dokumenttypen Kooperationsbilder, Wozu-Tabellen und Szenarien sowie deren Zusammenhang mit Prototypen. Wir argumentieren, daß diese Kombination von Hilfsmitteln geeignet ist, um sowohl abstrakt als auch konkret genug über fachliche und technische Fragestellungen bei der Entwicklung interaktiver Anwendungssysteme zur Unterstützung kooperativer Tätigkeiten diskutieren zu können.

## Schlüsselwörter

Werkzeug & Material Ansatz, Dokumenttypen, Kooperationsbilder, Wozu-Tabellen, T-Prototypen, Kooperative Tätigkeiten

## Die methodische Basis: Der Werkzeug & Material Ansatz (WAM)<sup>1</sup>

Das zentrale Anliegen des Werkzeug & Material Ansatzes ist es, qualifizierten Experten eines Anwendungsbereichs geeignete Arbeitsmittel (Werkzeuge) und Arbeitsgegenstände (Materialien) zur Erledigung ihrer Aufgaben zur Verfügung zu stellen. Diese Herangehensweise ist durch das *Leitbild* vom „Arbeitsplatz für die eigenverantwortliche Erledigung qualifizierter Aufgaben“ geprägt. Unter einem Leitbild verstehen wir dabei in Anlehnung an [Zül98, auch für die folgenden Definitionen]: Ein Leitbild ist eine benannte, mit Absicht eingenommene grundsätzliche Sichtweise. Es ist eine Orientierung, die von Menschen angenommen wird, anhand der sie einen Ausschnitt von Realität wahrnehmen, verstehen und gestalten. Das Leitbild bestimmt, wie Entwickler und Benutzer bei der Systementwicklung wechselseitig miteinander umgehen und beschreibt die Gestaltungsziele bei der Softwareentwicklung.

Diese Gestaltungsziele, die wertend und auf die Arbeitssituation bezogen sind, haben einen Einfluß auf die Art und Weise, wie der Umgang mit einem Anwendungssystem realisiert werden soll.

Dieser Zusammenhang von inhaltlichen Vorstellungen und gestalterischer Umsetzung wird durch *Entwurfsmetaphern*, wie Werkzeug & Material verstärkt und konkretisiert: Eine Entwurfsmetapher ist hier eine bildhafte Vorstellung, die sowohl fachlich als auch technisch konstruktiv interpretiert werden kann. Die fachliche Interpretation stellt den Zusammenhang zwischen den Komponenten des Softwaresystems und den zu bearbeitenden Aufgaben im Anwendungsbereich her. Die technische Interpretation legt fest, wie diese Komponenten mit den zur Verfügung stehenden Mitteln des objektorientierten Entwurfs und der Konstruktion realisiert werden können.

<sup>1</sup>Das Akronym WAM steht für Werkzeug, Automat, Material. Im Rahmen dieser Abhandlung sind Automaten als Entwurfsmetapher aber von geringerer Bedeutung und werden deshalb vernachlässigt. Einen Überblick über Automaten als Entwurfsmetapher gibt [Gry96].

Ein *Material* ist im Rahmen einer Aufgabenerledigung ein Arbeitsgegenstand. Materialien lassen sich in bestimmter Weise bearbeiten, d.h. verändern oder sondieren. Technisch ist eine Materialklasse durch Operationen zur Veränderung und Sondierung des Zustands von Objekten der Klasse definiert.

Komplementär ist ein *Werkzeug* ein Arbeitsmittel, mit dem Anwender in einer bestimmten Arbeitssituation ein Arbeitsgegenstand, das Material, bearbeiten. Ein Werkzeug verändert ein Material oder sondiert seinen Zustand. Mit einem Werkzeug wird sowohl eine fachliche Funktionalität ("es ist zu etwas gut") als auch eine bestimmte Art der Handhabung ("es muß richtig gehandhabt werden") verbunden. Technisch bestehen Werkzeuge aus einer Funktions- und einer Interaktionskomponente. Die Beziehung der Komponenten wird durch das Beobachter-Entwurfsmuster (vgl. [GHJV95]) beschrieben. Abbildung 1 zeigt als Beispiel das Werkzeug *Überweiser*, mit dem das Material *Überweisungsträger* bearbeitet werden kann.

Werkzeuge und Materialien innerhalb eines Leitbilds sollen für Anwender und Entwickler eine Vorstellung über die Art und Weise hervorrufen, wie Aufgaben grundsätzlich erledigt werden können. Sie grenzen andererseits den Gestaltungsspielraum für die Benutzungsschnittstelle ein.

Die produktorientierte Seite des Werkzeug & Material Ansatzes wird prozeßorientiert durch einen evolutionären Ansatz komplementiert. Dabei werden *anwendungsorientierte Dokumententypen* zur Beschreibung des Ist-Zustandes (Szenarios) in Verbindung mit Dokumenten zur Beschreibung des Soll-Zustandes (Systemvisionen) eingesetzt.

Ein *Szenario* beschreibt dabei als kurzer Prosatext in der Sprache der Anwendung alltägliche Arbeitssituationen, den Ablauf und Zweck von Handlungen und den Einsatz geeigneter Mittel. Die Beschreibung orientiert sich dabei im wesentlichen an den von *einer* Person wahrgenommenen Aufgabe. Szenarios helfen den Entwicklern, den Anwendungsbereich zu verstehen und sie schaffen die Grundlage der Zusammenarbeit mit den Anwendern.

