

# Groupware - Kommunikation, Kollaboration, Koordination

## Einleitende Zusammenfassung

In vielerlei Hinsicht entzieht sich Groupware einer Definition. Trotzdem hat diese Softwarekategorie die Aufmerksamkeit und Phantasie der Information Technologie (IT) Manager, der Reihe der Unternehmensmanager und Endbenutzer, nicht zu vergessen die Softwarelieferanten, auf sich gezogen. Obgleich die meisten Unternehmen keine klare Definition von Groupware entwickelt haben, ist ihnen voll bewußt, daß der Schlüssel zum Erfolg und zum Überleben in der Erweiterung des Wissens der Beschäftigten und der Handelspartner liegt. Außerdem wissen die Unternehmen, daß ein klarer Wettbewerbsvorteil bei denen liegt, die ihre intellektuellen Anlagen effektiv verwalten und nutzen.

Die meisten Definitionen über Groupware neigen zu einer Fokussierung auf einzelne Technologien mit relativ engen Designzentren. Es überrascht nicht, daß Lieferanten von Produkten, die mit Kommunikation zu tun haben, - - das "Pushen" von Information in einer Organisation -- Messaging als die Haupttechnologie für Groupware betrachten. Ebenso tendieren Lieferanten von Produkten, für die Kollaboration im Mittelpunkt steht - - Teilen von Informationen und gemeinsames "Verständnis schaffen" - - dahin, computerconferencing und gemeinsame Datenbanken als Kern von Groupware zu betrachten. Diejenigen Lieferanten, die darauf ausgerichtet sind, Individuen und Gruppen bei der Koordination komplexer Aufgaben zu unterstützen, welche ein reichhaltiges Mix an Delegation, sequentiellen Abmeldungen, etc. beinhalten, neigen dazu, Anwendungsentwicklungstools, die Aufgaben und Workflowautomation unterstützen, als das non plus ultra von Groupware zu betrachten. Dieses ist so, weil Groupware sich an die Technologien annähert, die früher als unabhängige Technologien betrachtet wurden ( Messaging, Conferencing, Workflow, etc. ), so daß hier viel Konfusion über ihre Definition und ihren Anwendungsbereich herrscht.

Es scheint nahe zu liegen, wenn wir von der Annahme ausgehen, daß Groupware Individuen helfen soll, in einer qualitativ besseren Art zusammenzuarbeiten, daß Groupware eine Integration dieser Technologien repräsentiert. Dieses Buch setzt einfache Rahmenbedingungen für Gruppenarbeit, die auf drei Kategorien basiert:

- **Kommunikation** - eine reichhaltige, multimediale elektronische Nachrichtenübermittlung (Messaging);

- **Kollaboration** - Erleichtern eines multimedialen, gemeinsamen, virtuellen Arbeitsraumes;

- **Koordination** - Hinzufügen der Struktur von Unternehmensprozessen zur Kommunikation und Kollaboration, so wie zur Implementierung einer Unternehmenspolitik.

Durch eine nähere Untersuchung werden wir die Bedingungen bestimmen, nach denen jedes Technologiemoell aufgeschlüsselt wird, wenn es von sich selbst genutzt wird. Daraus lernen wir, daß Gruppenapplikationen reichhaltige Kombinationen von Technologien erfordern. Was außerdem eine Groupware-Plattform so leistungsfähig macht, ist ihre Fähigkeit, dynamische Verschiebungen zwischen und durch diese drei Arten von Gruppenarbeit zu unterstützen: Kommunikation, Kollaboration und Koordination.

Deshalb ist Groupware nicht einfach eine Aufreihung von Merkmalen und Funktionalitäten, sondern statt dessen eine Plattform, die einfach und elegant diese Konvergenz widerspiegelt. Eine Groupware-Plattform wird deshalb durch die Integration dieser drei primären Technologien repräsentiert:

- Ein **Objektstore (Datenbank)**, in welcher Unternehmenswissen -- Nachrichten, Dokumente, Formen, Memos, Reporte -- gespeichert und verwaltet werden können.

- Ein **Verteilungs- und Zugriffsmodell**, das Usern erlaubt, einfach Informationen zu lokalisieren und verbreiten.

- **Rahmenbedingungen für Anwendungsentwicklungen**, die die systemeigenen zugrundeliegenden Dienste eines Objektstores und Verteilungs-/ Zugriffsmodells erweitern.

Natürlich muß eine Groupware-Infrastruktur die generellen Anforderungen an Workgroupumgebungen beachten. Diese beinhalten speziell:

- **Integration externer Ressourcen.** Der Gesichtspunkt der Herkunft für Arbeitsgruppeninformation ist oft extern zu der Groupwareumgebung ( d.h. Desktopproduktivitätstools, relationale Datenbanken, etc. ).

- **Plattformunabhängigkeit.** Groupwareanwendungen werden anfangs oft in bestimmten Abteilungen eingeführt und finden schließlich unternehmensweiten Einsatz. Plattformunabhängigkeit ist entscheidend, um universellen Gebrauch und Investitionsschutz zu gewährleisten.

- **Mobilität.** Eine Groupware-Infrastruktur muß fähig sein, viele geographisch zerstreute Anlagen, einschließlich Heim-, Laptop- und Notebookcomputer zu unterstützen.

- **Überbetriebliche Unternehmensanwendungen.** Da Unternehmen ab jetzt auf Kunden und Handelspartner als essentielle Teilnehmer in der Automation von Geschäftsprozessen angewiesen sind, ist die Fähigkeit zur nahtlosen Erweiterung der Anwendung -- von Beginn an oder hinzugefügt als Einfall -- ein wichtiger Teil einer Groupware-Infrastruktur.

Keine Anwendung für Unternehmensprozesse kann so geschrieben werden, daß vollständig jede Situation antizipiert werden kann. Egal wie viele Ausnahmen und

Spezialfälle erklärt werden, die Menschen werden neue Bedürfnisse entdecken, da sie die Tiefen einer Anwendung erforschen und sich ständig neue Geschäftssituationen ergeben. Wir fassen also zusammen, daß jedes System zur Schaffung, Verwaltung und Erweiterung von Unternehmenswissen die Anforderungen von Unternehmen einhalten muß und deshalb diese Kriterien erfüllen muß:

- Es muß die volle Breite von Klienten-, Netzwerk- und Serverbetriebssystemen unterstützen.
- Es muß mobile und entfernte Arbeiter unterstützen.
- Es muß nahtlose überbetriebliche Unternehmensaktivitäten unterstützen.

Ein Groupware-System, das architektonisch korrekt im Sinne der Konvergenz von Kommunikation, Kollaboration und Koordination ist, ist trotzdem auf Unternehmensebene zum Scheitern verurteilt, falls es sich nicht auch mit den pragmatischen Realitäten der nomadischen Arbeiter und der überbetrieblichen Unternehmenskommunikation befaßt.

## **Einführung**

Wissen ist der einzige dauerhafte Aktivposten einer Organisation. Die Fähigkeit, dieses Wissen ausbeuten und zu verwalten zu können, ist entscheidend für das Überleben und den Erfolg einer Organisation, unabhängig von der Unternehmensgröße.

Eine Softwaregattung, Groupware genannt, gewinnt als wichtige Technologie hierfür zunehmend an Bedeutung. Sie ermöglicht es Firmen, gemeinsames Wissen zu schaffen, zu teilen und weiterzugeben

Die Implementierung einer Groupware Infrastruktur ist die Herausforderung, vor der die Information Technology (IT) Manager stehen, um dem Benutzer das benötigte breite Spektrum an Groupwarediensten zu bieten.

Tatsächlich hängt die erfolgreiche Einführung von Groupware zum einen von der Fähigkeit zur Implementierung dieser Technologien und zum anderen der Notwendigkeit, dem IT Manager die Möglichkeit einzuräumen, die dafür notwendige organisatorische und systeminterne Infrastruktur im Unternehmen schaffen zu können.

Erfolgreiche IT Organisationen werden diese Infrastruktur so entwickeln, daß sie den Anforderungen der Endbenutzer möglichst weit entgegenkommt.

Der Begriff "Groupware" wird häufig benutzt, wenig verstanden, und ist häufig die Quelle von Verwirrung und Skepsis. Dieser bedauerliche Tatbestand ist das Anzeichen für zwei zugrundeliegenden (underlying) Tendenzen. Zum einen tendieren Lieferanten, Benutzer und Beobachter naturgemäß dazu, Groupware als Summe seiner Anwendungen zu sehen, ohne Berücksichtigung der optimalen Technologie und Infrastruktur.

Daher ist es schwierig, ein folgerichtiges Verständnis von Groupwaretechnologie

durch seine grundverschiedenen Anwendungsmöglichkeiten wie electronic mail (e-mail), group calendaring/scheduling, forms routing, bulletin boards, Arbeitsflußautomatisierung, Computer-, Videoconferencing, und vielen weiteren herzuleiten.

In vielen Fällen sind ursprünglich nicht als Groupware konzipierte Anwendungen über ihren ursprünglichen Anwendungsbereich hinaus so verändert und erweitert worden, damit sie den Anforderungen des Betriebes genügen.

In Wahrheit unterstützt eine vollständige Groupware-Infrastruktur nicht nur diese drei Arten von Gruppenarbeit, sie produziert Synergieeffekte zwischen ihnen, damit eine Gesamtheit schaffend, die mehr als die Summe der einzelnen Bestandteile ist.

Dieses Papier untersucht diese drei grundlegenden Dimensionen von Groupware - Kommunikation, Kollaboration und Koordination. Es diskutiert die Basistechnologien, die sie unterstützen - messaging, shared databases und Arbeitsflußautomatisierung (workflow automation) - und untersucht die Bedingungen, unter denen das System zusammenbricht, wenn es über seine ursprünglich vorgesehenen Anwendungsgebiete hinaus überfordert wird.

Dieses Papier beschreibt, wie eine komplette Groupware-Infrastruktur eine Verbindung zwischen diesen sonst verschiedenen Technologien schafft.

Debatten über Infrastruktur können manchmal dogmatisch werden und neigen häufig dazu, architektonische Reinheit unter dem Risiko zu produzieren, die tatsächlichen realen Gegebenheiten zu ignorieren; wie Individuen, Gruppen und Organisationen diese Anwendungen nutzen, um Geschäftswissen zu generieren, verwalten und weiterzugeben.

Die noch kurze Vergangenheit der Groupware Anwendungen hat gezeigt, daß die Benutzer selbst die Technologie über die ursprünglich vorgesehenen und konzipierten Anwendungsbereiche hinaus ausdehnen, damit alle ihre unterschiedlichen Bedürfnisse befriedigt werden können.

In diesem Papier bemühen wir uns, einen Ausgleich zwischen den architektonischen Anforderungen und praktischen Aussagen, d.h. wie Gruppen in der Realität wirklich arbeiten, zu finden.

Die Tatsache, daß die Endbenutzer selbst Technologie in unerlaubter Art und Weise ausdehnen, ist eines der wichtigsten Elemente einer Groupware-Infrastruktur: es muß flexibel genug sein, um auf die spezifischen Bedürfnisse der Benutzer einzugehen, um nicht auf Lösungen vertrauen zu müssen, die von anderen definiert wurden.

Es ist augenscheinlich, daß natürliche Synergieeffekte zwischen Kommunikation, Collaboration und Koordination in der Entwicklung von Gruppen und der sie unterstützenden Technologien entstehen.

Besonders die Technologien des Messaging und der gemeinsam genutzten Datenbanken (shared databases) bilden die fundamentale Grundlage für diese drei Arten von Gruppenarbeit und die Integration in ein gemeinsames System. Zugänglich durch Werkzeuge der Anwendungsentwicklung und eine dem Endbenutzer entsprechende Architektur der gemeinsam genutzten Datenbanken werden

Gruppenmitglieder mit der Fähigkeit zur Gestaltung, Verwaltung und Pflege von Gruppenprozessen ausgestattet. Ebenso bietet Messaging ein allgegenwärtiges store-and-forward Transportsystem und Kommunikationsmittel zwischen Menschen und Anwendungen. Diese duale Architektur ist entscheidend für die Flexibilität und Gebräuchlichkeit der Umgebung.

Dieses Papier richtet sich außerdem an praktische Fragen der Gruppenarbeit: das Bedürfnis nach Unterstützung entfernter und mobiler Gruppenmitglieder, heterogene Client-/Server Systeme und Netzwerkumgebungen, genauso wie den wachsenden Trend hin zu überbetrieblichen Systemen zwischen Kunden und Handelspartnern.

Es ist wichtig, zu bemerken, daß dieses Papier nicht den Anspruch erhebt, ein endgültiges Bild über Groupware Anwendungen abgeben zu können. Die Dynamik und die Entfaltungsmöglichkeiten hat dazu beigetragen, daß ein wenig Konsens bezüglich der Definition von Groupware besteht und es gibt keinen Grund, anzunehmen, daß Veränderungen in der Nutzung von Groupware sich bald verlangsamen werden.

Auch wenn Benutzer weiterhin Groupware auf unvorhergesehene und innovative Art und Weise nutzen werden, wird doch der architektonische Unterbau - Schwerpunkt dieses Papiers - in vorhersehbare Zeit voraussichtlich gleich bleiben. Dieses Papier schließt mit architektonischen Richtlinien für den Aufbau einer Groupware Infrastruktur.

## **Groupware definieren**

Frage zehn Leute - CIO's, IT Manager, Enduser und Softwareverkäufer -- nach der Definition für Groupware, und Du wirst zweifellos zehn verschiedene Antworten erhalten. Die einzig gemeinsame Basis, die die Befragten gemeinsam haben werden, ist, daß sie nicht genau wissen, wo Groupware beginnt und endet.

Definitionen von Groupware, die weiter gehen als "Software, die Gruppenarbeit unterstützt", konzentrieren sich gewöhnlich auf einen einzigen Aspekt von Gruppenarbeit und eine Hauptanwendung, die diese Art von Arbeit unterstützt. Die Erfahrung der meisten Benutzer mit Groupware ist gering gewesen: sie haben eine oder einige Groupwareanwendungen genutzt, aber sehen keinen Kontext, in dem man sie einordnen könnte. Deshalb sind User von E-Mail und mailunterstützten Anwendungen geneigt, Groupware durch die Perspektive des elektronischen Messaging zu betrachten. Benutzer von Formularen, die Produkte weiterschicken, sind natürlich geneigt, Groupware als eine Funktion von Arbeitsflußautomatisierung zu betrachten. Benutzer von elektronischen Konferenzsystemen oder dem World Wide Web neigen dazu, den gemeinsamen Zugriff auf Informationen als die Wurzel von Groupware zu betrachten.

Diese unterschiedlichen Ansichten entstehen, weil Groupware in der Tat ihren Ursprung in drei verschiedenen, aber zunehmend überschneidenden Anwendungsgebieten hat: elektronisches Messaging, Informationsmanagement und Workflow- / Prozeßautomatisierung. Jede dieser drei Technologiedomänen hat zu einer Zunahme populärer Groupware-Anwendungen geführt: E-Mail, elektronische Konferenzsysteme, bulletin boards, Routing und Tracking. Die profilierteste dieser

Anwendungen ist das elektronische Messaging; aus diesem Grund wird es oft als der Eckpfeiler von Groupware betrachtet.

Treten wir jedoch von diesen kurzsichtigen Betrachtungen der verrichteten Aufgaben innerhalb des Arbeitsplatzes zurück und nehmen eine breitere Sichtweise an, wie Leute wirklich arbeiten. So können wir Menschen finden, die sich von einer Arbeitssituation zu einer anderen begeben, mit wechselnden Arbeitsmethoden, mit wechselnder Arbeitsgruppenszusammensetzung, wechselnden Mengen von nur locker miteinander in Beziehung stehenden Aufgaben. Das alles erfordert Flexibilität während des Arbeitstages, und einige Aufgaben davon werden -- oder sogar ein erkennbar stabiler Zustand -- am Ende des Tages oder am Ende der Woche zur Vollendung gebracht. Wir werden Leute finden, die z.B. das Telefon nehmen, um eine schnell zu klärende Frage zu stellen -- und oft erkennt man, daß die Frage unbeantwortet bleibt, wenn z.B. der Anruf nicht beendet werden kann. Wir entdecken Sonderkommissionen, die gebildet und wieder aufgelöst werden -- neue Gruppen treffen sich in Konferenzräumen, um zu bestimmen, wie sie zusammenarbeiten werden, und etabliertere Gruppen, die kaum ihre Sätze beenden müssen, um ihre Ideen mit Phrasen und Gesten zu übermitteln.

Wenn wir Gruppen in diesem Licht betrachten, ist es klar, daß Groupware nicht als eine einzelne Technologie oder eine Kollektion von Anwendungen definiert werden kann. Gruppenbedürfnisse ändern sich mit der Zeit, und jede Gruppe unterscheidet sich von anderen. Damit ein Groupwaresystem effektiv ist, muß es imstande sein, alle Arten von Gruppenarbeit zu unterstützen. Speziell:

- Elektronisches Messaging ist ein effektives Tool für eine Mitteilungsübermittlung und stimmt deutlich mit den meisten Kommunikationsbedürfnissen überein. Messaging ist eine so leistungsfähige Lösung, daß Benutzer begannen, Messagestores als dauerhafte Nachrichtenspeicher einzusetzen. Weiterhin haben Benutzer ihre Funktionalität durch das Nutzen von Mailinglisten erweitert, um Gruppenmeetings zu unterstützen oder die Miterstellung und Revision von gemeinsamen Dokumenten.

- Jedoch werden bulletin boards und Computerkonferenzsysteme ( gemeinsame Datenbanken ) genutzt, um eine zusammenhängende gemeinsame Sichtweise von Gruppeninteraktionen zu schaffen. Diese Groupwareanwendungen sind mittlerweile in vielen Organisationen Alltag geworden.

