

ConceptTalk : Kooperationsunterstützung in Softwareumgebungen¹

Carlos Maltzahn
Universität Passau
Postfach 2540
D-8390 Passau

Thomas Rose
Dept. of Computer Science
University of Toronto
Toronto, Canada

Die meisten CAD-Umgebungen betonen die Unterstützung einzelner Arbeitsplätze und helfen nur sekundär bei deren Kooperation. Wir schlagen einen umgekehrten Ansatz vor: Entwürfe entstehen im Rahmen von interagierenden Sharing-Prozessen, die den gemeinsamen Zugang aller Beteiligten zu Konzepten, Aufgaben und Ergebnissen strukturieren. Dieser Ansatz und seine Konsequenzen werden am Beispiel des Software Engineering dargestellt. Auf der Basis einer Formalisierung dieser Prozesse steuert der ConceptTalk-Prototyp eine verteilte Softwareumgebung und spezielle Kommunikationswerkzeuge über das Wissensbanksystem ConceptBase. Erfahrungen mit ConceptTalk unterstützen ein neues Paradigma, das ein Informationssystem als Medium für komplexe Kommunikation betrachtet.

1 Einleitung

In der Industrie ist es heute Standard, daß selbst komplexe und innovative Entwicklungsprojekte in streng begrenzter Zeit durchgeführt werden müssen. Hieraus ergibt sich die Tendenz, solche Projekte mit vielen Mitarbeitern auszustatten, um Zeit einzusparen. Mit der Zahl der Mitarbeiter wächst der Aufwand, der nötig ist, um die Aufgaben optimal auf die Entwickler zu verteilen und die Ergebnisse dieser Verteilung zu einem Ganzen zusammenzufügen. Die Aufgaben sind i.a. zu komplex, als daß eine Aufteilung unmittelbar ersichtlich ist, und so stark vernetzt, daß sie nicht unkontrolliert getrennt gelöst werden können. Es sind Konzepte der *Kooperation* zur Verteilung von Aufgaben und die Integration von Arbeitsergebnissen gefordert.

Heutige Software-Entwicklungsumgebungen [14] konzentrieren sich auf das Produkt. Aus dieser Perspektive sind Software-Systeme komplexe Entwurfsobjekte, die in Komponenten strukturiert sind und durch die Modifikation der Komponenten entwickelt und gewartet werden. Softwaredatenbanken [2], [5] und Projektdatenbanken [13] modellieren die Struktur von Software-Systemen und verwalten Komponentenversionen. Die Strukturierung und Konfiguration von Systemen bildet eine notwendige Plattform für kooperative Projekte, weil sie den Zugriff zu einer großen Menge von Teilentwürfen organisiert und komplexe Zusammensetzungen mehrerer Systemvarianten verwaltet. Transaktionskonzepte regeln den Zugriff auf bestehende Entwürfe, die in einer gemeinsamen Datenbasis verwaltet werden. Ein Transaktionskonzept kann durch ein Datenbankverwaltungssystem oder durch Operationen auf Arbeitsbereiche innerhalb einer Entwicklungsumgebung gegeben sein [10]. Unabhängig davon sollen

¹ Diese Arbeit wird z.T. von der Deutschen Forschungsgemeinschaft im Schwerpunktprogramm "Objektbanken für Experten" (Ja-445/1-2) gefördert.

Transaktionskonzepte den Mehrbenutzerbetrieb möglichst *unsichtbar* erscheinen lassen, da dadurch der Zugriff auf die Datenbasis einfach und vor allem sicher ist. Aufgrund der damit einhergehenden technischen Synchronisationsmechanismen müssen sich jedoch eng kooperierende Entwickler bei der Bearbeitung des gleichen Entwurfs abwechseln. Zusätzliche Koordination ist erforderlich, um Interaktionen der Entwickler wieder *sichtbar* zu machen. Es gilt, Wege zu finden, die diesen Umweg überflüssig machen, ohne die Sicherheit von Entwurfsobjektbeständen zu gefährden.