Eine *Systemvision* beschreibt als kurzer Prosatext in der Sprache der Anwendung wie fachliche Aufgaben unter Verwendung von Werkzeugen, Automaten und Materialien des zukünftigen Anwendungssystems erledigt werden können. Projektabhängig werden verschiedene Typen von Systemvisionen geschrieben, um die Dimensionen eines Anwendungssystems (1) fachliche/technische Einbettung (2) Statik/Dynamik und (3) Überblick/Detail zu erfassen(vgl. [Fac97], [Zül98]).

Eine Systemvision hilft den Entwicklern, sich eine gemeinsame Vorstellung von der Gestaltung des zukünftigen

Abbildung 1: Werkzeug *Überweiser*

Anwendungssystems zu erarbeiten. Systemvisionen dokumentieren damit das jeweilige aktuelle Verständnis der Projektmitarbeiter über dieses Anwendungssystem. Während Szenarios unmittelbar für die Diskussion zwischen Entwicklern und Anwendern geeignet sind, können sich Anwender, besonders zu Beginn eines Entwicklungsprojektes meist keine genaue Vorstellung vom künftigen System und seines Einflusses auf die tägliche Arbeit machen. Hier helfen Prototypen.

Der Prozeß der „Visionierung“ und Diskussion zwischen den beteiligten Gruppen wird klassisch wesentlich durch den Bau verschiedener Arten von *Prototypen* unterstützt. Prototypen entsprechen in WAM den von [BKK+92] klassifizierten und beschriebenen Konzepten. In WAM besteht eine Trennung zwischen Demonstrationsprototypen und den anderen Prototyparten. *Demonstrationsprototypen* sind mit zugeschnittenen Entwicklungswerkzeugen (z.B. Visual Age) erstellte Oberflächenprototypen, die parallel zu Systemvisionen und neben den Funktionsprototypen entwickelt werden. Sie sind als Wegwerfprototypen gedacht. Dabei ist darauf zu achten, daß Demonstrationsprototypen immer nachgeordneten Charakter haben: Sie demonstrieren als Entwurfsskizze im Sinne eines Layout, was die fachliche Analyse und der fachliche Entwurf ergeben haben. Von zentraler Bedeutung sind die Prototypen im engeren Sinne, die neben einer ausgearbeiteten Benutzungsschnittstelle sowohl eine fachliche Komponente als eine diskussi-

onswürdige softwaretechnische Seite umfassen. Im Rahmen einer methodischen Vorgehensweise muß verankert werden, daß in größeren Abschnitten technische Revisionen, d.h.

Rekonstruktionen vorhandener Prototypen unter softwaretechnischen Gesichtspunkten erfolgen (können), um eine saubere konstruktive Grundlage für die evolutionäre Entwicklung der Prototypen hin zum Zielsystem zu gewährleisten.

Neben den Prototypen im engeren Sinne werden auch immer wieder Labormuster, d.h. Prototypen, die innerhalb des Entwicklerteams zur Bewertung von Designalternativen dienen, mit großem Gewinn erstellt. Sie sind nicht nur nützlich, weil sie den Entwicklern bei der Klärung eines Konstruktionsproblems helfen, sie sind oft auch Katalysatoren für neue und weitreichende Ideen für die Systemgestaltung.

Abbildung 1 zeigt einen im skizzierten Projektkontext erstellen Prototypen für das Werkzeug *Überweiser*. Prototypen sind also auch an ihrer Oberfläche von verschiedenen Dimensionen bestimmt: dem Leitbild mit seinen Metaphern (hier Werkzeug und Material an einem Arbeitsplatz), der fachlichen Analyse vorhandener Aufgaben (hier: Überweisung) und einer gemeinsamen Vorstellung über die zukünftige Arbeit (hier eine Übertragung vorhandener Arbeitsmittel auf den Computer).

Szenarios, Systemvisionen und Prototypen sind von ihrer Anlage her darauf ausgerichtet, Aufgaben die am einzelnen Arbeitsplatz erledigt werden, zu beschreiben. Szenarios und Systemvisionen sind konzeptionell nicht geeignet, kooperative Arbeitssituationen zu erfassen, da die Gesamtsicht auf kooperative Aufgaben nicht zu ihrem Focus gehört.

Wir haben bisher den methodischen Rahmen des (ursprünglichen) WAM-Ansatzes für den Entwurf und die Konstruktion von Einzelplatzsystemem skizziert. Im vorliegenden Projekt mußten wir weitere Grundlagen und Techniken für die Analyse kooperativer Arbeitssituationen schaffen. Es erwies sich als notwendig, das Repertoire der Dokumenttypen zu erweitern. Zur Verdeutlichung

beschreiben wir den Projektkontext mit seinen spezifischen Anforderungen im nächsten Abschnitt.