- Systeme, in der die meisten Informationen für Gruppen in Datenbanken gespeichert sind, leiden unter der von Natur aus passiven Beschaffenheit konventioneller Datenbanktechnologien. Jeder Teilnehmer in einer Gruppe hat Verantwortung für das Auffinden von Informationen und das Disponieren seiner Maßnahmen zu tragen, hierzu ist das Wissen über Veränderungen von Informationen notwendig. Deshalb ist der Bedarf an einer koordinierten Nutzung von Messaging zur Mitteilungsübermittlung und Datenbanktechnologie gemeinsamer Information entstanden. Die Integration von Messaging und gemeinsamen Datenbanken in einer einzigen Lösung ist als Resultat entstanden.

- Ein Arbeitsflußsystem, das einfach auf Formularen basiert, die von Person zu Person geschickt werden, läßt wieder jede Person in der Gruppe auf sich allein gestellt, ohne Überblick über den Prozeß. Dagegen kann ein Arbeitsflußsystem, das auf einer Datenbank basiert, das die Schlüsselmerkmale und ihren Status speichert, den vollen

Zusammenhang für jeden verschaffen. Hier repräsentiert Groupware die Konvergenz von Messaging, Informationsmanagement und Arbeitsflußautomatisierung, indem es die entstehenden Bedürfnisse von Gruppen zur Schaffung, Teilung und Erweiterung von Unternehmenswissen deckt.

Betrachtet man Groupware von einer mehr organischen Perspektive, wird deutlich, daß Information und Wissen bei der Unterstützung von drei primären Funktionen geteilt werden: Kommunikation, Kollaboration und Koordination. Es gibt zwei Dimensionen, die die Rolle charakterisieren, die die Technologie bei der Erleichterung von Gruppenarbeit spielt: den Grad der auferlegten Struktur dieser Technologie und die aktive / passive Rolle, die diese Technologie bei der Unterstützung von Gruppenarbeit spielt.

- Die erste Dimension beschäftigt sich mit dem wechselnden Grad an Struktur, die bei Gruppenarbeit erforderlich ist. Diese dürfte sich zwischen Situationen, in der Information *ad hoc* verteilt werden ( d.h. das Verschicken einer E-Mail an eine Gruppe ), und höher strukturierten Prozessen bewegen, in der Schritte vordefiniert und determiniert sind, wie das Schicken einer Bestellanforderung.

- Die zweite Dimension spricht die passive / aktive Beziehung zwischen dem Technologiemedium und dem Individuum oder der Arbeitsgruppe an. Passive Anwendungen lassen die Kontrolle in den Händen des Individuums oder der Arbeitsgruppe, während aktive Anwendungen eine aktivere oder direktive Rolle bei der Kontrolle des Gruppenarbeitsflusses spielen. Zum Beispiel ist ein gemeinsames Datenbanksystem passiv, das den Benutzern das Lenken eines Diskussionsfadens erlaubt, während ein System aktiv ist, wenn es einen Prozeß aktiv beobachtet und den User über ein Ereignis benachrichtigt.

Das Diagramm unten ist hilfreich, um zu verstehen, wie verschiedene Groupware-Anwendungen mit diesen Dimensionen korrelieren und wie ihre Beziehung untereinander ist.

Jedes Kommunikations-, Kollaborations- und Koordinationssystem hat seine eigenen, einzigartigen Charakteristiken. Den Designschwerpunkt jeden Bereichs zu verstehen, ist wichtig zur Erkennung der notwendigen Kriterien für Technologien und Anwendungen in jedem Bereich. Zusammengenommen formen diese Kriterien die Basis für die Bewertung einer umfassenden Groupware-Architektur -- nahtlose Unterstützung der Kommunikation, Kollaboration und Koordination zu jeder Zeit, an jedem Ort.

Unsere Definition von Groupware ist einfach diese -- Werkzeuge, die eine Zusammenarbeit von Menschen durch Kommunikation, Kollaboration und Koordination ermöglichen. Was eine Groupwareplattform so leistungsfähig macht, ist ihre Fähigkeit, die dynamischen Bewegungen zwischen Kommunikation, Kollaboration und Koordination zu unterstützen. Die folgenden Kapitel untersuchen diese Bereiche hinsichtlich ihrer Gruppenbedürfnisse und der Technologien, die sie unterstützen.

## **Kommunikation**

Kommunikation ist die Vermittlung von Wissen. In geschäftlichen Bereichen können Kollegen auf unterschiedlichste Art und Weise miteinander kommunizieren: durch förmliche Treffen und Präsentationen, durch innerbetriebliche Memos, durch das Telefon und in informellen Treffen unter Umgehung des Dienstweges.

Die Informationen und das Wissen, daß auf diese Art und Weise vermittelt wird, kann die Form sowohl verbaler (geschriebener und gesprochener) als auch sichtbarer (Bilder, Zeichnungen, Körpersprache) Kommunikation annehmen.

Die Rolle von Kommunikation ist die eines passiven Mediums zur Übermittlung von Informationen.

Variablen wie Zeit, Ort und Teilnehmerzahl bestimmen das am besten passende Kommunikationssystem in der gegebenen Situation. Gleiche Zeit, gleicher Ort, Eins-zu-Eins Kommunikation repräsentiert die einfachste Form der Kommunikation. Steigen die Kombinationen von Zeit, Ort und Teilnehmerzahlen, erhöht sich die Komplexität. Ob gut oder schlecht, es ist eine sehr komplexe Kombination dieser Dimensionen, zwischen denen Wissen geschaffen, geteilt und weitergegeben wird.

In zunehmendem Ausmaß vertrauen Individuen der elektronischen Post als Kommunikationsmittel innerhalb oder außerhalb des eigenen Unternehmens. Während elektronische Post die Effizienz und Genauigkeit von Kommunikation für einige gesteigert hat, stellt sie jedoch eine Herausforderung für andere dar, einschließlich den Benutzern und auch den Netzwerk- und Systemverwaltern.

## **Elektronisches Messaging: Eine Technologie für Kommunikation**

Elektronisches Messaging ist der store-and-forward Transport von elektronischen Objekten zwischen Menschen, zwischen Menschen und Anwendungen und zwischen Anwendungen. Das elektronische Messaging ist auf die asynchrone Übermittlung von Nachrichten von einem Ort zu einem anderen hin ausgerichtet. Nachrichten können entweder einfache oder komplexe Informationen enthalten, sie können zu spezifischen Individuen oder auch an Gruppen gesendet werden. Messaging unterstützt die Informationsverteilung zu unterschiedlichen Zeiten und unterschiedlichen Orten. Dies geschieht mit Hilfe seines store-and-forward- oder "Push"-, Modells der Übertragung oder Bewegung von Informationen. Informationen werden vom Sender zum Empfänger "gepusht".

Das store-and-forward Transportsystem von elektronischem Messaging unterscheidet sich von anderen Kommunikationstechnologien. store-and-forward Transport wird zum Bewegen oder "Pushen" eines Objektes von einem Punkt zu einem anderen entlang einer Anzahl von Zwischenpunkten ( d.h. von Postamt zu Postamt ) genutzt, bis es beim letztendlichen Empfänger ankommt. Messaging verschafft eine asynchrone Verbindung, weil Sender und Empfänger nicht zeitgleich anwesend sein müssen. Darin liegt der wahre Vorteil von store-and-forward Prozessen.

Dem Messaging wird die revolutionierende one-to-many Kommunikation zugeschrieben. Natürlich führt dies schnell zu many-to-many Kommunikation. Wie durch das "Netz" von Punkt-zu-Punkt Pfaden in dem Diagramm unten dargestellt

wird, hat der Gebrauch von E-Mail zur many-to-many Kommunikation das Volumen von E-Mail exponentiell erhöht. Wie wir sehen werden, ist diese Entwicklung im Bereich der Kommunikation nicht so einfach eine Verschiebung von store-and-forward Messaging.

## Der Message Store

Ursprünglich ist der message store als vorübergehende Aufbewahrungsstelle für Nachrichten benutzt worden, um von dort zu ihrem endgültigen Ziel zu gelangen. Im Laufe der Zeit, als Messaging zunehmend als one-to-many und many-to-many Kommunikation genutzt wurde, entwickelte das message store de facto zu einem "Datencontainer" großer Mengen gemeinsam genutzter Informationen.

Daher wurde der message store, der für vorübergehende Daten, ad hoc Nachrichten, konstruiert wurde, auch für halb-permanente Daten sowie auch für Daten mit einem undefinierten Lebenszyklus genutzt

Da die ursprüngliche Designkonstruktion des message store auf zeitweilige Nachrichten hin ausgerichtet ist, wurde es auf die Lieferung und Abruf von Nachrichten hin optimiert.

Dieses System setzt voraus, daß die Dokumente und Objekte, die durch das e.mail System geleitet (routet) werden, woanders gespeichert und verwaltet werden. (z.B. auf der Festplatte eines Desktop-Computers oder auf einem LAN basierten Fileserver.) Das message store wurde nicht für die dauerhafte Speicherung und Verwaltung von Informationen konstruiert.

Trotzdem haben sich Systeme zur Nachrichtenübermittlung zu einem Friedhof für immer mehr gemeinsam genutzte Informationen entwickelt. Mit dem Ergebnis, daß Benutzer und Lieferanten Möglichkeiten erforscht haben, Informationen durch verschiedene Formen des Folders, ruled-based processing und der Anwendungsentwicklung besser zu verwalten, manipulieren und die Informationsverarbeitung automatisieren zu können.

Messaging APIs (Application Programming Interfaces) entstanden um das Nachrichtensystem für eine programmgesteuerte Steuerung der Objekte von einem Anwender zu einer Applikation, von einer Applikation zu einem Benutzer und zwischen den Anwendungen nutzen zu können. Beispiele hierfür sind mail-enabled Desktop Anwendungen mit integriertem elektronischem Postsystem, Terminkalender sowie Arbeitspläne für Gruppen und Dokumentenfluß Anwendungen.

Infolge des Informationsverarbeitungsproblems haben sich Messaging Systeme zu einem virtuellen Kommunikations "hub" entwickelt, die einfache Messaging Systeme nicht mehr bewältigen können.

Trotz dem enormen Erfolg des Messaging Systems als Kommunikationsmedium beginnen sich Anwender nun zu beschweren, daß diese Technologie zunehmend unhandlich wird.

Es gibt zwei Aspekte von Messaging, die eine Herausforderung an das geschäftliche Informationsmanagement darstellen.

Zum einen sind die Benutzer nicht in der Lage, ein immer höheres Nachrichtenvolumen bewältigen zu können.

Zum anderen wird außerdem der Mangel an hochentwickelten Tools zur Verwaltung der in den e-mails enthalten Informationen zu einem immer größer werdenden Problem.

E-mail ist das Opfer seines eigenen Erfolges. Die Bequemlichkeit der Nutzung und seine weitverbreiteten Anwendungsgebiete haben zu einer Explosion des e-mail Verkehrs geführt. Dieses führt zu Informationsverlusten für den Benutzer und der Anhäufung von großen Rückständen von unbeantworteten Nachrichten. Kostbare Zeit wird mit dem Lesen und Sichten von Nachrichtenbeständen verwendet, die nur eine minimale Bedeutung haben. Einige Benutzer wurden so durch die Flut der täglich eingehenden e-mail Nachrichten so überanstrengt, daß sie begannen, konsequent viele Nachrichten komplett zu ignorieren ! Diese unglückliche Situation hat negative Folgen sowohl für die individuelle als auch gemeinschaftliche Effizienz, durch verlorene Informationen und verringerten Kundenansprache entstehend, die letztlich die Gesamteffizienz mit in Leidenschaft ziehen.

Dieses Problem wird an Bedeutung zunehmen. Die zunehmende Kapazität der e-mail Systeme (d.h. die Fähigkeit, ein 50 MB großes File an 10 oder mehr Benutzer zu senden) wird das Problem (exacerbate).

Die unten gezeigte Graphik verdeutlicht das Verhältnis zwischen Nachrichtenvolumen und Produktivität.

Anfänglich ist eine positive Auswirkung auf die Produktivität zu verzeichnen, da die Menschen die Informationen, die sie benötigen, schneller bekommen. Doch wenn das Volumen und die Komplexität der Informationen steigt, erreichen Benutzer einen Punkt, ab dem sich die Produktivität wieder verringert (Point of diminishing returns).

Das zunehmende Nachrichtenvolumen kann möglicherweise negative Auswirkungen auf die individuelle Produktivität haben, vor allem aus dem Grund, daß die Informationen, die wir brauchen, mit anderen Informationen vermengt sind. Informationen, die andere uns zusenden, nur weil sie denken, daß wir sie brauchen, oder weil sie meinen, daß wir sie haben sollen.

Als Antwort wurden Features wie rules, Filter und hierarchische Folder eingebaut, die uns bei dem Umgang mit dem e-mail Management unterstützen. An diesem Punkt angelangt geht jede durch die schnellere Informationsaufnahme geschaffene Produktivitätssteigerung durch den Versuch verloren, die Informationen nach relevanten Informationen zu sortieren.

## **Produktivität und Information Overload**

Das Phänomen der Informationsüberflutung und seine Auswirkungen auf die Produktivität wurde in der Wissenschaft schon seit einiger Zeit erkannt. Um diesen Zusammenhang besser verstehen zu können, ist es wichtig, zwischen den Vorteilen zu unterscheiden, die zum einen dem Absender aber auch dem Empfänger von e-mails entstehen.

Als Absender beschleunigt e-mail unsere Möglichkeiten, Informationen verteilen zu können. Als Empfänger profitieren wir davon, Nachrichten schneller bekommen zu

können.

Wenn die Rate, mit denen wir Informationen erhalten, unsere Fähigkeit zur Verarbeitung und Verwaltung dieser Informationen übersteigt, erreichen wir den Punkt der abnehmenden Produktivität.

In anderen Worten, e-mail löst das Verteilungsproblem für den Absender sehr effektiv, verursacht aber einen Informationsdschungel bei dem Empfänger.

Betrachten wir ein Beispiel. Ein Manager sendet eine e-mail Nachricht an ein Dutzend Menschen, die um Antwort für ein Projekt bitten. Wenn alle Empfänger alle ihre Antworten an alle Originalempfänger zurücksenden, übersteigt die Zahl von Nachrichten bei weitem die Fähigkeit eines einzelnen Empfängers, den Überblick über die erhaltenen Nachrichten zu behalten und damit sich durch diese Nachrichten wühlen zu können.

Darüber hinaus gibt es keinen Mechanismus, der dem Empfänger mitteilt, welche Antwort sich auf welche Nachricht bezieht. Je höher der Grad an Wechselwirkung, je größer die Teilnehmerzahl ist, je schwieriger wird es, zu wissen, wer was als Antwort zu wem gesagt hat.

Dieses zeigt die Informationsüberflutung und den Verlust den Kontexts, in dem sich die Information selbst befindet.

Um zur Wurzel dieses Problems zu gelangen, ist eine nähere Betrachtung der Anforderungen der many-to-many Kommunikation erforderlich sowie die Betrachtung eines Modells zur Informationsverwaltung notwendig.

Viele der Probleme, die mit dem überbelasteten Messaging System in Verbindung gebracht werden, können auf ihre ursprüngliche Charakteristik zurückgeführt werden: Unterstützung unstrukturierter Kommunikation durch ein "Push" Modell. Aus diesem Grund versagt dieses System bei der Unterstützung von Kommunikation mit einem höheren Grad an Komplexität, während es unstrukturierte Informationen effektiv unterstützen kann.

Im Gegensatz dazu benutzen Technologien, die für die Unterstützung komplexer many-to-many Kommunikation konzipiert wurden, ein vollkommen anderes Modell der Informationsverteilung und -verarbeitung.

## **Zusammenfassung**

Bis jetzt haben wir den Bereich der Kommunikation isoliert betrachtet, um ihre Beziehung zu anderen Aspekten der Groupware-Technologie besser zu verstehen. Wir haben festgestellt, daß e-mail ein effektives Medium für one-to-one und one-to-many Formen der Kommunikation ist. Durch den Mangel an Struktur und die Höhe des gesendeten Informationsvolumens, wird deutlich, daß eine many-to-many Kommunikation in dieser Umgebung schnell unverwaltbar wird.

Um eine angemessene Technologie zur Lösung dieser Probleme zu bestimmen, ist es wichtig, zwischen Informationslieferung und -management zu unterscheiden. Als store-and-forward Transport ist Messaging effektiv für Informationslieferung. Um das Problem des Informationsmanagement zu lösen, müssen wir andere Technologien betrachten.

## **Kollaboration**

Kollaboration basiert auf einem gemeinsam genutzten Raum (shared space), dieses kann ein Raum, eine Tafel, ein napkin ? (Übersetzt: Windel oder Serviettenring), oder ein gemeinsam genutzte Online - Datenumgebung sein. Shared space dient als Prüfstein für den Vorgang der Kollaboration und es ist ein unentbehrliches Medium zur Lenkung der vielschichtigen menschlichen Interaktionen. Tatsächlich sind diese shared spaces die kollaborativen Tools, die den Kontext erbringen, in dem die Gesamtheit der Beziehungen mehr als die Summe der Fach- bzw. Sachkenntnisse der einzelnen Teilnehmer ist.

Kollaboration kann zwischen zwei Menschen stattfinden, oder kann die Form eines Many-to-Many Informationsaustausches annehmen. Aktivitäten wie (gemeinsame) Problemlösungen, Brainstorming und Identifizierung und Lokalisierung von Daten, die von anderen erzeugt wurden, sind mögliche Formen von Kollaboration.