Produktorientierte Software-Entwicklungsumgebungen reichen jedoch nicht aus, um Teilentwürfe auf kooperative Weise aufeinander abzustimmen. Mitarbeiter und Kooperation sind als unabdingbare Bestandteile von Entwicklungsumgebungen zu berücksichtigen. Unstrukturierte Sichtbarkeit von Interaktionen führt allerdings zur Informationsüberladung der Mitarbeiter. Der Versuch, die Sichtbarkeit durch ein technisches System zu strukturieren, bedeutet aber eine mechanische Strukturierung der Gruppenkommunikation. Sind die gewachsenen Kooperationsbeziehungen der Gruppe zur mechanischen Kooperationsunterstützung inkompatibel, so führt dies zu Abwehrreaktionen der Mitarbeiter [8]. Um dies zu verhindern ist Kooperation auch als soziale Aktivität zu studieren und durch verträgliche Konzepte zu unterstützen: Z.B. werden in der Gruppe Strategien besprochen, Vereinbarungen getroffen, Meldungen gemacht und Ergebnisse begutachtet.

Wir bezeichnen diese Vorgänge als *Kooperationsprozesse*. Die Ereignisse eines solchen Prozesses sind Operationen, die allen Beteiligten in irgendeiner Weise kommuniziert werden. Dies kann durch Nachrichten geschehen, aber auch durch Manipulationen auf einem Objekt, welches allen Beteiligten sichtbar ist. In dieser Arbeit stellen wir ein System vor, das Kooperationsprozesse in Entwurfsverwaltungssysteme integriert. Ausgehend von einer Analyse der Anforderungen in Kapitel 2 skizzieren wir in Kapitel 3 unser Konzept sogenannter *Sharing-Prozesse* zur Steuerung und Dokumentation entwurfsbezogener Kooperation. Dieses Modell ist formal in der Wissensrepräsentationssprache Telos beschrieben. Dies ermöglicht es zum einen, die Bezüge der Kooperationsprozesse zu den von einer kommerziellen Softwareumgebung geleisteten Objektverwaltungsaufgaben herzustellen; zum anderen strukturiert und dokumentiert das Prozeßmodell den Nachrichtenaustausch zwischen den Entwicklern in asynchroner (Mail = elektronische Post) oder synchroner (Talk = Realzeitkonferenz) Gruppenarbeit. Diese konzeptbasierte Integration von Objektverwaltungs- und Kommunikationsaufgaben motiviert auch den Namen des von uns entwickelten Prototypsystems, *ConceptTalk*. Die prototypische Realisierung durch *ConceptTalk* wird in Kapitel 4 vorgestellt, während das abschließende Kapitel 5 über einige Erfahrungen und Schlußfolgerungen berichtet.

2 Kooperationsunterstützung in Entwicklungsprozessen

Es ist nützlich, sich als Metapher ein Software-Entwicklungsbüro vorzustellen, in dem sich Mitarbeiter regelmäßig treffen und besprechen, was als nächstes zu tun ist. Dabei kristallisieren sich Aufgaben heraus, die entweder sofort gelöst werden oder - bei umfangreicheren Projekten - an einzelne Personen oder kleinen Gruppen delegiert werden. In den Konferenzen werden auch Ergebnisse vorgeführt und begutachtet, Verbesserungsvorschläge gemacht oder Testabläufe vereinbart. Nachteilig ist, daß erstens die Entwurfsobjekte nicht direkt zur Verfügung stehen, zweitens nur durch individuell hastig notierte Stichwörter dokumentiert wird und drittens jeder Teilnehmer physisch zugegen sein muß. Ist letzteres nicht der Fall, wird die fehlende Dokumentation der Konferenz noch schwerwiegender. Eine auf vernetzten Workstations simulierte Konferenz würde diese Nachteile alle beheben: Entwurfsobjekte können eingeblendet, sogar manipuliert werden, Ereignisse in einer Konferenz können vom System protokolliert werden, und die Konferenzteilnehmer können physisch über die ganze Welt verteilt sein. In großen Organisationen können mehrere Konferenzen gleichzeitig stattfinden, in denen die jeweiligen Teilnehmer gleiche Entwurfsobjekte sehen, aber unterschiedlich manipulieren. Oder es werden in verschiedenen Konferenzen ähnliche Aufgaben identifiziert und unabsichtlich an unterschiedliche Personen delegiert.