### Der fachliche Projektkontext

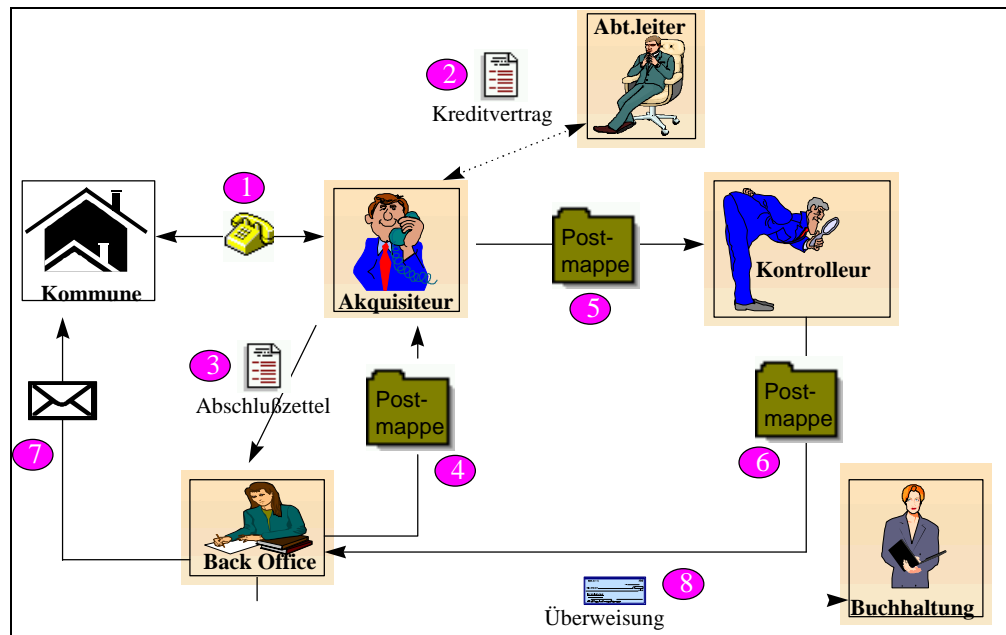


Abbildung 2: Kooperationsbild Kommunalkreditvergabe

Der Vergabeprozess für Kommunalkredite (das sind Kredite einer (Landes-) Bank an Kommunen und große öffentliche Träger) ist wesentlich durch die hohe Kooperation zwischen den beteiligten Kommunal-kreditbearbeitern und die Notwendigkeit zur schnellen Reaktion gekennzeichnet. Denn die Kreditakquisition und Kreditvergabe werden telefonisch abgewickelt und sind bereits vertragsrelevant. Die Gesamtaufgabe der Kreditvermittlung wird von unterschiedlichen Mitarbeitern in verschieden anspruchsvollen Teilaufgaben erledigt (s. Abb. 2).

Das eigentliche Kreditgespräch wird z.B. nur von dazu berechtigten und qualifizierten Kommunalkreditberatern (Akquisiteuren) geführt. Übersteigen Kredite eine bestimmte Höhe, müssen entsprechende Kompetenzträger hinzugezogen werden. Die Abwicklung eines Kredits wird von Verwaltungskräften übernommen. Bei dieser Art von Kooperation zur Vergabe eines Kommunalkredits werden unterschiedliche Dokumente ausgetauscht. Über noch zu erledigende Aufgaben wird zwischen den Beteiligten direkt kommuniziert. Neben der organisatorisch und rechtlich bedingten Arbeitsteilung kommt, wie gesagt, als wesentliches Merkmal gegenüber anderen Kreditformen hinzu, daß Kommunalkredite fernmündlich vergeben werden.

Die notwendige rasche Reaktion auf Kundenanforderungen erzwingt, daß Arbeitsplatzsoftware in dem Sinne unterstützend wirkt, als die

situationspezifischen Anforderungen bei der telefonischen Verhandlung berücksichtigt und der anschließenden schnellen Kreditabwicklung ermöglicht wird.

Für die Entwickler hieß dies, neben den einzelnen Teilaufgaben vor allem das enge Wechselspiel in der Zusammenarbeit der verschiedenen Personen zu verstehen. Die Art der Kooperation erforderte also, in jeder Arbeitssituation die selbständigen fachlichen Anteile und koordinierenden Tätigkeiten genau zu analysieren.

Hier kam es vor allem darauf an, geeignete Modellierungsmittel zur Darstellung der Kooperation einzusetzen, um die Rückkopplung mit den Experten des Anwendungsbereichs über die gesamte Dauer des Entwicklungsprozesses zu ermöglichen.

Da bisher keine adäquate Anwendungssoftware im Einsatz war, kam weiterhin dem Prototyping eine besondere Bedeutung beim Entwurf des zukünftigen Systems zu.

### **Kooperationsbilder & Wozu-Tabellen im Kontext übergreifender Aufgaben bei kooperativer Tätigkeit**

Die oben beschriebene Ausgangslage stellte besondere Anforderungen an den Entwicklungsprozeß. Wir haben festgestellt, daß sich mit Szenarios, Glossaren und Systemvisionen die verschiedenen Aufgaben und Aktivitäten sehr gut aus der Sicht der einzelnen Arbeitsplätze beschreiben lassen. Was fehlt ist die Gesamtsicht. Diese Gesamt-sicht kooperativer Arbeit kann nicht einfach erstellt werden, da kaum einer der Anwendungsexperten sich mit diesem Aspekt der täglichen Arbeit befassen mußte. Dies gilt übrigens nicht nur für das konkrete Projekt, sondern aus unserer Erfahrung für alle Projekte, in denen komplexe Kooperationsformen unterstützt werden sollen.

Auf der allgemeinen Ebene ist ein zweites Problem angesiedelt: Die Menge der Szenarios für einen gesamten Arbeitsbereich, etwa einer Bank oder eines Krankenhauses, stellt eine solche Textfülle dar, daß sie für einen Überblick über die Gesamtsituation nur begrenzt geeignet ist und damit nicht alleinige Grundlage einer gemeinsamen Diskussion der Beteiligten über die Kooperationszusammenhänge sein kann.