Wie in der Kommunikation ist eine der bedeutendsten Beiträge dieser Technologie zum Umfeld der Kollaboration die Eliminierung der Begrenzungen von Zeit und Raum. Persönliche Treffen sind dann gebräuchlich, wenn die Gruppenmitglieder in der Lage sind, gemeinsam Zeit und Ort zu teilen. Telefonieren hat viele der Barrieren von Raum gesprengt und voice mail hat ebenso die Grenzen von Zeit aufgehoben. Die Erfahrung zeigt, daß Gruppen, die kollaborative Technologien in ihre Umgebung implementiert haben, in der Lage sind, die Anzahl der persönlichen Treffen effektiv zu minimieren, die andernfalls notwendig gewesen wären, Informationen und Ideen auszutauschen.

Dieses erlaubt Gruppen, persönliche Treffen so optimieren, daß sie auf die Treffen begrenzt werden, bei denen diese Art von Interaktion am sinnvollsten ist - um Entscheidungen zu erzielen, einen Konsens zu erreichen, usw.

### **Kollaboration über Variablen der Zeit und des Ortes**

Aufgrund seiner Komplexität und seines Potentials, Arbeitsmethoden zu verbessern wird der Rest dieses Kapitels näher auf den Aspekt der Asynchronität der Kollaboration eingehen. Asynchrone Kollaboration (unterschiedliche Zeit und unterschiedlicher Ort) erlaubt es Menschen, am gleichen Ort und Zeit effizienter zu arbeiten.

Unser wichtigstes Ziel ist die Definition der Rolle, die die Technologie der gemeinsam genutzten Datenbanken und Informationen in dem Umfeld der Kollaboration spielt.

Beginnend damit, wie diese Technologie die Grenzen von e-mail für die many-to-many Interaktion anspricht um dann auf andere Formen von Kollaboration überzugehen.

## **Gemeinsame Datenbanken: Eine Technologie für Kollaboration**

In dem vorangegangenen Kapitel haben wir festgestellt, daß Messaging ein nützliches, allgemein anwendbares Kommunikationsmedium ist, vielseitig anwendbar durch die Möglichkeiten der Mailinglisten in einigen Gruppensituationen, aber es ist nicht

wirklich auf die Bedürfnisse der many-to-many Interaktion eingestellt. Aber die generelle Verfügbarkeit und die asynchrone Natur von Messaging hat zu seinem Gebrauch als Medium für Kollaboration geführt. Während dies als eine logische Erweiterung von Messaging erscheinen könnte, unterschreitet das Modell dies in vielerlei Hinsicht.

Messagingssysteme sind hauptsächlich damit beschäftigt, Nachrichten zwischen Sender und Empfänger in Dateiformat abzulegen. Dies macht es für Benutzer schwierig, Informationen auf ein Diskussionsschwerpunkt hin bezogen zu finden. Es ist durch die Probleme des Tracking (Zurückverfolgen) (wer antwortet auf wen und was?) schwierig, den Kontext des Diskussionsfadens, der über eine Serie von E-Mails stattfindet, aufrechtzuerhalten.

In vielerlei Hinsicht hat dies die Erweiterung von E-Mailssystemen zur Unterstützung eines gemeinsamen On-Line Diskussionsraumes hervorgerufen, in der Struktur in Form von Rollen, Zugriffskontrolle und interaktiver Struktur, über die der Benutzer verfügen kann, eingeführt wird.

Gemeinsame Datenbanktechnologie hat sich zu einer insgesamt unterschiedlichen Domäne von Messaging entwickelt, ihrem kommunikationszentrierten Verwandten. In der Tat können gemeinsame Datenbanken zu den ersten Timesharingsystemen zurückverfolgt werden, die auf frühen Großrechnern gefunden wurden. Gemeinsame Datenbanken erleichtern kollaborative Interaktion durch Schaffung eines gemeinsamen virtuellen Arbeitsbereiches mit einer gruppenzentrierten Verbindung, die es Teilnehmern erlaubt, Informationen und Ideen zu teilen. Im Gegensatz zu Messagingsystemen, die ein Sendemodell oder *Pushmodell* nutzen, unterstützen gemeinsame Datenbanktechnologien ein *Pullmodell* zur Informationsteilung. Das Pullmodell erlaubt den Benutzer den gezielten Abruf der benötigten Informationen. Dieses erleichtert die Teilnahme an verschiedenen Gruppendiskussionen. User werden nicht mehr durch ein E-Mail Raster eingeeengt, sondern können vielmehr aktiv die Verantwortung übernehmen, Informationen nach eigenem Ermessen abzurufen ( oder auch zu ignorieren ).

Gemeinsame Datenbanken unterscheiden sich auch von Messagingsystemen darin, daß sie nicht nur eine vollständige Menge an Nachrichten, sondern auch Diskussionselemente, Nebendokumente, Argumente -- das ist Wissen -- an einer Stelle speichern, sichtbar durch eine gemeinsame Struktur, und sie erstellen ein konsistentes Protokoll von dem, was passierte. Dieses erleichtert ein gemeinsames Verständnis und ist wesentlich für die einzelnen Beteiligten, kollektiv Hauptkonzepte und Problemstellungen zu erfassen. Außerdem sind in einer Situation, in der eine gemeinsame Sichtweise geschaffen wurde, multiple Formen der Präsentation wichtig, weil Individuen private, angepaßte Ansichten haben wollen, um ihre eigenen, speziellen Aufgaben zu unterstützen. Diese vielfältigen Darstellungen können das Bedürfnis eines einzigen Teilnehmers befriedigen, Informationen datumbezogen, autorbezogen, dokumenttypbezogen, u.s.w. zu betrachten. Sie unterstützen auch das Bedürfnis, Informationen verschiedenen Gruppen zu präsentieren, zum Beispiel gegliedert nach Kundennamen oder nach Marktsegment für die Marketinggruppe oder geordnet nach Teilenummer oder Produktname für die Technikgruppe. Jede Form stellt eine andere Perspektive dar, durch die die kollaborative Aufgabe betrachtet wird, während Informationen in einen verschiedenen Kontext eingeordnet werden. Auch weil die Arbeit sich von Zeit zu Zeit ändert, ist die Verfügbarkeit von einfach zu bedienenden Designtools für den Endbenutzer wichtig. Sie versorgen einzelne

Beteiligte mit der Möglichkeit, Informationen anzupassen und zu modifizieren; andernfalls wird das System nicht benutzt werden und somit scheitern.

Die Verfügbarkeit von Werkzeugen und die Leichtigkeit, mit der diese Anwendungen entwickelt oder angepaßt werden können, ist ein Schlüssel zum Langzeiterfolg von Systemimplementationen.

Wichtige Attribute für diesen Erfolg ist in erster Linie die Möglichkeit, Benutzerschnittstellen anzupassen, die Granularität, mit der Daten und Informationen betrachtet und manipuliert werden können, als auch die Möglichkeit, viele Variablen zu beschreiben, die mit diesem System assoziiert werden. Kommerzielle Produkte, die eine spezifische Anwendung liefern (z.B. eine Gruppenkonferenz), und keinen Zugriff auf die zugrundeliegende Datenbankplattform verschaffen, sind in dieser Hinsicht beschränkt. Während out-of-the-box Systeme wie diese einen spezifischen, kurzfristigen Bedarf ausfüllen könnten, ermangelt es ihnen an Flexibilität und Anpassungsfähigkeit, die zur Lösung einer weiten Auswahl kollaborativer Probleme benötigt werden.

Produkte, die die zugrundeliegende Datenbankplattform einbeziehen und anpassungsfähige Tools und Anwendungen anbieten, verschaffen viel mehr Flexibilität. Um das zu illustrieren, nehmen wir das obere Beispiel noch einmal. Die Marketinggruppe könnte erkennen, daß es jetzt Kundeninformationen geordnet nach geographischen Regionen analysieren muß, in Ergänzung zum Marktsegment. Um dies zu erfassen, muß ein neues Feld zur Erfassung von Informationen über die Region zu der existierenden Datenbank hinzugefügt werden. Nur ein kollaboratives Tool mit Zugriff zu einer zugrundeliegenden Datenbasis, die es erlaubt, die Informationen in einem Dokument auf dem Feldlevel zu erfassen, kann modifiziert werden, um solche spezifischen ( und wechselnden ) Gruppenanforderungen einfach zu unterstützen.

## **Kollaborative Anwendungen**

Bei der Entwicklung von Anforderungen an eine flexible und gebräuchliche Umgebung für Kollaboration, ist es wichtig, zwischen der Technologie und den Anwendungen, die durch diese Technologie eingeführt wurden, zu unterscheiden. Eine flexible, gemeinsam genutzte Datenbank ist der Kern eines Groupware-Systems, die eine Plattform zur Verfügung stellt, auf der eine breite Palette von Anwendungsprogrammen entwickelt werden können.

Dieses kann von einer einfachen Diskussionsdatenbank bis hin zu einer vielschichtigen Wissensdatenbank reichen, die z.B. Kundenunterstützung oder Teamantworten auf Fragen bietet.

### **Elektronische Konferenzsysteme**

Elektronische Konferenzsysteme (Diskussionsdatenbanken, öffentliche Foren) erleichtern die asynchrone

Kollaboration durch ein Maß an Struktur, die den Prozeß der gemeinsamen Nutzung, Verwaltung und Steuerung von Informationen passiv erleichtert. Es wird ein interaktiver elektronischen Raum geschaffen, der als gemeinsamer Raum für Beiträge dient.

Solche Raume werden z.B. für Problemlösungsarbeit benutzt, das solche

grundsätzlichen Anforderungen des Brainstorming umfaßt, um Ideen generieren, strukturieren und bewerten zu können.

Elektronische Konferenzsysteme nutzen die Technologie der gemeinsam genutzten Datenbanken, um eine Teilnahme unabhängig von Zeit und Raum zu ermöglichen. Sie stellt die Struktur sicher, die notwendig ist, diese o.g. Schritte zu erleichtern.

In elektronischen Konferenzsystemen werden Nachrichten im Gegensatz zu den individuellen Mailboxen des Messaging Systems in einer gemeinsam genutzten Datenbank abgelegt

Alle Teilnehmer können neue Nachrichten erkennen und entsprechend reagieren. Ein Moderator kann eingreifen und neue Diskussionsthemen beginnen, um eine Gruppe von einer Stufe des Prozesses zur nächsten Stufe zu lenken.

### **Erweiterung gemeinsamer Datenbanken für andere Anwendungen**

Eine strukturierte Datenbank kann genutzt werden, um ein gemeinsames Verständnis einer Diskussion bereitzustellen. Auch andere Datentypen können in strukturierter Form eingelagert werden, um Gruppen die in einem speziellen Anwendungsgebiet arbeiten, zu unterstützen. Im Falle z.B. einer gemeinsamen Gruppenarbeit sollte eine Gruppe, die gemeinsam ein Dokument schreibt, einen gemeinsamen Entwurf des Dokuments teilen.

Um leichter Änderungen des jeweils anderen kommentieren und wiederum ändern zu können, kann das Dokument als Sammlung von Paragraphen, Kapiteln und Abschnitten gespeichert werden.

Diese können auf verschiedene Art und Weisen betrachtet werden: im linearen Modus, um wie ein gewöhnliches Verarbeitungsdokument auszusehen, in einem Übersichtsmodus, in einem "Revisions" Modus mit alten und neuen Versionen , usw. Jeder Paragraph kann als ein "Thema" für eine Diskussion dienen, und Gespräche können rund um die gegensätzlichen Streitpunkte des Papiers herum strukturiert werden; oder um einen beliebigen Satz von "Themennotizen", die typischerweise in einem on-line Forum gefunden werden.

Dieser Gebrauch von gemeinsamen Datenbanken erhöht die Nützlichkeit jeder Anwendung für Gruppen, die versuchen, Material zu entwickeln.

Gruppenarbeitsfähige Anwendungen wie z.B. Textprozessoren, Tabellenkalkulationen , die ihre Informationen besser in Datenbanken als in unstrukturierten Files speichern können, gewinnen an Bedeutung.

Wenn die Information erst einmal in einer gemeinsamen Datenbank enthalten ist, kann sie durch die ursprüngliche Anwendung oder durch eine übergeordnete Groupware Schnittstelle betrachtet werden, die Diskussionen, Debatten und Entscheidungsfindungen erleichtert.

## **Reference Publishing Systeme**

Refernce Publishing Systeme -- Systeme zur Veröffentlichung und Verbreitung von Dokumenten -- werden als Groupware betrachtet, weil sie eine Informationsteilung erleichtern.

Informationen werden elektronisch von Anbietern veröffentlicht und von vielen Konsumenten gelesen.

Für irgendein gegebenes Thema stellt dies eine one-to-many Übertragung von Informationen dar -- das akkumulierte Wissen von vergangenen Erfahrungen wird in Dokumenten erfaßt, wie Konkurrenzberichte, Vorhersagen und Rückblicke, Vorgehensweisen und manuelle Verfahrensweisen, Schulungsmaterialien, Rundschreiben und Zeitschriften.

Um ihre Rolle in Unternehmensumgebungen besser zu verstehen, ist es nützlich, Reference Publishing mit kollaborativen Anwendungen zu vergleichen und kontrastieren. Beide sind ähnlich in der Struktur - beide Systeme nutzen ein Pullmodell, um Anwendern das Navigieren und Blättern in großen Mengen von öffentlichen oder Unternehmensinformationen zu erlauben. Aber kollaborative Anwendungen sind interaktiv, während Reference Publishinganwendungen sich nur in eine Richtung erstrecken.

In der Tat ist das Reference Publishing typischerweise so einseitig ausgerichtet, daß es fraglich ist, ob man es kollaborative Anwendung oder eine Kommunikationsanwendung nennen soll. Es ist ein leistungsfähiges one-to-many Kommunikationstool -- mit einem schwachen Verteilungsmodell. Während sich typische Implementationen des Reference Publishing in eine Richtung erstrecken, kann die Unterstützung beidseitig gerichteter Interaktionen bedeutend die Brauchbarkeit von Reference Publishing Systemen erweitern.

Ein Reference Publishing System ist typischerweise eine verteilte Datenbasis ( oder zumindest ein verteiltes Dateiensystem ), die "beendete" Dokumente enthält. Wenn eine reichhaltig verteilte Datenbanktechnologie dem Publishing System zugrundeliegt, kann es in eine dynamische Umgebung transformiert werden, indem man einfach ergänzend integrierte Kommunikations- und Kollaborationsmöglichkeiten anbietet. Diese Umgebung kann das Lernen und Errichten einer gemeinsamen Vision unterstützen, vielmehr als einfach eine Informationssuch- und Informationsabrufumgebung für den persönlichen Gebrauch. Zum Beispiel könnte ein Team eine Diskussionsdatenbank nutzen, um im Anwendungsbereich über Richtung und Details einer neuen Produktstrategie zu kollaborieren. Das Resultat dieser Kollaboration ist ein Strategiedokument, das wiederum in einem Reference Publishing System gespeichert wird. Das Dokument ist dann für alle entsprechenden Zuhörer, externe oder interne, erreichbar. Leser ( z.B. Kunden, Finanz- und Industrieanalytiker ) des endgültigen Dokuments könnten ihre eigenen Antworten, Empfehlungen und Vorhersagen geben. Deshalb könnte das Unternehmen einen Link zwischen dem Strategiedokument in dem Reference Publishing System und einer Diskussionsdatenbank einfügen, zu dem alle Leser (auch die ursprünglichen Dokumentautoren ) etwas beitragen und antworten können. Außerdem können Teilnehmer der Diskussionsdatenbank Zeiger zu Dokumenten in anderen Reference Publishing Systemen als Teil ihres Kommentars einschließen.

## **Die passive Natur gemeinsamer Datenbanken**

Das Modell der gemeinsamen Datenbanken zur Speicherung und Pflege von On-Line Informationen bietet viele Vorteile gegenüber einem Modell, das auf dem Messaging

System basiert. Der wichtigste Vorteil ist die Möglichkeit, nur die Information zu ziehen, die vom Datenkonsumenten benötigt werden. Dieses ermöglicht eine bessere Kontrolle der erhaltenen und konsumierten Informationen; dieses gilt vor allem im Gegensatz zur Kontrolle über versandte Informationen bei Anwendung des Messaging Systems.

Aber auch diese Technologie leidet es auch unter gewissen Grenzen, wenn sie ohne einige Notifikationen oder Nachrichtenunterstützungen benutzt wird. Durch die passive Natur gemeinsamer Datenbanken hängt die Nutzung von den Konsumenten ab, die nach Informationen suchen. Gemeinsamen Datenbanken fehlt die Möglichkeit, Benutzern eine Hinzufügung oder Veränderung von Informationen zu melden. Betrachten wir z.B. eine Referenzdatenbank, die standardisierte Vorgehensweisen enthält: wenn Handlungsweisen in der Datenbank aktualisiert werden, ist eine Mitteilung über diese Änderungen wichtig.

Genauso wie eine Flut von e-mails unweigerlich zu einer Informationsüberlastung führt, kann eine Explosion von Dutzenden, Hunderten oder auch Tausenden von gemeinsamen Datenbanken den Benutzer eventuell mit einer überwältigenden und verwirrenden Umgebung konfrontieren. Ein ähnliches Phänomen tritt beim World Wide Web mit der Erscheinung von Tausenden neuer Homepage Seiten in jedem Monat auf.

Die Folge ist, daß große Datenmengen in Datenbanken gespeichert werden. Wichtige, relevante oder regelmäßig geänderte Informationen finden dadurch nicht den geeigneten Weg zum interessierten Benutzer.

## **Das World Wide Web als ein Kollaborationstool**

Das World Wide Web ( WWW oder Web ) hat sich rasant zu einem bedeutendem Netzwerkparadigma für inner- und überbetriebliche Veröffentlichungen und für andere kollaborative Anwendungen entwickelt. Grundsätzlich ist das Web eine Menge von Protokollen, die über das Internet betrieben werden ( als auch private, interne Netzwerke ). Diese Protokolle dienen als die Basis für eine client / server Umgebung, die Informationsteilung und seit kurzem Transaktionsverarbeitung und elektronischen Handel unterstützt.

Diese rapide Entwicklung hat dazu geführt, daß viele Organisationen das Web als die Basis eines viel breiteren Bereich von Anwendungen betrachten, von denen viele in die Kategorie von Groupware eingeordnet werden können, wie wir sie definiert haben, und ganz speziell in die Domäne der Kollaboration.

### **Die Webinfrastruktur**

Da sind drei wesentliche Technologien, die das World Wide Web heutzutage definieren, weil sie die Kommunikation zwischen einem Webserver und einem Webklienten verbunden über ein TCP / IP Netzwerk definieren.

- HTTP, *Hyper Text Transport Protocol*, regelt die Kommunikation zwischen einem Webserver und einem Webklienten.

- HTML, *Hypertext Markup Language*, ist das Dokumentformat für Webdokumente oder "Seiten".

Traditionelle Textverarbeitungsprogramme nutzen eine spezifische Methode, Dokumentattribute wie Schriftart, Zeichenabstand, u.s.w . darzustellen. HTML benutzt Direktiven, anstelle ein Format ausdrücklich anzugeben, es überläßt das eigentliche Formattieren dem Klienten. HTML ist eine Teilmenge von SGML ( Standard Generalized Markup Language ), die für die Formatierung von Hypertextverbindungen zwischen Dokumenten benutzt werden.

- URL, *Uniform Resource Locator*, ist ein Zeiger zu einer Ressource im Internet. Es dient als globales Adressierungsschema für Seiten, die auf Webservern gespeichert sind. Der URL enthält den Namen des Webservers ( z.B. WWW.Lotus.Com ) und mögliche Erweiterungen, die eine spezifische Webseite oder andere Informationen beschreiben, die vom Webserver genutzt werden.

Während viele Menschen heute das Web als ein Instrument betrachten, das auf Produkte fokussiert ist, wird sich das zweifelsohne schnell ändern, so daß wir das Web als eine Menge von Möglichkeiten sehen, die durch spezifische Protokolle implementiert werden. Diese werden durch einen breiten Bereich von Produkten unterstützt. Obwohl Webbrowser, so wie wir sie heute kennen, weiterhin genutzt werden, können wir fest erwarten, daß Webbrowserfunktionen direkt in andere Programme eingegliedert werden.

Ein "Webbrowser" wird jedes Programm sein, das Webklientenprotokolle implementiert, und ein "Webserver" wird jedes Programm sein, das Webserverprotokolle implementiert. Bald werden wir in einer Umgebung sein, in der Anwendungen wie Mailsysteme Webprotokolle nutzen, um Zugriff zu Datenbankservern zu haben. Dieses geschieht auch, um Webprotokolle zu implementieren. Daher werden die Protokolle festlegen und eingrenzen, was auf dem Web getan werden kann. Die Entwicklung der Webprotokolle ( speziell HTTP und HTML ) hat zwei Phasen durchlaufen und tritt jetzt in die dritte Phase ein.

**Phase Eins -- Informationsveröffentlichung.** Die erste Version von HTML unterstützte nur Informationsveröffentlichung durch ein Hypertextmodell. Ein Dokumentsegment konnte auf andere Dokumentsegmente in unterschiedlichen Dokumenten auf unterschiedlichen Servern verweisen, so daß User sich in einem "Informationsraum" in einer nichtlinearen Art bewegen konnten.

Jeder konnte Webseiten in HTML erstellen und diese auf einen HTTP Server speichern, und jeder mit einem Browser konnte HTTP nutzen, um Zugriff auf die Webseite zu haben. Webentwickler haben schnell erkannt, daß es Aufgaben gibt, die nicht mit den einfachen Webprotokollen erfüllt werden konnten, so daß ein "Trapdoor" oder Anwendungsausgang entwickelt wurde, genannt Gatewayanschluß oder CGI. Ein URL konnte, neben dem Verweis auf eine bestimmte Seite auf dem Webserver, auf ein Programm verweisen, das auf dem Webserver durch ein CGI aufgerufen wird. Ein CGI Programm kann dynamisch Webseiten erstellen. Dies eröffnet eine ganz neue Welt des Web, weil "dynamische Veröffentlichung" zusätzlich zum klassischen, elektronischen Veröffentlichen unterstützt werden konnte. Wir betrachten zwei einfache Beispiele, um den Unterschied zu sehen..

In dem ersten Fall betrachten wir einen Webserver, der Forschungspapiere speichert und sie elektronisch den Browsern weltweit erreichbar macht. Sind die Papiere erst einmal auf dem Server gespeichert, ändern sie sich nicht mehr. Eine Indexseite könnte

umgeschrieben werden, jedes Mal wenn die Papiere dem Server hinzugefügt werden, aber die Papiere selbst ändern sich nicht. Es ist eine statische Webinstallation.

Betrachten wir jetzt eine Firma, die sorgfältig Unternehmen in einer spezifischen Industrie verfolgt ( z.B. Fluggesellschaften ) und diese Information für die Teilnehmer am Web erreichbar machen will.

Einige der Informationen sind statisch und, einmal definiert, können sie auf Standardwebseiten gespeichert und via Webanschlüsse abgerufen werden. Aber der Veröffentlicher möchte vielleicht auch aktuelle Neuigkeiten als einen Teil des Service machen. Informationen über eine Fluggesellschaft könnte einen Standardüberblick über das Unternehmen beinhalten, aber auch neueste Nachrichten. Falls ein Teilnehmer "News" aus einem der vordefinierten Seiten auswählt, könnte dies ein CGI Programm aufrufen, das Zugriff zu einem News Transportservice hat, Informationen über die Fluggesellschaft sammelt, eine Webseite formattiert und die Webseite zurück zum Browser via HTTP schickt. Diese Webseite unterstützt dynamische Veröffentlichung.

Viel von der hochwertigen Interaktion auf dem Web kommt von dynamisch erstellten Seiten via CGI Programme. Damit ist der Hauptfaktor bei der Erstellung dieser Klasse von Anwendungen nicht das, was in HTML und HTTP ist, sondern vielmehr die Art der Anwendungsentwicklungsumgebung zum Schreiben von CGI Programmen, die mit externen Datenquellen interagiert. Deshalb sehen wir eine neue Generation von Anwendungsentwicklungstools für das Web. Daraus ist zu schließen, daß multimediale Anwendungsentwicklungsumgebung, die bereits kollaborative Anwendungen unterstützen, auch Webprotokolle unterstützen.

**Phase Zwei -- Feldformen.** Nehmen wir noch einmal das Beispiel der Fluggesellschaft auf. Reale User würden wahrscheinlich viel spezifischeres von den Nachrichten wissen wollen, die sie suchen ( z.B. Neuigkeiten über die Fluggesellschaft der letzten 24 Stunden oder einfach Artikel, die die Wörter "Flugpreis" und "international" enthalten ). Wir haben uns nun vom dynamischen Veröffentlichen zu *interaktiven Fragen* bewegt. Um diese Anforderungen zu unterstützen, müssen wir eine Seite dem Benutzer präsentieren, in der z.B. spezifische Suchbegriffe eingegeben werden können. Die Erweiterung von HTML zur Unterstützung von "Feldformen" beschreibt den zweiten großen Schritt in der Entwicklung des Web. Sind erstmal Daten in ein Feld eingegeben, wird ein URL zum Server zurückgeschickt, der ein CGI Programm aufruft, das die Daten aus der Form herausholt, die Information bearbeitet und dynamisch eine Webseite erstellt und zum User zurückschickt.

Feldformen ermöglichen nicht nur, daß das Web interaktive Frageanwendungen unterstützt, sie ermöglichen auch elektronischen Handel und Transaktionsdienste ( z.B. Reservationssysteme der Fluggesellschaft, bestandsanzeigenden Service, Einzelhandelskataloge und Auftragsformulare, Unterstützungssysteme für Kunden ). Die Vorteile eines webbasierten Transaktionsprozesses für die Lieferanten sind (1) weltweite Verbindung durch das Internet und (2) universelle Unterstützung für jedes Kundengerät, das einen Webbrowser hat. Während das Web als kollaboratives System für Informationsteilung via Veröffentlichungsmodell gestartet ist, ist man heute wegen den Transaktionssystemen und elektronischem Handel für das Web begeistert -- Bereiche außerhalb dessen, was normalerweise als kollaboratives Computing gesehen

wird. Da das Modell der dynamischen Veröffentlichung immer wichtiger für und wird, ist Programmierung jenseits von HTML via CGI erforderlich, was zur nächsten Phase der Entwicklung des World Wide Web führt.

**Phase Drei -- Programmiersprachen.** Die Anfangsversion von HTML sagte ausdrücklich, wie ein Dokument formatiert werden sollte. Betrachtet man dies jedoch von einer leicht anderen Sichtweise, so war HTML eine einfache Programmiersprache. Ein HTML Statement könnte z.B. aussagen, "wenn ein User auf diesen Hot Spot mit der Maus klickt, schicke die folgenden Hot Spot Koordinaten als eine Bitte an den geeigneten Webserver, zu dem der Webserver dann eine URL zurückschickt." Dies ist ein einfaches Programm. Es ist die Entwicklung des HTML, die die dritte Phase der Entwicklung des Web beschreibt.

Genauso wie Programmiersprachen sich zu einem objektorientiertem Modell zur Unterstützung der Anforderungen von modernen Anwendungen entwickelt haben, wird sich auch HTML zu einem objektorientierten Modell entwickeln. Browser können einen Code haben, der ein Programm entziffert, und Webseiten können Programme haben. Wenn eine Webseite geöffnet wurde und zu einem anfragenden Browser gesendet wurde, interpretiert der Browser das Programm und führt die erwünschten Aktionen aus ( z.B. Animation zu einem Bild verschaffen ). Das Auftreten von webspezifischen Sprachen erweitert erheblich die Programmierbarkeit und Erweiterbarkeit von HTML und folglich auch von Browsern.

### **Grenzen des Web**

Das Web entwickelt sich rapide, und seine Grenzen und Einschränkungen sind in diesem frühen Stadium nicht so leicht zu erkennen. Nach gewisser Zeit werden wir erkennen, daß es, wie bei jeder Innovation, Grenzen gibt, die verstanden und respektiert werden müssen. Es ist bereits klar, wo wir in naher Zukunft einige Herausforderungen sehen werden.

Was das Web und die Webbrowser anfangs so attraktiv machte, war die ungeheure Einfachheit der Technologie und dem resultierendem Produkt. Webbrowser waren kompakt, einfach zu implementieren und zu benutzen. Komplexität wird mit steigender Funktionalität neuer Browser zunehmen, die folgendes beinhalten:

- Unterstützung multipler Kodierungsformate ( z.B. Adobe Acrobat in Ergänzung zu HTML ).
- Einschließen und Unterstützen multipler Sprachen.
- Unterstützung lokaler Filesysteme ( z.B. ein lokales Messagestore für ein E-Mailsystem, geschrieben für das Web ) und sein Nutzen für "caching" webseiten, oder das Speichern von Repliken der Daten auf Webservern.
- Der Zusatz von reichhaltigen Sicherheitsmerkmalen für Browser und bessere Zugriffskontrollen auf Servern.

Um deutlich hervor zu heben, dies ist keine düstere und verurteilende Vorhersage für das Web. Ganz im Gegenteil, das Web hatte und wird eine tiefgreifende Auswirkung

auf die Industrie der Informationstechnologie und Gesellschaft haben. Aber durch das Hinzufügen von Funktionalität zu dem Web, fügen wir automatisch Komplexität hinzu.

Die Einführung von Komplexität wiederum macht die Interoperabilität schwieriger. Das Web ist heute sehr offen. Webserver und Webbrowser sind interoperabel, weil die Protokolle einfach und uniform implementiert sind. Jetzt, da sich das Web aus einem Forschungsprojekt, in dem Konzepte wie Profit und Marktanteil keine Themen waren, zu einer marktgetriebenen Industrie entwickelt hat, sehen wir Erweiterungen von HTML Merkmalen, so daß das Webserverprodukt eines Verkäufers am besten mit dem Browser dieses Verkäufers funktioniert. Dies könnte den negativen Effekt haben, daß User verschiedene Browser implementieren müssen. So benutzen sie Browser A, weil der auf den Stockquotingservice hin optimiert wurde und Browser B, weil dieser am besten mit dem lokalen Server der Benutzers arbeitet. Die Komplexität wird weiter zunehmen, bis die "Standard" Kriege geführt und gelöst sind.

Wir werden weiterhin dramatische Investitionen in das Web und die zugehörigen Technologien sehen. Webtechnologie hat sich aus Kollaboration ( dynamische Informationsveröffentlichung ) zu elektronischem Handel entwickelt. Während es weiterhin Innovationen in den kollaborativen Aspekten des Web geben wird, ist es wahrscheinlich, daß der Schwerpunkt sich zum elektronischem Handel verschieben wird. Der Schwerpunkt von Groupware, die auch mit einer kollaborativen Basis startete, wird zu Koordinationsanwendungen verschoben werden, dem Thema des nächsten Kapitels. Weil die entsprechenden Pfade sich noch entwickeln und verändern, können wir weitaus mehr Komplementarität als Überschneidung erwarten.

## Folgerungen

Bei der Untersuchung der Kombination von kommunikations- und kollaborationsbasierten Technologien wird folgendes deutlich:

- \* Die Anforderungen an Kollaboration und Kommunikation sind unterschiedlich. Daraus folgt, daß electronic messaging alleine nicht ausreichend den Prozeß der Kollaboration erleichtert. Datenbanktechnologien wenden ein "pull" Modell der Informationsverteilung an, die Benutzer aktiv in den kollaborativen Prozeß einbezieht.
- \* Kollaboration benötigt ein System, daß diese "push" und "pull" Modelle kombiniert und robuste Rahmenbedingungen schafft, um die vielen Wege der Kommunikation und Kollaboration der Benutzer verwerten zu können.
- \* Eine gemeinsame Datenbank ist wesentlich für gemeinsame Arbeit, gemeinsame Sichten und zur Einbindung von Informationen in das Wissen der Organisation. Ein Weg der Integration der "push" und "pull" Modelle geschieht durch Tools, die eine koordinierte Nutzung der Technologien des Messaging und der gemeinsamen Datenbanken unterstützen. Diese spiegeln das Bedürfnis der Gruppen um Koordination ihrer Arbeitsbemühungen wieder und beseitigen Schwierigkeiten beim Übergang zwischen den verschiedenen Arbeitsweisen.

Die dritte Kategorie der Gruppenarbeit - Koordination - wird von beiden dieser o.g.

Technologien unterstützt, genauso wie durch Tools, die Gruppen erlauben, ihren kombinierten Nutzen dieser beiden Technologien zu "programmieren". Das ist Thema des nächsten Kapitels.

## **Koordination**

Bis jetzt haben wir darüber gesprochen, wie Gruppen von Menschen kommunizieren und kollaborieren, um Informationen zu teilen und Wissen zu erweitern, das ihnen hilft, ihre Arbeit effizienter und effektiver auszuführen. Was vieles von dieser Interaktion charakterisiert, ist ihre ad hoc und unstrukturierte Beschaffenheit. Das bedeutet, Menschen schicken sich gegenseitig nach ihren eigenen Ermessen e-mails zu, und sie verweisen auf gemeinsame Ressourcen, wenn das Bedürfnis existiert. Die Aktivitäten geschehen auf einer dynamischen Basis. Wenn wir an Kollaboration denken, denken wir an "Brainstorming" Sitzungen, gemeinsames Erstellen eines Forschungspapieres, oder anders "wir wissen nicht genau, wo es enden wird" Arten von kreativen Aktivitäten.

Doch viele Geschäftsaktivitäten sind in Wirklichkeit viel strukturierter. Unternehmen erwarten nicht, daß die Leute über das Erstellen eines Ausgabenreports "kollaborieren"; vielmehr beschreibt ein Unternehmen spezifische Richtlinien darüber, wie ein Ausgabenreport durch ein Unternehmen geschickt werden sollte, so daß er angemessen geprüft, genehmigt und gesichert ist.

Viele Menschen sind dabei involviert, aber die Unternehmensrichtlinien sagen ausdrücklich vor oder diktieren sogar die erforderliche Koordination der Menschen, um ein definiertes Ziel zu erreichen. Die erfolgreiche Erfüllung eines vordefinierten Betriebsprozesses hängt von der Koordination der Leute bei der Erfüllung einer Menge strukturierter Aufgaben in einer bestimmten Reihenfolge und innerhalb erwarteter Zeiteinschränkungen ab. Zum großen Teil ist dies der Bereich der Automatisierungssysteme des Arbeitsflusses gewesen -- eine Konzentration auf hochstrukturierte Betriebsprozesse, die durch vordefinierte, konditionale Workflows charakterisiert sind, die auf Status und Bedingungen basieren. Während Kollaboration relativ passiv, aus der Perspektive des Systems gesehen, ist ( wir schaffen einen gemeinsamen Arbeitsplatz, aber wie schreiben nicht vor, wie dieser Platz genutzt wird ), ist Koordination, aus der Perspektive des Systems gesehen, sehr aktiv ( wir sagen ausdrücklich, wie Aktivitäten erfüllt werden müssen ).

Wenn wir eine Verschiebung von Kollaboration zu Koordination für ein spezifisches Problem vornehmen, implementieren wir Arbeitsflußsysteme, in die wir Formulare definieren. Es werden bestimmte Abläufe auf diesen Formularen angegeben, eine logische Route wird für sie vorgeschrieben, es wird angegeben, wie man Zugriff auf externe Daten hat oder sie verändern kann, Auslöseaktionen vorgeschrieben, die eintreten, wenn bestimmte Bedingungen eintreffen, u.s.w. . Wir entwickeln Workflowanwendungen, und dafür braucht man Anwendungsentwicklungstools. Koordination in diesem Sinne bezieht sich auf Anwendungsentwicklungstools für Anwendungen, die generell als Workflow bezeichnet werden können. Eine Haupteigenschaft ist oft das "Tracking" ( Auffinden, Suchen ) einer Ressource. Das wesentliche Tool für das Errichten eines Koordinationssystems ist eine Anwendungsentwicklungsumgebung.