Darüber hinaus muß das Verhältnis kooperativer und individueller Aufgaben klar differenziert werden, um für die Experten des Anwendungsbereichs ein System anbieten zu können, das sich bruchlos in die tägliche Arbeit einbetten läßt. Dazu definieren wir *kooperative Tätigkeiten* in Anlehnung an [Zül98] wie folgt: Kooperative Tätigkeiten sind Tätigkeiten, bei der verschiedene Personen geplant und koordiniert zusammenarbeiten, um ein gemeinsames Ergebnis zu erreichen. Aus dieser bewußt allgemein gehaltenen Definition folgern wir zunächst, daß kooperative und individuelle Tätigkeiten zur Erledigung einer Aufgabe nicht in den Kategorien „besser/schlechter“ oder „angestrebt/nicht angestrebt“ betrachtet werden sollten.

Vielmehr halten wir pragmatisch fest, daß die Rechnerunterstützung kooperativer Tätigkeiten uns als Entwickler zwingt, die Art der Kooperation im Anwendungsbereich zu hinterfragen, um darauf aufbauend entscheiden zu können, welches Kooperationsmodell und welche Kooperationsmittel im Anwendungssystem realisiert werden sollten. Denn erst wenn diese analytisch konzeptionellen Fragen geklärt sind läßt sich ein konstruktiver Ansatz durch Prototypen verfolgen.

Einen ersten Einstieg zur Lösung der angesprochenen Probleme bieten die auch in diesem Projekt erfolgreich eingesetzten sog. *Kooperationsbilder* (vgl. [KRW96]). Diese Bilder basieren im wesentlichen auf allgemeinverständlichen Piktogrammen. Sie stellen dar, welche Arbeitsgegenstände und Informationen auf welchem Weg zwischen den Beteiligten ausgetauscht werden. Dabei können über Pfeile und Numerierungen das Zusammenspiel einzelner Teilaufgaben und die Art ihrer Erledigung dargestellt werden (s. Abb. 2). Faßt man mehrere zusammengehörige Aufgaben in einem Bild zusammen und läßt die Nummern weg, dann ergibt sich ein Überblick über diskussionswürdige „Brennpunkte“ und „Schleifen“ in der Kommunikation und Kooperation.

Kooperationsbilder werden in gemeinsamen Sitzungen mit den Beteiligten erstellt und diskutiert. Sie dienen zum einen zur Repräsentation der Ist-Situation, können aber ebenso gut für die Darstellung der möglichen Veränderungen beim Einsatz des zukünftigen Systems verwendet werden. Abbildung 2 zeigt das Kooperationsbild zur Vergabe von Kommunalkrediten, aus dem ersichtlich wird, wer mit wem kommuniziert und welche Dokumente austauscht, um einen Kommunalkredit zu vergeben. Dabei sind die organisatorischen und rechtlichen Randbedingungen explizit berücksichtigt.

Anhand dieses Kooperationsbilds und der ergänzenden Wozu-Tabelle (vgl. Abbildung 3) wird für Entwickler und Anwender der Diskussionsgegenstand für *übergreifende Aufgaben* geschaffen. Nach [KraRatWet96] und [Rat97] sind übergreifende Aufgaben durch folgende Merkmale charakterisiert:

- Sie erfordern die Zusammenarbeit einer Vielzahl von Einzelpersonen mit zum Teil unterschiedlichen Qualifikationen.
- Sie erfordern ein hohes Maß von Flexibilität, da die Aufgabenerledigung häufig von äußeren Faktoren abhängig ist, die im vorhinein nicht vollständig bestimmt werden können.
- Die Durchführung übergreifender Aufgaben erfordert die *Koordination* der von Einzelnen zu erledigenden Aufgaben. Unter Koordination wird dabei nach [Sta89] die Abstimmung und Harmonisierung von Organisationsmitgliedern im Hinblick auf den Zweck der Aufgabe verstanden. Zu den koordinativen Tätigkeiten gehören z.B. die Weitergabe von Dokumenten, die

zeitliche Koordinierung und das Signalisieren von Änderungen.

- Teile einer übergreifenden Aufgabe werden an *Gruppen-* oder *Funktionsarbeitsplätzen* erledigt. Ein Gruppenarbeitsplatz ist ein gemeinsamer Arbeitsplatz, bei dem zwischen den Mitgliedern einer Gruppe eine Konvention über dessen Benutzung hergestellt wurde. Ein Funktionsarbeitsplatz ist speziell auf die Durchführung einer Aufgabe zugeschnitten und bietet nicht die Flexibilität oder die Gestaltungsmöglichkeiten eines persönlichen oder Gruppenarbeitsplatzes.

- die Anzahl der an der Kreditvermittlung beteiligten Personen ist klein und diese Personen bilden eine Arbeitsgemeinschaft, die mit ihren Kooperationspartnern und Aufgaben vertraut ist,
- die Zeitvorgaben für die Vergabe von Kommunkrediten sind eng, d.h. sie liegen im Stundenbereich und können unter keinen Umständen verlängert werden, da das Geschäft sonst nicht zustande kommt.

Es zeigt sich, daß dieses allgemeine Konzept von übergreifenden Aufgaben den Entwicklern als analytisches Instrument hilft, die vorhandene Arbeitssituation überhaupt ein-

	Wer	erledigt was	mit wem/was	WOZU
1	Kundenberater	erstellt Angebot, bestätigt Auftrag	Kunde, Bevollmächtigter ( Abt.Leiter)	um Angebot und Abschluss zu tätigen und zu bewilligen
2	Kundenberater	erstellt Abschlußzettel	Kundeninformationen und Unterlagen	um die Grundvoraussetzungen für die Bearbeitung des Kredits zu schaffen
3	Sachbearbeiter	erstellt Kreditunterlagen und Überweisungsträger	Abschlusszettel, Formulare	aus gesetzlichen und verwaltungstechnischen Gründen
4	Kundenberater	prüft die Korrektheit der Kreditunterlagen	Abschlusszettel, Kreditunterlagen	um die sachl. Korrektheit der Kreditunterlagen zu gewähren
5	Kreditkontrolle	prüft die Korrektheit der Kreditunterlagen	Kreditunterlagen	aus gesetzlichen und verwaltungstechnischen Gründen
6	Sachbearbeiter	versendet Kreditunterlagen an Kunden	Kreditunterlagen	aus gesetzlichen und verwaltungstechnischen Gründen
7	Sachbearbeiter	erstellt und veranlasst Überweisungen	Kreditunterlagen, Formulare	um Kredit und Maklerprovision auszuzahlen

**Abbildung 3: Wozu-Tabelle Kommunalkreditvergabe**

- Die Beteiligten an einer übergreifenden Aufgabe verfügen über ein sicheres Handlungswissen bzgl. der Erledigung ihrer Anteile an der übergreifenden Aufgabe.
- Übergreifende Aufgaben sind mit engen zeitlichen Vorgaben versehen.

Im Sinne dieser Begriffsdefinition ist die Vergabe von Kommunalkrediten eine übergreifende Aufgabe, denn

- die Kreditvergabe wird von unterschiedlich qualifizierten und berechtigten Personen durchgeführt,
- die Anforderungen der Kunden der kreditnehmenden Kommunen bestimmen den Ablauf der Kreditvermittlung,
- die Reihenfolge der zu erledigenden Aufgaben muß zwischen den beteiligten Personen ständig abgestimmt werden; dies gilt auch für die weiterzuleitenden Arbeitsmaterialien,
- die Gestaltung der Verwaltungsarbeitsplatzes („Back Office“) geht in Richtung eines Funktionsarbeitsplatzes,

schätzen und verstehen zu können. Darüber hinaus bildet das Konzept die Grundlage, ein entsprechendes Kooperationsmodell für das Anwendungssystem zu entwickeln.

Durch den Einsatz von Kooperationsbildern wird deutlich, wie und auf welche Art und Weise die Beteiligten bei der Erledigung einer übergreifenden Aufgabe kooperieren. In unserem Fall ist dabei besonders die Weitergabe der Materialien von einem Arbeitsplatz zu einem anderen von Bedeutung.

Diese Analyse ermöglicht, die Art der Softwareunterstützung für die Kooperation zu gestalten. Kooperationsbilder liefern Anwendern und Entwicklern in Besprechungen und Workshops die Basis, um ein gemeinsames Verständnis über Charakter und Ablauf kooperativer Tätigkeiten zu erarbeiten. Kooperationsbilder sind gleichzeitig die Grundlage, um die Veränderung bei organisatorischen Abläufen und in der Verteilung von kooperativen Aufgaben zwischen den Anwendern zu diskutieren.

In den Kooperationsbildern wird festgehalten, daß im Rahmen einer Kooperation Gegenstände und Informationen

ausgetauscht werden. Was in der Gesamtsicht fehlt, ist die Angabe des Motivs für diese Kooperation zwischen Personen. Zu diesem Zweck werden Wozu-Tabellen benötigt. Eine *Wozu-Tabelle* ist eine Tabelle, in der für eine kooperative Tätigkeit festgehalten wird, wer macht was mit wem oder womit und wozu.

Konzeptionelle Grundlage der Wozu-Tabellen ist eine entsprechende Technik aus der Object Behavior Analysis (OBA) nach [RubGol92], und [Heeg95]. Dort werden Aufgaben nach dem Muster „wer - macht was - womit - welchen Dienst muß der Gegenstand bereitstellen“ beschrieben (vgl. [Rat97]).

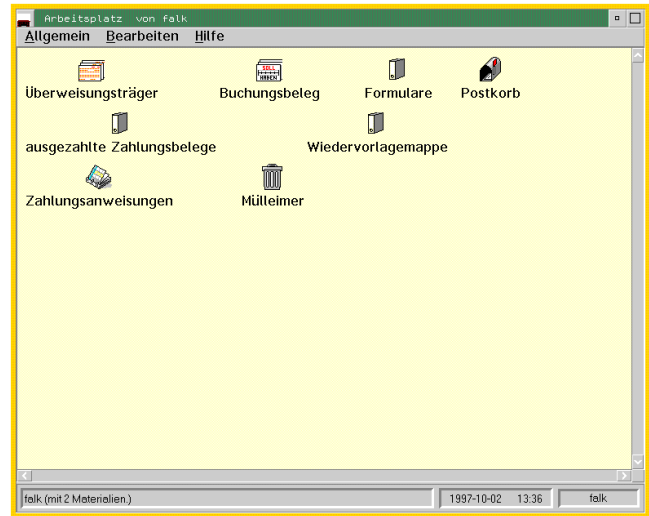
Wozu-Tabellen sollen im wesentlichen den Zweck der kooperativ zu erledigenden Aufgaben festzuhalten. Dadurch wird bieten sie im Zusammenhang mit Kooperationsbildern sowohl den Anwendern als auch den Entwicklern die Möglichkeit, nicht nur den Ist-Zustand zu erfassen, sondern auch über Änderungen in der Organisation der übergreifenden Aufgabe zu diskutieren.