Während das Nutzen einer Struktur in Workflowanwendungen wichtig ist, übersieht sie ein bedeutendes Segment der Koordination. Damit sind Aufgaben und Aktivitäten

gemeint, die nicht vordefiniert sind. Die meiste reale Arbeit enthält eine Kombination hoch strukturierter Prozesse und Arbeiten, in die der Prozeß unklar ist und die Regeln, Flußfolge und Rollen dynamisch definiert sind, wenn die Arbeit getan wird. Deshalb sind Workflowsysteme allein, ohne die kollaborativen und kommunikativen Komponenten, die für "softe" Interaktion sorgen, oft nicht erfolgreich und können als zu "starr" erachtet werden.

Betrachte zum Beispiel eine Anwendung für Softwarefehler. Die strukturierten Schritte könnten die anfängliche Registration des Fehlers beinhalten, dem Projektmanager einreichen, dem Programmierer / Analytiker übereignen, zur Qualitätssicherung weiter schicken, dem Managementspezialisten für die Konfiguration liefern und einer Präsenzbibliothek ( z.B. dem World Wide Web oder bulletin board ) zum Einsehen für Kunden bekannt machen. Während dieses Prozesses aber gibt es wahrscheinlich einige unstrukturierte Schritte, die nicht antizipiert oder automatisiert werden können. Z.B. kann eine fehlersuchende Datenbank nicht in der Lage sein, ähnliche Fehler und Mängel zu identifizieren. Koordination ist dann mehr als die Automation einer Reihe strukturierter Aufgaben und mehr als Menschen in und aus einem Prozeß zu bringen. Wenn wir uns ansehen, wie Arbeit wirklich getan wird, sehen wir, daß Wissen, das wesentlich zur Erfüllung eines Prozesses ist, als ein Resultat der Beziehungen unter den verschiedenen Teilnehmern erworben wird. Dies geschieht außerhalb des Kontext des Prozesses selbst. Vollständige Koordination beinhaltet die Unterstützung informeller Gespräche ( durch E-Mail, Diskussionsdatenbanken und Referenzpublishingsystemen ), die es Menschen erlaubt, Informationen zu sammeln, die sie zur Erfüllung ihrer Arbeit benötigen, besonders wenn diese Gespräche im Zusammenhang eines mehr strukturierter Prozesse stattfinden.

In den vorangegangenen Kapiteln untersuchten wir, wie die einzelnen Modelle von Kommunikation und Kollaboration, die auf das Senden und Holen von Informationen basieren, unter der Unvollständigkeit litten, wenn sie über ihren ursprünglichen Designpunkt hinaus angewandt wurden. Eine integrierte Lösung, die den Gebrauch von Messaging für Mitteilungen und gemeinsame Datenbanken für Kollaboration koordiniert, verschafft einen ausgeglicheneren und umfassenden Zugang zur Unterstützung strukturierter und unstrukturierter Prozesse.

## **Vier Generationen des Messaging**

Die vierte Generation der Messaging Systeme verwertet die koordinierte Nutzung von Messaging unter Einbeziehung gemeinsamer Datenbanken. Wenn wir uns anschauen, wie sich Messaging Systeme entwickelt haben, können wir erkennen, daß die erste Generation nur in der Lage war, einfache Textnachrichten zu unterstützen.

Die zweite Generation verbesserte diese durch die Fähigkeit, binäre Dokumente an die einfachen Textnachrichten anzufügen.

Die dritte Generation unterstützte zusätzlich rich text Format (z.B. Farben, verschiedene Fonts, Schriftgrößen usw.) und bot außerdem die Möglichkeit, Objekte wie ein binäres Attachment direkt in die Nachricht einzubetten.

Schließlich repräsentiert die vierte Generation einen deutlichen Fortschritt des Messaging Systems durch die Unterstützung von Hypertext auf Dokumente, die in gemeinsamen Datenbanken und herkömmlichen File-Systemen enthalten sein können. Anstelle ein Objekt an einer Nachricht anzufügen, fügen wir einen "doc link"

, einen elektronischen Wegweiser auf ein Objekt in einer gemeinsamen Datenbank, in die Nachricht ein.

Wenn der Benutzer einen Doppelklick auf den Doclink macht, wird das Objekt automatisch sofort geladen und dem Benutzer präsentiert. Das Messaging System wurde verbessert, weil es nicht länger erforderlich ist, daß der Benutzer Objekte und Attachments an Nachrichten direkt anhängt. Das System der gemeinsamen Datenbanken wurde verbessert, weil nun der Benutzer anderen Benutzern die Existenz relevanter und wichtiger Informationen mitteilen kann, die andernfalls möglicherweise in einer Datenbank untergegangen wären, unbenutzt und unbemerkt. Wir nennen verweisen hierzu auf den Punkt "integrated messaging and groupware"

### **Vier Generationen des Messaging**

Betrachten wir eine koordinierte Aktivität. Ein Marketingdirektor versendet zur Überprüfung regelmäßig Entwürfe (komplett mit Texten und Grafiken) gedruckter Anzeigen an eine Gruppe von Editoren.

Für jede Anzeige benutzt der Direktor e-mail, um die Anzeige an die Editorengruppe weiterzuleiten. Diese Anzeige wird als binäres Attachment zu der Nachricht gesendet. Antworten verschiedener Gruppenmitglieder - in Form von aktuellen Änderungen der Anzeigen oder Vorschlägen und Kommentaren - führen schnell zu einem Problem der Versions - und Dokumentenkontrolle. Gleichzeitig, erhöht sich durch das ständige Hin- und Herschieben von Nachrichten das Übermittlungsvolumen und damit der Netzwerkverkehr dramatisch.

Bei der alternativen Nutzung einer gemeinsamen Datenbank vermeidet der Direktor viele dieser Probleme.

Die gemeinsame Datenbank erlaubt dem Direktor und den Korrektoren die Betrachtung der jüngsten Version und vermeidet eine unstrukturierte Diskussion, die zu verschiedenen Änderungen führt. Auf der anderen Seiten ermangelt es dieser Lösungen an der Möglichkeit zur Notifikation, die benötigt wird, wenn eine Anzeige zur Ansicht in die Datenbank eingefügt wurde.

### **Vierte Generation des Messaging**

Die koordinierte Nutzung von Messaging und gemeinsamer Datenbanken in der vierten Generation der e-mail Systeme (integrated messaging and groupware) löst beide dieser Probleme.. Wenn der Direktor eine neue Anzeige in die Datenbank einfügt, wird automatisch eine e-mail Nachricht geschaffen, die einen Hypertextlink oder "pointer" auf die neue Anzeige enthält.

E-mail wird benutzt, um die Betrachter über die neue Anzeige zu informieren, aber die Anzeige selbst wird nicht physikalisch als Attachment zu der Nachricht transferiert. Statt dessen führt der Dokumentenlink in der Nachricht den Betrachter dynamisch zu der neuen Anzeige, die physikalisch weiterhin in der gemeinsamen Datenbank verbleibt. Dadurch können Anzeigen zentral verwaltet und gelagert werden, was eine Versionskontrolle und Kollaboration in der gemeinsamen Datenbank ermöglicht.

## **Integriertes Messaging und Groupware**

In einem vorangegangenen Kapitel erklärten wir, daß e-mail Benutzer Groupware als

Messaging betrachten, während Benutzer von Konferenz- und On-Line Publishing-Systemen, Groupware als eine Funktion gemeinsamer Datenbanken sehen. Benutzer, die auf die Automatisierung strukturierter Betriebsprozesse konzentriert sind, verbinden Groupware mit Workflow-Automation.

Jede dieser Technologien allein betrachtet ist eine spezifische Dimension von Groupware. Messaging und gemeinsame Datenbanktechnologien sind die Grundlage für Workflow-Automationsysteme geworden. Messaging hat sich in den Workflowbereich verschoben, indem es seine APIs den Einrichtungen und Tools der Anwendungsentwicklungs zur Verfügung stellt. Damit kann ein Verfahren zur Führung ( routing ) des Arbeitsflusses erstellt werden, um Prozesse zu automatisieren. Gemeinsame Datenbanken sind ähnlich erweitert worden, um ein Verfahren zur Verfolgung (tracking) der Automatisierung des Arbeitsflusses( Workflow-Automation ) zu unterstützen.

### **Messagingmodell von Workflow**

Workflow-Automation wird als automatisches Weiterschicken von Dokumenten wie z.B. Kostenprotokolle betrachtet. Auf eine Route basierende Workflow-Automation benutzt im allgemeinen das vorhandene Messagingsystem, um ein Dokument zur nächsten Person zu schicken, die eine Handlung vornehmen muß ( z.B. das Kostenprotokoll genehmigen ). Die Route kann stark kodiert sein, oder eine Regel kann den Pfad des Routing festlegen, der von speziellen Werten ( z.B. die Summe der Kosten ) oder von der Rolle einer Person ( z.B. der Aufseher des Initiators ) abhängt. Diese Regeln können sehr ausgearbeitet sein, indem sie sogar eine externe Anwendung abrufen können, um gewisse Daten einzuholen ( z.B. die Autoritätsbeschränkung des Aufsehers ).

Der Arbeitsfluß, der auf das Weiterschicken basiert, ist leistungsfähig, weil es das Modell des Weiterschicken eines Papers erfüllt: das Dokument wird bearbeitet und zur nächsten Person für weitere Handlungen gesendet.

Es gibt einen entscheidenden Nachteil bei dem Workflow-Modell, das auf das Senden eines Dokuments basiert: wenn das Dokument unterwegs ist, ist es für jeden unerreichbar, außer der Person, in deren Inbox es liegt. Die damit verbundenen Probleme werden in dem folgenden Beispiel des Vertragsrouting dargestellt, in dem ein Vertrag mit einem Kunden ausgehandelt wird. Um ihn zu unterzeichnen, muß der Vertrag erst intern von einigen Leuten genehmigt werden. Ein mailbasiertes Workflow-System schickt den Vertrag zu jede Person, um Genehmigung, Verweigerung und / oder Kommentare zu erhalten. Angenommen der Kunde würde den Status des Vertages wissen wollen oder würde gern Änderungen an dem Vertrag während des Prüfungsprozesses vornehmen wollen. Ist es möglich festzustellen, wer den Vertrag zu dem jeweiligen Zeitpunkt des Prozesses hat ? Falls ja, kann man was abfragen ? An welcher Stelle können Änderungen vorgenommen werden ? Was passiert mit dem Genehmigungsprozeß, wenn eine Änderung einmal vorgenommen wurde ?

Lösung einiger dieser Probleme ist möglich ( vielleicht die Regel umschreiben oder Hinzufügen einer neuen Regel, die eintritt, wenn eine Person in Urlaub geht ), aber eine strategische Anwendung auf solchen Pfaden zu bilden, ist für Unternehmen beschwerlich, und das Wahrnehmen aller Bedingungen und Ausnahmen ist unmöglich.

## **Gemeinsames Datenbankmodell von Workflow**

Das zweite Workflow-Modell ist die gemeinsame Datenbank. In diesem Modell konsultieren Benutzer eine tracking Datenbank, um den Status von bestimmten Dokumenten festzustellen.

Das gemeinsame Datenbankmodell hat drei Vorteile.

Erstens, die Datenbank befindet sich auf dem Server und ist Sachgebiet von Prozessen, die auf dem Server ablaufen ( wie RDBMS Trigger, Agenten oder Makros ). Diese können eine Aktion ohne eine bestimmte Aktivität des Benutzers auslösen. In vielen Fällen könnte die Aktion aufgrund fehlender Benutzeraktivität ( ein Verkäufer hat in 30 Tagen einen Kunden nicht kontaktiert, ein Monatsreport wurde nicht abgegeben, ein Vertrag, der genehmigt werden muß, ist von der dafür vorgesehenen Person schon seit 24 Stunden unbearbeitet, usw. ) oder wegen einer externen Bedingung ausgelöst werden ( Materialbestand ist auf den Nachbestellungsstand gesunken, Kreditwürdigkeit eines Kunden hat sich geändert, eine Frist naht ).

Zweitens, das gemeinsame Datenbankmodell hält das betreffende Dokument oder Protokoll für andere erreichbar, während der Arbeitsfluß weitergeht. In dem Beispiel der Vertragsgenehmigung könnten die Änderungen an dem Originaldokument in der Datenbank vorgenommen werden und, abhängig von den Änderungen, könnte der Arbeitsfluß fortfahren oder abgebrochen und neu gestartet werden.

Der dritte Vorteil ist, daß das gemeinsame Datenbankmodell das Verwalten und Makromanagement des Arbeitsflusses viel einfacher macht. Der Server kann sowohl bestimmte Stellen des Prozesses beobachten, als auch zusammengefaßte Statistiken über den Gesamtprozeß halten. Das letztere erlaubt ein besseres Management und eine bessere Planung des Arbeitsflusses.

Die grundlegende Einschränkung des gemeinsamen Datenbankmodells ist das Fehlen von Mitteilungen, die von bestimmten Ereignissen abhängen und durch das System den Workflow-Teilnehmern berichtet werden. Das heißt, es ist incumbent dem Benutzer gegenüber, die Datenbank zu kontrollieren.

## **Ein integriertes Modell**

Diese Einschränkungen der Messaging- und der gemeinsamen Datenbankmodelle des Arbeitsflusses sind bekannt, weil sie Einschränkungen ihrer zugrundeliegenden Technologien reflektieren. Messaging ist sehr effizient beim Senden eines Dokuments, aber es verschafft keine Möglichkeit, das Dokument zu verwalten, wenn es seine Route durchläuft. Gemeinsame Datenbanken sind fähig beim Verwalten von Dokumenten und verschaffen einen Überblick, aber sie sind unzureichend beim Anzeigen von Status- oder Informationsänderungen den Benutzern.

Wir haben bereits die Vorteile eines vollkommen integrierten Messaging / gemeinsamen Datenbanksystems über vier Generationen Messaging gesehen. Wenn wir dieses Modell verallgemeinern, wird folgendes klar: Workflow-Anwendungen, die auf einer gemeinsamen Plattform errichtet sind und ursprünglich beide Modelle unterstützen, -- ein Untersystem von Messaging mit eingeschränkten Sendefähigkeiten und eine gemeinsame Datenbank zum Speichern, Abrufen, Betrachten und Verwalten

von Arbeitsprozessen -- verschaffen die nötige Robustheit und Flexibilität, um effektiv Arbeitsprozesse zu automatisieren.

Zur Illustration betrachten wir das selbe System des Vertragsrouting und Tracking, das vorher angesprochen wurde. Wie könnte es implementiert sein, wenn man ein integriertes Modell benutzt ? Der Vertrag wird erstellt und in einer gemeinsamen Datenbank gespeichert. Dann wird ein Mail zum ersten Prüfer gesendet. Die Nachricht selbst ist nicht der Vertrag und enthält auch keine Information, wo der Vertrag zu finden ist. Die Nachricht enthält einen Hypertextlink auf den Vertrag. Wenn dieser aktiviert wird, wird der Vertragsinhalt geöffnet, so daß der Prüfer ihn bearbeiten kann, während der Vertrag in der gemeinsamen Datenbank bleibt. Weitere Prüfungen und anschließendes Senden können vorgenommen werden, aber die aktuellste Version des Vertrages ist immer in der Datenbank abrufbar. Jede Anfrage über Status kann von jedem mit Zugriff zur Datenbank beantwortet werden.

## **Das erweiterte Transaktionsmodell**

Wie wir gesehen haben existieren zwei grundlegende Verfahren eines Arbeitsflußsystems: ein System basiert auf dem Messaging Modell, das andere System basiert auf dem Modell der gemeinsamen Datenbanken. Workflow-Systeme, die beide Modelle und wie die weniger strukturierten kollaborativen Funktionen unterstützen, werden die erfolgreichsten Systeme sein.

Es ist wichtig zu erkennen, daß diese Workflow-Systeme nur selten isoliert existieren. Häufig bilden die Workflow-Prozesse die "front-office" Komponenten, die mit spezifischen, schon seit langem automatisierten, "back-office" Systemen verbunden sind.

Folglich ist die Verbindung zwischen "Output" des Workflow-Systems und dem "Input" des eher klassischen Transaktionssystems der kritische Punkt. Ein Beispiel wird hilfreich sein und der Prozeß des Einkaufes dient als gutes Beispiel.

Ein typischer "Einkaufsprozeß" beginnt mit genehmigten Einkaufsanträgen, die in den offiziellen Einkaufsauftrag einmünden. Dieses Systeme sind typischerweise sehr stark automatisiert und die meisten Kosteneinsparungsmöglichkeiten durch Automatisierung sind hier bereits realisiert worden..

Der Input in dieses System der. Der genehmigte Einkaufsantrag ist das Ergebnis eines Workflow-Prozesses, der damit beginnt, daß jemand entscheidet, daß sie oder er einen Bedarf an etwas hat.

Diese Person schreibt einen Einkaufsantrag, und diese Anforderung "geht seinen Weg" durch die verschiedenen Systeme, bis sie schließlich genehmigt oder verworfen wird. Wenn die Anforderung bestätigt wird, wird die "Transaktion" im klassischen Sinne eingeleitet. In der Realität jedoch begann die Geschäftstransaktion damit, wenn jemand einen Einkaufsantrag erschafft. Die meisten back-office Systeme besitzen wesentliche front-office workflow Komponenten.

Wir bezeichnen diese breitere und mehr geschäftsorientierte Sicht von Transaktionen als das erweiterte Transaktionsmodell. Durch die enge Einbindung der front-office Arbeitsflußprozesse in die existierenden back-office Prozesse können erhebliche ökonomische Gewinne erreicht werden. Um die enge Einbindung zu erreichen, müssen wir Tools verwerten, die uns eine kontrollierte Einwirkung auf die einzelnen

Transaktionssysteme erlauben. Viele Transaktionsprozeß-Systeme unterstützen diese Form programmierter Schnittstellen.

Bei der Betrachtung von Workflow als Teil einer erweiterten Transaktionen wird sehr deutlich, daß der Schlüssel zur Entwicklung von multimedialen Arbeitsfluß-/Koordinationsanwendungen eine in das Workflow-System integrierte Anwendungsumgebung ist, die ein Teil dieses Systems ist.

## **Rahmenbedingung für Anwendungsentwicklung**

Es gibt eine Anzahl von wesentlichen Komponenten für eine reichhaltige Anwendungsentwicklungsumgebung zur Errichtung von Koordinationsanwendungen:

- Ein reichhaltiger Formulardesigner- und Füllzeichentool sind erforderlich. Formulare beinhalten Text, Bilder, Geräusche, Felder, Buttons, Nachrichtenfenster, usw. und können aus Teilformularen bestehen. Diese Teilformulare können Fenster oder Dialogfenster mit bekannten Nachrichtenfenstern, Komboboxen, usw. sein. Leistungsfähige Editiermöglichkeiten sind für Felder in dem Formular erforderlich, und diese Editierung könnte eine Verbindung zu externen Datenbanken benötigen, um Zugriffe von Benutzern auf ihre Gültigkeit zu überprüfen.
- Einige Formen von Programmiermöglichkeiten ist in der Form einer Scripting-Sprache, Makrosprache oder umfassenden Programmiersprache erforderlich. Eine reiche Menge von APIs ist für die zugrundeliegende Technologie erforderlich. Entwickler müssen in der Lage sein, jedesmal Instruktionen zu beschaffen ( z.B. durch Skripte ), wenn Ereignisse wie das Öffnen eines Formulars, Wechseln eines Feldes oder Ergänzung eines neuen Dokuments an der Datenbank vorgenommen werden.
- Eine Agentenmöglichkeit ist erforderlich, so daß jedesmal Aktionen vorgenommen werden können ( wie das Starten eines Skriptes ), wenn ein bestimmtes Ereignis eintritt.
- Eine Entwicklungsumgebung, die eine reichhaltige Fehlerbeseitigungsmöglichkeit unterstützt, ist erforderlich.
- Auf Enduser abgestimmte Ansichten sind erforderlich, um die kundenspezifische Anzeige von Informationen zu unterstützen, die den Aufgaben, individuellen und Gruppenanforderungen entsprechen.

Wenn diese Tools entsprechend angewendet werden, um Workflow-Anwendungen zu entwickeln, kann das Resultat sehr robuste Anwendungen sein, die eine natürliche Benutzerschnittstelle und gleichzeitig die nötige Kontrolle für Unternehmensanwendungen haben.

## **Folgerungen**

In den meisten Geschäftsprozessen ist der Prozeß unklar und die Regeln, Wege und Rollen werden durch die aktuell zu bewältigende Arbeit bestimmt. Es ist dieser Aspekt der Verschwommenheit der Arbeit, die die Wissensvermittlung und -übertragung so anspruchsvoll macht und für die Manager verantwortlich gemacht werden. Unsere Diskussion der strukturierten und unstrukturierten Aktivitäten, die jeder Geschäftsprozeß enthält, führt zu folgenden Schlüssen:

\* Die meiste in der Realität geleistete Arbeit hat eine dynamische Bewegung zwischen strukturierter, unstrukturierter, ad-hoc und vordefinierter Arbeit zur Folge, die ein integriertes push/pull Modell benötigt. Es unterstützt den Benutzer bei ihrem Übergang von einem Arbeitstyp zum nächsten, wie dies typischerweise bei einem normalen Geschäftsverlauf zu finden ist.

\* Strukturierte, vordefinierte Gruppenaktivitäten können durch programmierte Workflow Anwendungen unterstützt werden, von denen es zwei grundlegende Typen gibt: routing, das auf der Messaging Technologie basiert, und tracking, daß auf der Technologie der gemeinsamen Datenbanken basiert. Die Integration dieser beiden Verfahren zu einer Workflow-Automatisierung ist durch eine integrierte Anwendungsumgebung erreicht worden, die die Dienste sowohl des Messaging Systems und das System der gemeinsamen Datenbanken ausnutzt.

\* Die Anwendungsentwicklungsumgebung ist eine Schlüsselkomponente einer Groupware Systemarchitektur. Dieses wird ausführlicher im Kapitel "Architektonische Überlegungen" diskutiert.

## Architektonische Betrachtungen

Bis jetzt haben wir den Bedarf an einer Groupware-Infrastruktur festgestellt, die nicht nur Anwendungen unterstützt, die individuell von Kommunikation, Kollaboration oder Koordination abhängen. Sie sollte vielmehr durch die Integration dieser drei Bereiche die entstehende Synergie ausnutzen.

Dadurch können wir die Hauptanforderungen an eine solche Infrastruktur feststellen.

- Ein Informationsmodell, das auf eine Datenbank / verteilte gemeinsame Datenbank basiert und die Daten speichert und verwaltet, ohne Rücksicht auf ihre ursprüngliche Quelle.

- Ein **Verteilungs- und Zugriffsmodell**, das auf Messaging und Datenbankreplikation für die mehrfache Verschiebung von Daten von einer Stelle zu einer anderen in der Organisation basiert.

- Ein **Anwendungsentwicklungsrahmenbedingung**, die die ursprünglich zugrundeliegenden Dienste eines Objektstores und Verteilungs- / Zugriffsmodells erweitert, um eine Umgebung für Kundengroupware-Anwendungen zu entwickeln.

Während dies die Hauptkomponenten einer integrierten Groupware-Architektur darstellt, ist die Integration externer Datenquellen, Sicherheit und Verzeichnisdienste enorm wichtig.

# Information Model: The Groupware Objekt Store

Das Object Store ist das Herz einer Groupware Infrastruktur. Um eine effektive Verwaltung von Geschäftsinformationen zu leisten, muß das Objektstore die Fähigkeit besitzen, einen gemeinsamen Zugang zwischen Benutzern und Applikationen durch das ganze Dokument hindurch zu gewährleisten. Konkret ist das Objektstore der Nachrichtenspeicher (Message-store) für Kommunikationsanwendungen, ein virtueller gemeinsamer workspace für kollaborative Anwendungen und eine gemeinsame Datenbank für die Koordination.

Das Objektstore sollte sich intern über alle der o.g. Anwendungen erstrecken. In anderen Worten, es gibt keinen Grund, warum die interne Struktur eines Nachrichtenstore einer Benutzermailbox architektonisch unterschiedlich zu einer Konferenzdatenbank oder einer workflow Anwendung aufgebaut sein sollte. Das Erscheinungsbild auf der Benutzerebene mag leicht unterschiedlich sein, aber auf dem niedrigsten Niveau der Datenebene (Daten in individuellen oder rich text Feldern, Attachments und Objekten) kann und sollte die Organisation dieser Daten übereinstimmend sein.

Dieses Modell bietet folgende Vorteile:

- \* Konsequente und folgerichtige Methode der Informationsverarbeitung durch alle Stufen der Kommunikation, Kollaboration und Koordination - für Endbenutzer wie auch Anwendungsentwickler
- \* Wirkliche Trennung zwischen Daten und Anwendungen
- \* Ein Datensatz an Informationen für viele Verwendungszwecke - eliminiert Datenredundanz und die damit verbundenen Probleme.

Das Groupware Objekt Store ist eine verteilte, gemeinsame Datenbank. Auf seiner niedrigsten Stufe enthält das Objektstore Objekte. Eine verteilte Groupwareumgebung besteht aus einem beliebigen Netzwerk von Servern. Innerhalb eines Servers existiert eine Anzahl individueller Datenbanken. Innerhalb der gegebenen Datenbank existiert wiederum eine Anzahl von Dokumenten. Das Dokument selbst besteht aus einer Anzahl von Feldern.

Ein Dokument präsentiert im Regelfall die Daten dem Endbenutzer. Auf dieser Basis erfolgt die Verwaltung von Daten in einem Objektstore. Die Fähigkeit, Informationen auf diesem Level verwalten zu können, erlaubt eine wesentlich effektivere Informationsverarbeitung als dieses herkömmliche Message-Stores ermöglichen.

Um ein vollständiges System zur Informationsverarbeitung darzustellen, sollte das object store folgendes unterstützen:

- \* Rich Objekte. Die Breite von Objekttypen, die ein object store unterstützen kann, bestimmt und begrenzt die Anzahl von Anwendungen, die es unterstützt. Ein Dokument sollte eine große Vielfalt von Objekten unterstützen, einschließlich Zahlen,

Text, rich text, Grafiken, Bilder, Sprache, Video, Verknüpfungen zu anderen Dokumenten und eingebetteten Anwendungen.

\* Dokumentenhierarchie. Ein object store benötigt die Fähigkeit, Eltern-Kind Beziehungen von Dokumenten und den entsprechenden Antworten aufdecken zu können. Dies ist besonders bei Gruppenkonferenzen und Diskussionsdatenbanken weitverbreitet .

\* Versionskontrolle (Versioning) Unterstützung einer Versionskontrolle, falls wichtige Änderungen an einem Dokument vorgenommen werden, an dem mehrere Autoren beteiligt sind.

\* Hypertext Links. Unterstützung von Verkettungen zwischen Dokumenten sowohl innerhalb einer Datenbank als auch zwischen unterschiedlichen Datenbanken. Dieses ist eine sehr flexible Möglichkeit, auf weitere Informationen, die in Datenbanken enthalten sind, verweisen zu können. Diese Verkettungen sollten Teil des Dokuments sein, so daß sie in einer verteilten Umgebung gepflegt und verwaltet werden können.

\* Konsistenz. Alle Anwendungen des darunterliegenden Objektstore sollten auf einem Satz grundlegender architektonischen Grundsätze basieren. Diese Konsistenz ermöglicht eine Anzahl von wichtige Merkmalen und Fähigkeiten, die für das System als Gesamtheit gelten. Volltextsuche und Möglichkeiten zur Rückgängigmachung eines Befehls können, wenn sie in das Groupwaresystem eingebunden wurden, einheitlich über alle Typen der Datenlagerung hinweg handeln, sei es e-mail oder eine Hilfeanwendung. Dieses ermöglicht dem Anwender eine konsistente Anwendererfahrung für Daten und Anwendungen, unabhängig von Form oder Stelle. (location)

## Verteilungstechnologien

### Store-and-Forward Routing

Store-and-forward routing ist die Schlüsselkomponente für Messaging und das push model, die im Bereich der Kommunikation verwendet werden. Das Verteilungsmodell des Messaging basiert auf einen store-and-forward asynchronen Transport. Dieser Transport ist verantwortlich für das Weiterschicken von Informationen von Sender zu Empfänger, wie durch das push model im Bereich der Kommunikation beschrieben wurde.

SMTP/MIME und X.400 haben sich zu den geläufigsten Industrienormen der Messaging-Transporte entwickelt. Unterstützung dieser Normen ist wichtig, da sie eine wichtige Rolle bei der Unterstützung der Interoperationalität von Messaging spielt, die inner- und überbetrieblich zwischen heterogenen Messaging-Umgebungen stattfinden.

Haben die Nachrichten erst einmal ihren endgültigen Bestimmungsort erreicht ( d.h. Messagestore / Objektstore ), kann der Benutzer die Nachrichten vom Server zum Client replizieren, so daß er beim Benutzen dieser Nachrichten nicht am Server angeschlossen sein muß. Sie haben den Status wie jedes andere Dokument in der Datenbank auch und enthält alle Leistungen dieser Datenbank ( vgl. Client

Replication im folgenden Abschnitt ). Für das Senden von Server zu Server aber, werden Mails anders als andere Dokumente im store-and-forward Mechanismus verteilt.

## **Replikation**

Die Groupwaredatenbank ist eine verteilte gemeinsame Datenbasis. Während dieses die Vorteile eines verteilten computing model beibehält, ist noch eine Technologie erforderlich, um eine konsistente und logische Ansicht von physisch verteilten Informationen zu präsentieren. Datenbankreplikation erfüllt dieses, indem sie Veränderungen an vielfältigen Kopien derselben Datenbank an geographisch verteilten Standorten synchronisiert. Zum Beispiel kann von einer entfernten Anlage in San Francisco eine Replik der Datenbank in Paris gemacht werden. Dieses erlaubt Benutzern in San Francisco, Zugriff zu dieser Information auf einem lokalen Server zu haben, ohne an den Datenbankserver in Paris angeschlossen zu sein. Replikation synchronisiert automatisch Veränderungen, die an beiden Orten gemacht werden, so daß Arbeitsgruppen in San Francisco und Paris eine konsistente, logische Ansicht der Datenbank haben. Während der Replikation werden vom jeweiligen Server ergänzte, veränderte oder gelöschte Dokumente untersucht. Danach werden beide Datenbanken auf den neuesten Stand gebracht, so daß sie beide identisch sind.

Während der Prozeß der Replikation klar erscheint, ist es wichtig, den Mechanismus der Replikation bei der Auswahl von Technologie genauer zu untersuchen. Implementierungen variieren stark und sie haben eine entscheidende Wirkung auf Netzwerk- Topologien. Diese sind erforderlich bei der Unterstützung einer verteilten Umgebung. Das folgende ist eine nähere Betrachtung der Anforderungen an eine Replikation.

- **Bidirektionale Replikation.** Sind zwei Repliken einer DB erst einmal synchronisiert, fangen Arbeitsgruppenmitglieder an, auf verschiedenen Netzwerken und Servern, Änderungen, Löschungen und Ergänzungen an ihnen vorzunehmen. Danach sind beide Repliken nicht mehr synchron zueinander. Deshalb sollte der Server in San Francisco bei der nächsten Replikation in der Lage sein, alle Änderungen, Ergänzungen und Löschungen auf den Server in Paris zu übertragen. Gleichzeitig sollten alle Änderungen, Ergänzungen und Löschungen zurückrepliziert werden, die auf dem Server in Paris gemacht wurden.

- **Effizienz.** Angenommen, daß Netzwerke über Unternehmen hinaus stark variieren. Der Prozeß der Replikation sollte hoch anpassungsfähig und optimiert sein, das Volumen des Netzwerkverkehrs zu minimieren. Die wichtigste Determinante für optimale Netzwerknutzung ist die Granularität von Replikation. Zum Beispiel muß der Server in Paris nicht jedesmal die gesamte San Francisco Datenbank kopieren, wenn eine Änderung in einem Feld oder auf Dokumentenebene erfolgt ist. Nur die Felder, die sich an einer Stelle geändert haben, müssen repliziert werden.

- **Client Replication.** Mobile oder nomadische Benutzer in einer Arbeitsgruppe brauchen den selben Grad, um wie angeschlossene Benutzer Zugriff zur Datenbank zu haben . Die Art, wie ein Benutzer zu Informationen Zugriff hat und mit ihnen arbeitet, sollte konsistent sein. Es spielt keine Rolle, ob er mit dem Netzwerk verbunden ist

oder nicht. Deshalb sollte Replikation nicht auf Server-zu-Server Verbindung beschränkt sein, sondern auch Client-zu-Server Verbindung einschließen.

Client-zu-Server Replikation gibt dem mobilen Benutzer die Möglichkeit eine lokale Replik einer Datenbank zu behalten (oder einige unterschiedliche Datenbanken von einigen unterschiedlichen Servern). Er kann mit ihr Off-Line genauso arbeiten, als ob er mit dem Server verbunden ist. Besteht dann wieder einmal Verbindung zum Server, wird die Replikation benutzt, um Änderungen zu synchronisieren, die Off-Line erstellt wurden.

Dabei ermöglicht die Server -zu-Server Replikation, daß die Änderungen über das Unternehmen verteilt werden. Die Fähigkeit, Dokumente auszuwählen und Datenbanken "mit auf den Weg zu nehmen", ist eine natürliche Erweiterung der Art, wie wir Aktentaschen packen, wenn wir auf Geschäftsreise gehen. Es kann alles repliziert werden, was in einer Datenbank gespeichert werden kann. Deshalb ermöglicht Client Replication Benutzern, eine konsistente Ebene von Kommunikation, Kollaboration und Koordination zu jeder Zeit, an jedem Ort zu haben. Nomadische Benutzer können Messagespeicher ( E-Mail ), Diskussionsdatenbanken, Verweisinformationen und jede Koordination- oder Workflow-Anwendung mit auf den Weg nehmen, indem sie eine einzige Methode benutzen.

- **Programmable Control.** Replikation sollte ein großes Maß an Kontrolle den Verwaltern und Endbenutzern geben, indem sie ihnen erlaubt, Dokumente, die auf relevanten Kriterien basieren, selektiv zu replizieren; zum Beispiel Dokumente, die sich geändert haben, wie z.B. ein bestimmtes Datum , ein Dokument vom Autor, Größe, Kundename. Wegen der begrenzten Zeit und Ressourcen-- wie limitierter Zugang zu teuren Telefonleitungen, Funkkommunikation oder begrenzter Speicherplatz auf der Platte-- haben mobile Benutzer sogar einen größeren Bedarf an dieser Art von Programmierung.

- **Anwendungsdistribution.** Eine der Herausforderungen des Verwaltens einer Plattform übergreifenden, verteilten Client / Server Umgebung ist die Installation und die Verbesserung von Anwendungen. Dieses wird besonders deutlich in einer Groupware Umgebung, in der Gruppenbedürfnisse sich ständig ändern. Deshalb sollte eine Anwendungseinsatzumgebung die Datendistributions-/- replikationsmöglichkeiten der Datenbank erweitern. Replikation sollte benutzt werden, um Anwendungsobjekte und Änderungen an Anwendungen genauso einfach wie Daten zu verteilen. Jedesmal wenn ein Benutzer Zugriff zur Datenbank hat, wird die aktuellste Version der Anwendung gestartet.