Gleichzeitig wird durch die Kombination von Kooperationsbildern und Wozu-Tabellen deutlich, welche Aufgaben ein Mitarbeiter alleine erledigen kann und soll, welche Aufgaben von anderen und welche nur gemeinsam erledigt werden können. Diese Trennung führt uns zur Abgrenzung der Konzepte *Arbeitsplatz* und *Arbeitsumgebung*. Auch dies hat wieder zentralen Einfluß auf die Gestaltung von Anwendungssystemen und damit auch auf die zu erstellenden Prototypen.

Die *Arbeitsumgebung* ist der Ort für die anwendungsfachlich motivierte Anordnung von Werkzeugen, Materialien und anderen Gegenständen, die bei der Erledigung von Aufgaben griffbereit sein müssen. Die Arbeitsumgebung ist gegen den Zugriff von außen geschützt. Wenn nur die Arbeit eines einzelnen Benutzers unterstützt werden soll, bezeichnen wir die Arbeitsumgebung auch als (Einzel-) *Arbeitsplatz*. Soll ein Anwendungssystem für die Unterstützung kooperativer Arbeit ausgelegt sein, dann gibt es außer dem persönlichen Arbeitsplatz noch weitere Komponenten und Orte, die zur Arbeitsumgebung gehören.

In einer Arbeitsumgebung werden neben den Werkzeugen und Materialien für die individuelle Arbeit auch die *Kooperationsmittel* zur Erledigung arbeitsteiliger Aufgaben bereitgestellt. Unter einem Kooperationsmittel verstehen wir Materialien, die explizit Teil der Arbeitsumgebung sind und dort speziell für die Unterstützung der Kooperation bereitliegen. Kooperationsmittel sind z.B. Postfächer oder Vorgangsmappen. In Abbildung 4 wird die prototypische Arbeitsumgebung eines Kommunalcreditberaters dargestellt. Zu sehen ist ein Arbeitsplatz mit seinen fachlich motivierten Werkzeugen und Materialien und einem Postkorb für den (adressierten) Austausch von Materialien.

Diese Aufteilung verdeutlicht unser Konzept der Kooperationsunterstützung: Für den Kommunalcreditbearbeiter ist



**Abbildung 4: Arbeitsumgebung eines Kommunalberaters**

es sowohl wesentlich innerhalb seiner privaten Arbeitsumgebung die Werkzeuge und Materialien zur Erledigung seiner individuellen Arbeit vorzufinden als auch die Gegenstände, die ihm die Kooperation mit anderen Mitarbeitern erlauben. Grundlage für diese Integration eines minimalen expliziten Kooperationsmodells (Postaustausch von Arbeitsmaterialien) in die Arbeitsumgebung sind dabei die Ergebnisse der Analyse auf der Basis der beschriebenen Dokumententypen.

Ein Ergebnis unserer Projektarbeit ist also, daß erst das Zusammenspiel bestimmter anleitender Vorstellungen, dazu passender Konzepte, Darstellungsmittel und Vorgehensweise die Grundlage für ein erfolgreiches Prototyping bieten. Die in Abbildung 4 dargestellte Umgebung ist das Ergebnis eines in diesem Sinne iterativ durchgeführten Prototyping-Prozesses. Im nächsten Abschnitt beschreiben wir, wie dieser Prozeß durchgeführt wurde.

#### T-Prototypen & Werkstattbesuche

Die Ausgangslage des hier beschriebenen Projekts läßt mit Blick auf den Stellenwert des Prototyping wie folgt charakterisieren:

- Das Entwicklerteam hatte zum Projektstart keine Erfahrung in der Herstellung objektorientierter Software.
- In der Entwicklungsorganisation war nicht durchgängig Wissen über die Verbindung von Host-Systemen und Client-Server Architekturen vorhanden.
- Im Anwendungsbereich wurde keine Software zur durchgängigen Unterstützung der Arbeitsaufgaben eingesetzt.

Vor dem Hintergrund dieser Ausgangslage sollte deutlich sein, daß die Konstruktion von verschiedenen Arten von



Prototypen unbedingt erfolgen mußte, um die folgenden Ziele zu erreichen:

1. Für die Anwender mußte deutlich werden, daß der aufwendige Analyseprozeß unter Verwendung der oben genannten Dokumenttypen zu sinnvollen und operationalisierbaren Ergebnissen führt, die ihre Arbeit unterstützen.
2. Für das Entwicklerteam war entscheidend, daß die Visionen nicht den Charakter von Illusionen haben, d.h. die Konstruierbarkeit der angestrebten Lösung mußte gezeigt werden.
3. Für das Entwicklungsmanagement mußte deutlich werden, daß der innovative Prozeß in absehbarer Zeit mit nachvollziehbarem Aufwand zu tragfähigen Ergebnissen (Zielsystemem) führt, die stufenweise in die bestehende Umgebung integriert werden können.