Die Groupware-Infrastruktur benutzt die Replikation zur Verteilung und zum Einsatz von Daten. Die Anwendung wird automatisch durch eine Client Verbindung zum Server aktualisiert. Dieses erleichtert den Anwendungsentwicklern und Systemadministratoren die enorme Last der manuellen Installation neuer oder modifizierter Anwendungen für verteilte Clients und Server über ein Unternehmen hinaus.

- **Punkt-zu-Punkt Transfer.** Zuverlässige Synchronisierung von Datenbanken erfordert ein interaktives Mittel eines "Händeschüttelns " zwischen Systemen. Ein solches Modell muß dafür synchron sein. RPC (Remote Procedure Call ) sorgt für einen zuverlässigen Transport für Punkt-zu-Punkt Replikation. Dieses wird durch direkte Verbindungen mit dem Endpunkt der Replikation erreicht.

Das alles steht im Gegensatz zu dem asynchronen Modell, das beim Messaging store-and-forward benutzt wird, welches Nachrichten über viele Zwischenetappen schickt. In diesem Modell würden Veränderungen, die auf der Datenbank vorgenommen werden, zu anderen Datenbanken gesendet. Auf dem ersten Blick scheint dies, eine kluge Lösung zu sein, um eine bestehende Messaging Infrastruktur zu erweitern. In der Praxis jedoch ist store-and-forward Replikation deutlich komplizierter und schwieriger zu verwalten. Einer der größten Probleme ist, daß eine garantierte Sendung von Nachrichten nicht so leicht zu bestimmen ist, wie in einer Punkt-zu-Punkt Umgebung. Store-and-forward Messaging dauert zum Beispiel eine ungewisse Länge an; d.h., man kann nicht genau vorhersagen, wie lange eine elektronische Nachricht dauert, wenn sie über einige Stellen verläuft, bis sie ihr Ziel erreicht. Dieses Problem betrifft besonders große Unternehmens- und öffentliche Netzwerke, in denen die Route einer e-mail dynamisch festgelegt ist ( d.h., wenn die Nachricht gesendet wird; und dieses kann unterschiedlich zu jeder Zeit sein ). Deshalb ist es schwierig zu wissen, wann, oder sogar ob die Datenbanken tatsächlich auf dem neuesten Stand sind.

## Umgebung für Anwendungsentwicklungen

Ein wichtiges Charakteristika einer Groupware-Plattform ist die Möglichkeit, schnell Anwendungen entwickeln und einsetzen zu können. Die Plattform sollte die Fähigkeit besitzen, das volle Spektrum von Anwendungsentwicklungen unterstützen zu können, die vom Endbenutzer ohne Programmiererfahrung über erfahrene Benutzer bis hin zu professionellen Entwickler reicht.

Jede Gruppe von Entwicklern hat natürlich eigene, spezielle Anforderungen: Endbenutzer, die eine leichte Bedienung verlangen und professionelle Entwickler, die auf die Stabilität der Anwendungsumgebung vertrauen.

Eine stabile, robuste Anwendungsumgebung sollte den Entwickler effektiv vor den meisten Anwendungsüberlegungen schützen:

\* **Plattformen.** Da die Groupware-Infrastruktur wie eine Schicht zwischen Anwendungen und der darunterliegenden Hardware/Software Betriebsumgebung handelt, können Anwendungen ohne Rücksicht auf ihre spezifische Desktop, Netzwerk und Serverumgebung entwickelt werden. Alle systemspezifischen Attribute einer Anwendung, wie individuelle Merkmale einer Desktop Grafikschnittstelle, werden ohne zusätzliche Programmierung oder Modifikation unterstützt.

\* **Anwendungseinsatz.** Da die interne Anwendungslogik wie jedes andere Objekt innerhalb des Object-store behandelt wird, kann der Anwendungsentwickler die Distributionsmöglichkeiten des Objekt-stores für die Verbreitung seiner Anwendung benutzen. Entwickler müssen sich nicht über die Verteilung und Installation von Anwendungen sowie um mögliche Updates kümmern. Anwendungen sind selbstverbreitend und können nach allen Seiten hin verteilt werden, unabhängig von der geographischen Herkunft.

\* **Unterstützung der mobilen Nutzung.** Die Fähigkeit der Groupware-Infrastruktur, mobile Benutzer unterstützen zu können, befreit Entwickler von der Schaffung einer mobilen Version ihrer Anwendungen. Jede Anwendung arbeitet gleich, unabhängig

von ihrer Nutzung im On-Line oder Off-Line Modus.

## **Programmiersprachen und Tools**

Wie bereits diskutiert wurde, befindet sich bei einer integrierten Messaging und Groupware Infrastruktur die interne Logik der Anwendungen in dem Object-Store. Um dem Entwickler die Schaffung entsprechender Prototypen zu erleichtern, bietet es eine moderne Plattform für eine schnelle Anwendungsentwicklung.

Wenn eine Logik einmal in eine Datenbank eingefügt wurde, ist keine weitere Kompilierung notwendig.

Jede Anwendung kann an eine Auswahl an Benutzer zu Testzwecken verteilt werden, auch um eine Akzeptanz zu gewähren. Jede vom Benutzer gewünschte Änderung kann ohne einen langatmigen Reprogrammierungs- und Rückverteilungsprozeß entsprechend umgesetzt und schnell wieder an die Benutzer zurückgeben werden.

Eine integrierte Programmiersprache innerhalb dieser Umgebung sollte die zwei folgenden Kriterienbündel erfüllen:

Zum ersten muß die Programmiersprache die grundlegenden Anforderungen jeder professionellen Entwicklungsumgebung erfüllen. Dieser Satz von Anforderungen unterscheidet es von einer Anwendungsentwicklung, die Templates und Makrosprachen benutzt

\* **Volle Strukturierung.** Um hochentwickelte Groupware Anwendungen entwickeln zu können, müssen Entwickler über komplexe logische und flow-of-control - Fähigkeiten verfügen, genauso wie die Programmierung von Konstrukten wie z.B. Schleifen und Verzweigungen in der Programmiersprache benötigt wird.

### **\* Professionelle Entwicklungsumgebung**

Im Gegensatz zur spartanischen Kommandoschnittstelle, das die meisten Makrosprachen unterscheidet, benötigt eine robuste Groupware Programmiersprache wie jede Programmiersprache eine moderne, vollständig interaktive Entwicklungsumgebung, einschließlich einem hochentwickelten Editor, einem Referenzwerk und einem interaktiven Debugger. Dieses ist die Antwort auf die Tatsache, daß die Programmiersprache als Werkzeug für professionelle Entwickler und erfahrene Benutzer mit echter Programmiererfahrung dient.

### **\* Hoher Grad an Abstraktion**

Moderne Client-Server Systeme und ihre ereignisbezogenen graphischen Benutzerschnittstellen verlangen ein hohes Maß an Fähigkeiten von Seiten der Programmierer. Die Sprache sollte dem Anwender notfalls auch ohne große Programmierkenntnisse die Entwicklung graphischer ereignisorientierter Schnittstellen ermöglichen.

### **\* Moderne Benutzerschnittstellen Unterstützung**

Die Programmiersprache muß die Entwicklung von Benutzerschnittstellen in einer modernen GUI Umgebung erleichtern. Es muß einfach sein, Formulare entwickeln,

anzeigen und editieren zu können, pop-up Fenster (.z.B. Dialogboxen) zu schaffen und standardisierte GUI Schnittstellen Tools wie z.B. Buttons zu unterstützen.

Eine starke ereignisorientierung muß unterstützt werden, so daß ein Entwickler bestimmen kann, welche Handlungen vorgenommen werden soll, wenn z.B. ein Benutzer auf die Maus klickt oder wenn er Daten in ein Feld eingibt.

Zum zweiten soll eine hochintegrierte Sprache die Vorteile aller Systemdienste der Groupware-Plattform ausnutzen. Dieses erfordert eine Unterscheidung zwischen Entwicklungswerkzeugen, die eine Groupware Entwicklungsumgebung unterstützen, aber nicht notwendigerweise die einzigartigen Plattformdienste der Groupware-Infrastruktur verwerten.

\* **Unterstützung von Client und Server Plattformen.** Die meisten Groupware Anwendungen beinhalten ihre Funktionalitäten sowohl auf dem Server als auch auf dem Clienten. Applikationen, die mit einer Programmiersprache geschrieben sind, sollten nicht durch nur-client oder nur-server Skripte eingeengt werden. Dieses ist im Gegensatz zu herkömmlichen auf datenbankbasierten Skripten basierenden Sprachen, die nur auf dem Server durch gespeicherte Prozeduren operieren. Ebenso arbeiten naturgemäß Desktop Programmiersprachen nur auf Desktop Plattformen. Eine einzelne Groupware-Programmiersprache sollte auf beiden Seiten des Netzwerkes arbeiten, alle Vorteile der Client/server Architektur in Anspruch nehmen.

\* **Einfacher Zugang zu Systemdiensten.** Entwickler benötigen kompletten Zugriff zu den Möglichkeiten des Object-Store und der Messaging Infrastruktur. Möglichkeiten wie Replikation, Sicherheit und Messaging müssen dem Entwickler, der diese Programmiersprache benutzt, zugänglich gemacht werden.

\* **Vielfältige Plattformunterstützung.** Moderne client/server Anwendungen unterstützen vollständig die geschäftlichen Abläufe, häufig über Unternehmensgrenzen hinweg, um Kunden, Lieferanten und Geschäftspartner einzubeziehen. Programme, die mit einer Groupware Programmiersprache geschrieben wurden, müssen daher auf allen Ergänzungen der client und server Plattform lauffähig sein. Darüber hinaus sollten diese Programme ohne Änderung oder Rekompilierung auf plattformspezifische Anforderungen hin anpaßbar sein.

\* **Interoperationalität.** Es ist wahrscheinlich, daß Entwickler mehr als ein Programmierwerkzeug anwenden werden, um eine einzelne Anwendung zu schaffen oder eine existierende Anwendung zu verändern. Daher benutzen Entwickler häufig Templates, vorprogrammierte Buttons oder Makrosprachen einer Groupware-Umgebung. Daher sollte die Programmiersprache systemübergreifend mit diesen anderen Entwicklungsobjekten zusammenarbeiten können. Zusätzlich muß die Sprache Entwicklern erlauben, API's Dritter einzubeziehen, um sich dem Ziel einer komplexen Systemintegration zu nähern.

Eine integrierte Programmiersprache, die diese Kriterien erfüllt, ist eine interessante Alternative zu Templates, Makrosprachen und zu stand-alone Tools und Sprachen zur Schaffung hochwertiger client/server Groupware Anwendungen.

## **Endbenutzerentwicklung**

Die größte Herausforderung an die Groupware-Infrastruktur ist das Bedürfnis, zwei sich diametral entgegenstehende Prinzipien in Einklang zu bringen: Anwendungen zu schaffen, die von Gruppen genutzt werden, während gleichzeitig mit großer Wahrscheinlichkeit die individuellen Benutzer die einzelnen Komponenten des gemeinsam genutzten Anwendung in persönlicher, einzigartiger Gestaltung betrachten wollen.

Bei der Diskussion über die Charakteristika der Rahmenbedingungen für Groupware-Anwendungs-Entwicklungen erwähnten wir das Bedürfnis nach leichtem Zugang zu den Diensten des Groupware Object-Store und den entsprechenden Verteilungseinrichtungen. Entsprechend haben die Benutzer das Bedürfnis nach leichtem Zugang zu den im Objekt Modell gespeicherten Daten, um sie auf spezifische Zwecke hin zu anzupassen. Beispiele für benutzerdefinierte Erweiterungen existierender Anwendungen beinhalten:

\* **Individuell maßgeschneiderte Views.** Es gibt kein einziges optimales Format, Informationen zu präsentieren. Jedes Gruppenmitglied hat seine oder ihre persönlichen, besonderen Bedürfnisse. Manche möchten Dokumente nach Datum sortiert haben, andere wünschen eine Sortierung nach Namen oder nach Titel. Darüber hinaus ist auch der Informationsumfang, der auf dem Bildschirm präsentiert wird, von Bedeutung. Aus diesem Grund werden Benutzer bestehende Anwendungen durch die Schaffung ihrer eigenen privaten Informationssichten verändern.

\* **Kategorisierung.** Die meisten Menschen organisieren Informationen durch die Aufspaltung dieser Informationen in Kategorien. Eine Kategorie könnte ein Projektname sein, eine Benutzerreferenz, oder ein bedeutungsvolles Thema. (z.B. konkurrierende Informationen). In der Realität können einige Informationen in mehr als eine Kategorien fallen. Der wichtige Punkt ist hier natürlich, daß es dem Benutzer obliegt, diese Kategorien zu definieren, daß es der Benutzer ist, der die Bedeutung der einzelnen Kategorienamen kennt. Daraus folgt, daß alle Dokumente, die sich z.B. auf ein Projekt zur Fehlerbehebung beziehen, von einer Person unter der Kategorie "Fehlerbehebung", von einer anderen Person unter der Kategorie "Wichtig" und von einer dritten Person unter dem direkten Projektnamen geführt werden können.

\* **Agenten.** Jede Person hat seine eigene Auffassung, was wichtig, relevant oder dringend ist. Nachrichten von Personen zu empfangen, die selbst entscheiden, was dringend oder wichtig ist, ist eine Sache. Eine andere Sache ist die uneingeschränkte Fähigkeit, einen Berg von Informationen zu sichten und für einen selbst zu entscheiden, was wichtig oder nicht wichtig ist. Durch den Aufbau von Agenten können die einzelnen Benutzer die Informationssuche automatisieren, so daß das Groupwaresystem selbst anhand benutzerdefinierter Kriterien Informationen sucht, findet und zurückholt.

\* **Weitreichende Erweiterungsmöglichkeiten.** Anwendungen, die für relativ kleine Gruppen geschaffen wurden, wachsen häufig, um mehr Benutzer einzuschließen. Die Marketingabteilung umfaßt in einem bestimmten Geschäftsvorgang auch Mitglieder der Verkaufs- und Produktentwicklungsabteilung und möchte diesen neuen Teilnehmern Zugang zu einer Anwendung und dessen Daten gewähren. Anstelle sich an den ursprünglichen Designer zu wenden, können erfahrene Benutzer selbst

Änderungen an den Zugangsberechtigungen vorzunehmen, also die ursprüngliche Teilnehmergruppe zu erweitern.

## Integration Externer Datenquellen

Die meisten Informationen, die in einer Groupware Anwendung vorhanden sind, wird auch dort konkret erstellt. Das heißt, ein Individuum gibt die Information direkt in ein Groupwaredokument ein. Die meisten Betriebsprozesse verlassen sich aber auf Daten, die in anderen Datenspeichern in Ergänzung zu der Groupwaredatenbank existieren. Eine Groupware Infrastruktur sollte problemlos strukturierte Daten, die in relationalen Datenbanken gespeichert sind, einführen, teilen und erweitern. Das Gleiche sollte sie mit semi-strukturierten Daten vornehmen können, die in externen Datenquellen wie Desktoptools ( z.B. Textverarbeitungsprogramme und Arbeitsblätter ), Dokumentmanagementsysteme und öffentliche Informationsnetzwerke gefunden werden können.

- **Bezugsdaten.** Halbstrukturierte Daten verschaffen oft erst den Kontext, in dem strukturierte Daten eine Bedeutung und Relevanz bekommen. Bestandsebenen und Statistiken über Auftragsbearbeitungen sagen mehr aus, wenn illustrative Details wie Beschreibungen der Produktzyklen, verarbeitende Techniken, eine sich ändernde Wettbewerbsumgebung und Kundenfeedback mit angeführt werden. Die Integration von strukturierten und halbstrukturierten Informationen ist ganz entscheidend für den Wert einer Groupware Anwendung und Infrastruktur.

Die Groupwaredatenbank kann nicht das RDBMS ersetzen; das Designzentrum eines RDBMS erfordert gewisse Charakteristiken ( leistungsfähige Verschlusssysteme, Transaktionen, ... ).

Im Gegensatz dazu erfordert das Designzentrum einer Groupwaredatenbank Unterstützung für ein verteiltes, gelegentlich angeschlossenes Modell. Die beiden Technologien ergänzen sich gegenseitig.

- **Bild / Videoserver.** Bildrepositorien und Videoserver haben ähnliche Anforderungen -- beide müssen nämlich einen großen Speicher haben, und Videoserver müssen zusätzlich die Fähigkeit haben, Daten in isochroner Art ( konstanter Datensatz ) zu liefern. Weder ein traditionelles RDBMS, noch eine Groupwaredatenbank bieten die geeigneten Mittel für solche Formen von Daten. Trotzdem erkennt der Daten- / Informationsbedarf von Benutzern und Groupware-Anwendungen nicht die technologischen Grenzen von Informationsspeichern und -management. Um dieses besser zu erkennen, ist es wichtig, zwischen einer logischen und physischen Ansicht von Informationen zu unterscheiden. Um eine konsistente, logische Ansicht von Informationen den Benutzern oder Groupware-Anwendungen zu präsentieren, muß die Datenbank der Integrationspunkt von verschiedenen Datenquellen sein.

- **Desktop Produktivitätstools.** Microsoft hat das Objekt Linking und Embedding ( und in einem geringen Ausmaß den dynamischen Datenaustausch ) als Standard für seine Windowsfamilie der Betriebssysteme definiert. Ein Konsortium, bestehend aus Apple Computer, IBM / Lotus Development und Novell hat Fortschritte bei der Bestimmung von OpenDoc, einem plattformübergreifenden Datenintegrationsstandard, gemacht, das auf IBM's System Object Model basiert. Der

gemeinsame Datenspeicher muß sich nach diesen Normen richten, um eine weitreichende Integration mit Desktopprodukten zu gewährleisten.