Um diese Ziele zu erreichen, wurden auf der Basis der Dokumenttypen im wesentlichen zwei Arten von Prototypen entwickelt: Demonstrationsprototypen zur Darstellung des prinzipiellen Umgangs mit dem System und Labormuster für den Nachweis der Realisierbarkeit technischer Entwurfsentscheidungen (hier vor allem die Verbindung von dezentralem Server und zentralem Host-System). Abbildung 5 verdeutlicht die Herangehensweise: Demonstrationsprototypen werden „breit“ angelegt, um an der Oberfläche die prinzipielle Gestaltung und Handhabung des Anwendungssystems zu verdeutlichen. Labormuster werden „tief“ entwickelt, um für die Entwicklerorganisation den Nachweis der technischen Realisierbarkeit zu liefern. Die Verbindung von Demonstrationsprototypen und Labormustern geschieht in der durch die Graphik angedeuteten „T“ Form, so daß Anteile der fachlichen Arbeit durchgängig realisiert werden können.

Fachlicher Gegenstand der Demonstrationsprototypen sind dabei jeweils Aufgaben, die anhand der Dokumenttypen als wesentlich erkannt worden waren. Zur Bewertung der Prototypen durch die Anwender werden *Drehbücher* erstellt. Ein Drehbuch schildert eine fachliche Arbeitssituation, in der nahe an der Alltagswirklichkeit bestimmte Aufgaben zu erledigen sind. Die Szenen des Drehbuchs müssen dann unter Verwendung des Prototypen „durchgespielt“ werden. Der Wert von Drehbüchern liegt

vor allem darin, daß damit eine Bewertung der Prototypen ermöglicht wird, die sich auf die Erledigung relevanter Aufgaben bezieht. Damit wird bei der Vorführung von Demonstrationsprototypen oder auch bei der testweisen Erprobung durch Anwender die Gefahr vermieden, sich durch interaktives „Herumspielen“ oder Diskussionen über Details des Bildschirmlayouts in Unwesentlichem zu verlieren. Vielmehr fördert die Aufgabenorientierung der Drehbücher fachlich motivierte Diskussionen zwischen Entwicklern und Anwendern über die Angemessenheit der modellierten Werkzeuge und Materialien und der angebotenen Umgangsformen. Als Randbemerkung sei erwähnt, daß sich solche fachlich motivierten Drehbücher auch für die Bewertung von Standardsoftware sehr gut eignen.

Um die Erledigung einer Aufgabe vollständig durch Prototypen abdecken zu können, ist die Implementation von Funktionalität im Sinne des dargestellten T in Abbildung 5 notwendig. Durch diese Vorgehensweise werden die genannten Ziele erreicht.

Im Entwicklungsprozeß werden für die Demonstration der Prototypen *Werkstattbesuche* durchgeführt. Werkstattbesuche sind „informelle Treffen“ bei den Entwicklern. Dabei stellen Entwickler in ihrer Arbeitsumgebung den Anwendern die „in Arbeit befindlichen“ Prototypen vor. Dabei können natürlich schon die o.g. Drehbücher verwendet werden. Diese Art der Demonstration von Prototypen hat mehrere strategische Vorzüge gegenüber den normalen Arbeitstreffen, Workshops oder gar Reviews. Zunächst ist die Atmosphäre informell. Da immer nur einzelne Anwender einen solchen Werkstattbesuch machen, hat die Kommunikation einen eher persönlichen Charakter. Viele Problempunkte lassen sich so vorab, d.h. vor offiziellen Projektterminen oder Reviews klären. Ein weiterer Vorteil ist der Beitrag zur kontinuierlichen Zusammenarbeit zwischen Entwicklern und Anwendern. Auch ehe Prototypen vollständig fertiggestellt sind, lassen sich bei einem Werkstattbesuch Teilergebnisse präsentieren. Niemand wird die Erwartung an ein fertiges Produkt haben. Trotzdem wird deutlich, daß etwas passiert, in welche Richtung die Entwicklung geht und daß die Anwender tatsächlich einbezogen werden.

#### Diskussion und Ausblick

Wir haben in diesem Artikel unsere Vorgehensweise und die Dokumenttypen zur Analyse und Gestaltung objektorientierter Anwendungssysteme für kooperative Tätigkeiten nach dem Werkzeug & Material Ansatz vorgestellt.

Dazu haben wir Kooperationsbilder und Wozu-Tabellen als Dokumenttypen für die Analyse und Beschreibung kooperativer Tätigkeiten eingeführt. Auf der Grundlage dieser Dokumenttypen haben wir gezeigt, wie ein Kooperationsmodell unter Verwendung geeigneter Kooperationsmittel bruchlos in die durch Software

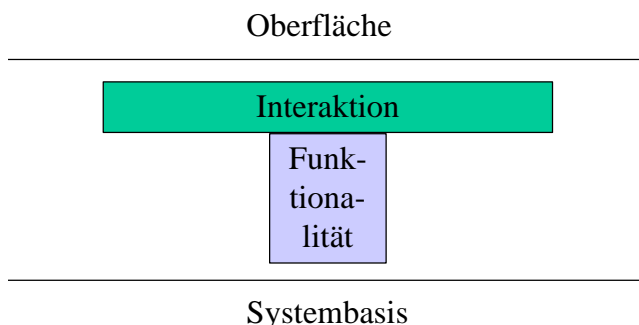


Abbildung 5: Konzept des T-Prototypen

modellierte Arbeitsumgebung der Anwender integriert werden kann.

Dabei haben wir betont, daß die im Entwicklungsprozeß erstellen Prototypen von „innen“ nach „außen“ konstruiert werden, d.h. ausgehend von dem als fachlich relevant erkannten Kern von Aufgaben (innen) wird die Oberfläche (außen) eines Prototypen konstruiert. Diese Vorgehensweise steht im Gegensatz zu der von visuellen Programmierumgebungen nahegelegten Vorgehensweise, Applikationen von der Oberfläche her zu konstruieren. Nach unserer Überzeugung stehen dem nicht nur die im Artikel genannten fachlichen Gründe entgegen. Vielmehr stößt diese Vorgehensweise bei praxisrelevanten größeren Anwendungen auch schnell an softwaretechnische Grenzen bzgl. der Modularisierung und anderer softwaretechnischer Prinzipien der objektorientierten Konstruktion (vgl. [BKS+95], [BBL+96]). Mit der Integration expliziter Kooperationsmittel in eine Arbeitsumgebung haben wir einen ersten Schritt in die Richtung Arbeitsumgebung für kooperative Tätigkeiten gemacht. Weitere Schritte müssen folgen, um andere Kooperationsmodelle zu integrieren. Dazu zählt die Unterstützung impliziter Kooperation, d.h. des konkurrierenden Zugriffs auf knappe Ressourcen oder die Unterstützung von Gruppenarbeitsplätzen, bei denen vor allem die Problematik des Anwenderwechsels am Arbeitsplatz in den Vordergrund tritt.

#### Danksagung

Dieser Artikel basiert auf Arbeiten die die Autoren zusammen mit dem Projektteam „Kommunalkreditsoftware“ der Südwest LB, namentlich Miriam Witcisk, Bernd Gruber, Andreas Frosch und Thomas Schlegel durchgeführt haben. Unsere Kolleginnen an der Universität Hamburg Christiane Floyd, Anita Krabbel, Sabine Ratuski und Ingrid Wetzels haben im Krankenhausprojekt „Alten Eichen“ die konzeptionellen Grundlagen für die hier vorgestellten Techniken der Analyse kooperativer Tätigkeiten geleistet.

#### Literatur

[BBL+96] D. Bäumer, W. Bischofberger, H. Lichten, H. Züllighoven: User Interface Prototyping - Concepts, Tools, and Experience Proceedings of the 18th International Conference on Software Engineering, Berlin, March 25-29, 1996, IEEE Computer Society Press, S. 532 - 541.

[BGZ95] U. Bürkle, G. Gryczan, H. Züllighoven: Object-Oriented System Development in a Banking Project: Methodology, Experience, and Conclusions. In: Human-Computer Interaction, Special Issue: Empirical Studies of Object-Oriented Design, Volume 10, Numbers 2 & 3, 1995, S. 293-336. Lawrence Erlbaum Associates Publishers Hillsdale, New Jersey, England.

- [BGK+97] Bäumer, D., Gryczan, G., Knoll, R., Lilienthal, C., Riehle, D., Züllighoven, H.: Framework Development for Large Systems. Communications of the ACM, October 97, Vol. 40, No. 10, S. 52 - 59.
- [BKG+96] D. Bäumer, R. Knoll, G. Gryczan, H. Züllighoven: Large Scale Object-oriented Software-Development in a banking Environment - An Experience Report. Object Oriented Programming: 10th European conference; proceedings / ECOOP '96, Linz, Austria, Juli 8 - 12, 1996. Pierre Cointe (ed.). - Berlin: Springer, 1996. S. 73 - 90.
- [BKK+92] Budde, R.; Kautz, K.; Kühlenkamp, K.; Züllighoven, H.: Prototyping. Springer 1992.
- [BKS+95] D. Bäumer, R. Budde, K.-H. Sylla, G. Gryczan, H. Züllighoven: Objektorientierte Konstruktion von Software-Werkzeugen und -Materialien. Informatik-Spektrum, Band 18, Heft 4, August 1995, Springer-Verlag, Berlin, Heidelberg. S. 203-210.
- [Fac97] P. Fach: Prototypen im Rahmen der Werkzeug-Automat-Material-Metapher. In diesem Tagungsband.
- [GHJV95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley 1995.
- [Gry96] G. Gryczan: Prozeßmuster zur Unterstützung kooperativer Tätigkeit. Wiesbaden: DUV, Dt. Univ.-Verl., 1996 (DUV: Informatik)
- [Heeg95] G. Heeg: Objektorientierung als neue Technologie. In: Fachseminar Objektorientierung, Systems Kongreß, 1995, S. 81 - 90.
- [KRW96] A. Krabbel, I. Wetzels, S. Ratuski: Analysetechniken für übergreifende Aufgaben In: Jürgen Ebert (Hrsg.): GI-Fachtagung Softwaretechnik '96, September 1996, S. 65 - 72.
- [Rat97] S. Ratuski: Vorgehensweise zur Auswahl von Standardsoftware am Beispiel von Krankenhausinformationssystemen. Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1997.
- [RG92] K. S. Rubin, A. Goldberg: Object Behavior Analysis. In Communications of the ACM, Vol. 35, No. 9, 1992, S. 48 - 62.
- [Sta89] H. Staehle: Management. Vahlen-Verlag, München, 1989.
- [Zül98] H. Züllighoven: Das objektorientierte Konstruktionshandbuch nach dem Werkzeug- & Material-Ansatz. dpunkt.verlag, 1998 (in Arbeit).