- **Internet und andere öffentliche Informationsnetzwerke.** Eine der reichhaltigsten Speicher von semi-strukturierten Informationen kann man in den Diskussionsgruppen und World Wide Web Seiten des Internet finden. Diese Informationen stellen genauso viel an Unternehmenswissen dar wie eine Menge an internen Mitteilungen und Diskussionsdatenbanken.

Daher muß eine vollständige, moderne Groupware-Infrastruktur eine Möglichkeit bieten, um dieses Wissen zu erweitern ( verteilen ).

Internetressourcen sollten dem Groupwarebenutzer als ursprüngliche Ressourcen erscheinen, und alternativ sollten Benutzer in der Lage sein, ursprüngliche Groupwareinformationen direkt dem öffentlichen Netzwerk bereit zu stellen.

## Sicherheit

Messaging Systeme und andere semi-strukturierte Datenspeicher haben traditionell ein angemessen mächtiges und ausgefeiltes Sicherheitssystem. Die meisten Informationen, die in einer Nachricht oder in einer persönlichen Leistungsaufzeichnung gespeichert sind, haben einen persönliche Natur. Daher werden die Nachrichten, die in dem persönlichen Postkorb des Benutzers enthalten sind genauso wie die Dokumente, die auf der Festplatte des individuellen PCs gespeichert sind, als "Eigentum" des einzelnen Benutzers angesehen. Um sicherzustellen, daß nur der Besitzer dieser Informationen Zugang zu ihnen hat, soll das Zugangssystem gegen unberechtigten direkten Zugriff auf den Datenspeicher schützen. Entsprechend sind alle Informationen, die in einem öffentlichen Netzwerk System wie z.B. das Internet oder kommerziellen Onlinediensten gespeichert sind, allen autorisierten Nutzern zugänglich. Für die Autorisierung ist lediglich eine Anfangs-Zugangsberechtigung zu dem entsprechenden Service erforderlich. Für diese Gruppen von Datenmanagement dienen lange Zeit private Paßwörter als angemessenes Sicherheitssystem zum Schutz gegen unberechtigtem Zugriff.

Groupware dagegen macht Gebrauch von dem darunterliegenden Object-Store, von dem Benutzer durch einen "pull" Mechanismus Informationen beziehen können. Dieses Object-Store ist im Gegensatz zu den persönlichen Ressourcen des message stores eine gemeinsam genutzte organisatorische Ressource, die üblicherweise vertrauliche und geschützte Informationen enthält. Die Informationen, die in einem gemeinsam genutzten Objekt enthalten sind, benötigen daher ein höherentwickeltes Sicherheitssystem. Es ist nicht ausreichend, nur den Zugriff auf das System als Ganzes zu beschränken. Das Sicherheitssystem muß vielmehr verschiedene Grade von Zugriffsrechten schaffen und kontrollieren können. Dieses wird durch den Gebrauch verschiedener Schichten von Sicherheitsmechanismen erreicht: Autorisierung, die den Zugriff auf das System als Gesamtheit kontrolliert; Zugangskontrolle, die verschiedene Kategorien von Benutzerzugriffen auf Dokumente und Informationen etabliert, und Dokumenten- und Feldverschlüsselung, die ausgewählte Dokumente und Felder vor unauthorisierter Betrachtung schützt.

Auch damit das Messaging-System selbst seine Bedeutung als Medium für gemeinsam genutzte wichtige Informationen beibehalten kann, muß es als vertrauenswürdiger Nachrichtenkourier angesehen werden. Die Integrität individueller

Nachrichten - ihr Inhalt und ihre Urheberschaft - muß geschützt werden. Digitale Unterschriften, die auf den gleichen Verschlüsselungstechnologien aufbauen, die die Autorisierung nutzt, bilden eine vierte Schicht des Sicherheitssystems einer integrierten Groupware Plattform..

### **Autorisierung**

Die Fähigkeit, Identitäten von Benutzern als auch von Netzwerken schaffen zu können, ist der Eckpunkt eines vertrauenswürdigen Systems. Die Funktionalität der anderen Sicherheitssysteme beruht auf der Vertrauenswürdigkeit der Autorisierung. Autorisierungen, die auf einem System von Zertifikaten und Verschlüsselungen basieren, werden als "state of the art" angesehen: der de facto Industriestandard zum Zugang zu X.500 Verzeichnissen ist das X.509 Zertifikat, das auf der RSA Verschlüsselungstechnologie basiert. Diese Technologie gilt als das einzige Verschlüsselungssystem, das nicht Kompromissen ausgesetzt ist.

Die Verschlüsselung arbeitet wie folgt: Ein Benutzer erhält ein Zertifikat (oder ID-File), das den Benutzer durch Name, Paßwort, Lizenznummer, und einen persönlichen Verschlüsselungscode identifiziert. Der private Schlüssel ist der Gegenpart zum "öffentlichen Schlüssel", der in einem öffentlich zugänglichen Verzeichnis gespeichert ist. Es ist mathematisch unmöglich, den privaten Schlüssel aus dem öffentlichen Schlüssel herzuleiten. Wenn der Benutzer versucht, Zugang zu einem Server zu finden, sichert der folgende Vorgang die Autorisierung des Benutzers:

- \* Der Server sendet dem PC des Benutzers eine zufällig ausgewählte Zahl zu.
- \* Der lokale PC verschlüsselt diese Nummer mit dem privaten Schlüssel des Benutzers, der auf dem lokalen PC des Benutzers abgelegt ist.
- \* Das Ergebnis dieser Verschlüsselung wird an den Server zurückgegeben.
- \* Der Server verschlüsselt diese Zahlen mit dem öffentlichen Schlüssel des Benutzers, der in einem Verzeichnis auf dem Server abgelegt ist.
- \* Wenn die Nummer identisch ist, ist der Benutzer autorisiert und wird mit entsprechendem Zugang zu dem Server ausgestattet.

### **Zugangskontrolle**

Einige Personen müssen Zugriff auf bestimmte Informationen oder ganze Datenbanken haben, sollten aber von anderen, empfindlicheren Dingen ausgeschlossen werden können. Daher sollte ein Systemadministrator in der Lage sein, verschiedenen Gruppen oder Individuen unterschiedliche Zugriffsrechte einzuräumen, einschließlich Zugang zu Datenbanken, Dokumenten und Feldern innerhalb eines Dokuments. Auch der Zugang zu jeder Ressource muß besser unterteilt werden: in Fähigkeiten zum betreten, lesen, schreiben, verändern und Löschen von Objekten.

Zugangskontrollen sollten weiterhin flexibel genug sein, um die verschiedenen

"Modi" , die ein Benutzer möglicherweise benötigt, individuell anzupassen. Wenn z.B. kann ein Benutzer möglicherweise vollen Zugriff ("manager access") auf eine Datenbank haben, wenn diese an ein Netzwerk angeschlossen ist, und nur Leseerlaubnis wenn Autorisierung nicht möglich ist, z.B. wenn diese Datenbank an einen telefonischen Client angeschlossen ist.

### **Verschlüsselung auf Feld- und Dokumentenebene**

Manchmal möchte ein Benutzer gemeinsam genutzte Informationen auf der Feldebene innerhalb eines Dokuments an bestimmte Benutzer weitergeben, während gleichzeitig sichergestellt werden soll, daß keine weiteren Benutzer diese Informationen sehen können. Die Zugangskontrolle kann den Zugang auf der Feldebene für bestimmte Nutzerkategorien (Reader, Manager usw.) begrenzen aber nicht für einzelne Individuen. Daher kann der Datenbankentwickler diese vertraulichen Informationen, die nur für bestimmte Personen lesbar sein sollen, verschlüsseln. Die Verschlüsselung erfolgt durch den öffentlichen Schlüssel der Zielperson in dem entsprechenden vertraulichen Textfeld. Dadurch sind nur Benutzer mit dem korrespondierenden privaten Schlüssel in der Lage, dieses verschlüsselte Feld zu lesen. Diese Verschlüsselung kann auch zwischen Servern benutzt werden, so daß nur autorisierte Server besondere Dokumente oder Felder lesen kann.

### **Digitale Unterschriften**

Benutzer müssen häufig bestätigen, daß die sie erreichenden Informationen auch von dem gesendet wurden, der auf dem Dokument angegeben ist. Sie müssen außerdem sicherstellen, daß keine in dem Dokument enthaltene Information manipuliert wurde. Diese Bestätigung wird durch die digitale Unterschrift organisiert. Dieser Dienst ist das digitale Äquivalent zu dem vertrauenswürdigen Kurier, der einen Umschlag mit einem Wachssiegel überbringt. Wenn der Benutzer A digital eine Nachricht "unterschreibt", wird durch die Verwendung des privaten Benutzerschlüssels ein mathematischer Verschlüsselungsalgorithmus erzeugt und an die Nachricht hinzugefügt. Benutzer B erhält dieses verschlüsselte Dokument und entschlüsselt die Nachricht den öffentlichen Schlüssel des Absenders (der in einem öffentlich zugänglichen Verzeichnis verfügbar ist). Durch die Benutzung der Daten in der Nachricht kann die Verschlüsselung zurückgerechnet werden. Wenn die Schlüssel passen, ist die Absenderidentität bestätigt. Auf diesem Weg kann B sicherstellen, daß das Dokument tatsächlich von Benutzer A gesendet wurde und daß niemand die Nachricht auf seinem Übertragungsweg abgefangen hat.

## **Verzeichnisse**

Eine der zentralen Komponenten eines Groupwaresystems ist das Verzeichnis. Es ist ein Informationsspeicher, der Informationen über Benutzer und Ressourcen enthält. Als Teil der Groupwaredatenbank bekommt das Verzeichnis die selben Systemdienste wie die Datenbank selbst. Dieses unterscheidet es von den relativ einfachen message store Verzeichnissen durch die folgenden drei Arten: ihre Inhalte beinhalteten Daten, die über Name und Adresse hinausgehen; sie können Informationen speichern, die nicht nur Menschen betreffen; sie können über das verteilte Netzwerk repliziert werden.

Um eine Integration und Interoperationalität ( Zusammenarbeit ) mit anderen Verzeichnissen zu versichern, sollte das Groupwareverzeichnis konsistent mit der X.500 Architektur sein. Das Groupwareverzeichnis sollte eine reiche Quelle von Informationen sein, die seine Rolle über die sogenannten "white pages" und "yellow pages" der Namen und Orte der Beschäftigten hinaus erweitert.

Speziell sollten sie folgendes unterstützen:

- **Reichhaltigen Text.** Als Teil der Datenbank selbst, sollten Verzeichnisse und ihr Inhalt die selbe rich text Unterstützung haben wie andere Dokumente auch. Deshalb kann ein Verzeichnisdokument Objekte wie Userimages und voice, eingebettete Objekte und Tabellen enthalten.

- **Benutzerdefinierte Felder.** Systemadministratoren oder Enduser sollten die Informationen innerhalb des Verzeichnisses erkennen können. Durch die Festlegung individueller Felder können Benutzer persönliche, private Ansichten von Daten gebrauchen, die für andere Benutzer des Verzeichnisses nicht verfügbar sind.

- **Linked Documents.** Da das Verzeichnis selbst ein Dokument in der Datenbank ist, kann es Links zu anderen Dokumenten in der Datenbank erstellen. Dieses erlaubt Verwalten, wichtige beschreibende oder erklärende Informationen an den "Eingang" des Verzeichnisses "anzuhängen", ohne eine zusätzliche Speicherbelastung zu verursachen. Linked Documents helfen, den Bedarf an separaten Verzeichnissen zu mindern, der durch vielfältige "Besitzer" von Verzeichnisressourcen entsteht

Das Verzeichnis sollte Informationen enthalten, die nicht nur Benutzer-, sondern auch Unternehmensressourcen berücksichtigen. Das Verzeichnis dient sowohl für Benutzer als auch für das System selbst als Informationsquelle.

- **Systeme.** Für einige Anwendungen ist das Ziel einer Nachricht nicht eine Person, sondern ein Server, ein Fax, ein Telefon oder andere elektronische "Bestimmungspunkte". Die Adressen dieser Ressourcen können im Verzeichnis gespeichert werden.

- **Distributionslisten.** Elektronische Adressenlisten können im Verzeichnis gehalten werden, damit eine Verteilung durch e-mail oder Fax erreicht werden kann.

- **Öffentliche Schlüsselzertifikate.** Das Verzeichnis ist ein angemessenes Repository für das Speichern öffentlicher Schlüsselzertifikate. Sichere Anwendungen, die auf öffentliche Schlüsseltechnologien basieren, erfordern ein Repository, um öffentliche Schlüssel zu speichern. Diese verifizieren digitale Unterschriften und entschlüsseln Nachrichteninhalte unter einer Gruppe von Benutzern.

- **Rollen.** Verzeichnisse können Individuen mit organisatorischen Rollen in Verbindung bringen. Dieses kann in einem Arbeitsflußprozeß genutzt werden, und somit können workflowbasierte Anwendungen Rollen ( roles ) besser nutzen als bestimmte Individuen. Dieses ermöglicht eine leichtere Verwaltung von Arbeitsflußprozeßsen, wenn Menschen im Urlaub sind oder ihre

Arbeitsverantwortlichkeit sich ändert.

- **Routing und Replikationslisten.** Der Weg einer Nachricht über hubs und router ist durch Verfügbarkeit und Kosten festgelegt. Dieser ändert sich, in Abhängigkeit von der Tageszeit, der Dringlichkeit der Nachricht und anderen Kriterien. Ein Einstieg in ein Verzeichnis kann beschreibende Information bezüglich der routing Logik enthalten. Deshalb wird eine bestimmte Adresse immer mit einer Wegbeschreibung angegeben, um pünktlichen und effizienten Transport zu gewährleisten. Ähnlich verhält es sich, wenn eine Replikation initiiert wird ( normalerweise zu bestimmten periodischen Intervallen oder auf Anfrage eines Verwalters ). Der Replikationsprozeß stellt die Stelle (location) des zu replizierenden Gegenstücks fest und bestimmt dann den effizientesten Weg, um eine Verbindung herzustellen ( via Lan Verbindung, Telefonanruf, usw. ).

Das Groupwareverzeichnis beinhaltet auch die Replikationsfunktionalität des gesamten Systems.

Replikation erlaubt eine einfache Implementierung der Verzeichnissynchronisation. Erstens ist der Replikationsprozeß directional. Die Änderung, die an einem Verzeichnis auf einem bestimmten Server vorgenommen werden, werden automatisch während der Replikation synchronisiert. Dadurch kann jede Löschung, Änderung und Ergänzung, die auf beiden Seiten der Verbindung entstehen, nachvollzogen werden. Dieses kann im Gegensatz zum Verzeichis Propagation, die nur Änderungen in eine Richtung sendet und jede Änderung auf dem Empfängerverzeichnis überschreibt. Zweitens verwalten Verzeichnisse oft große Datenbanken und benötigen daher erhebliche Netzwerkkapazitäten, um Replikation innerhalb eines Unternehmens durchführen zu können.

Der Replikationsprozeß sollte erkennen können, welche Felder innerhalb des Verzeichnis sich geändert haben , ergänzt oder gelöscht wurden, und nur diese Änderungen replizieren. Replikation der gesamten Datenbank ( oder des Ordners ) würde die Systemressourcen sinnlos belasten.

## Über Lotus

Die Lotus Development Corporation wurde 1982 gegründet und ist eine 100 % Tochter der IBM Corporation. Lotus bietet hochqualitative Softwareprodukte und unterstützt Dienste, die das einzigartige Firmenverständnis für neue Wege zur erfolgreichen individuellen und geschäftlichen Zusammenarbeit reflektieren. Lotus` innovative Einstellung wird durch eine neue Klasse von Anwendungen deutlich, die in vorher nie möglich gewesener Art und Weise erlauben, auf Informationen zuzugreifen und entsprechend zu kommunizieren, sowohl innerhalb als auch außerhalb organisatorischer Grenzen.

Lotus bietet umfassende, preisgekrönte Produkte für Windows, OS/2, Macintosh, NT und UNIX Umgebungen, die leicht zu benutzen sind, und auch leicht miteinander zu benutzen sind.

### Ein Wort zu Lotus Notes

Lotus Notes ist das führende integrierte Messaging und Groupware Produkt. Ironischerweise gibt es nur wenig Einigkeit zwischen industriellen Analysten,

Kunden, Geschäftspartnern und Konkurrenten über das, was Notes wirklich ist. Es wird alternativ als Client-/Server Plattform zur Entwicklung und zum Einsatz von Groupware Anwendungen beschrieben, als eine Plattform für hochspezialisierte Applikationen und eine offene Umgebung, in die unabhängige Softwareentwickler ihre eigenen Produkte integrieren können.

Notes ist ein Produkt mit vielen Gesichtern. Die meisten Menschen verbinden Notes mit den Anwendungen, die sie sehen und mit der Umgebung, in der sie in Notes arbeiten - Mail, Diskussionsdatenbanken usw. Die Macht von Notes liegt darin, daß diese Anwendungen nur ausgewählte Beispiele einer großen Bandbreite von Anwendungen sind, die in Notes geschrieben werden können. Es gibt kein echtes "Notes Konferenzsystem"; vielmehr existieren unglaublich viele Beispiele für Konferenzsystem, die in Notes geschrieben werden können.

Notes ist im Kern eine Entwicklungsumgebung für Anwendungen. Häufig nimmt jemand ein bestehendes Konferenzsystem auf und ergänzt Felder zu der bestehenden Form. Dadurch kann die Struktur auf einen bestimmten Zweck hin ergänzt werden. (wenn Sie z.B. Informationen über eine Kundenanfrage hinzufügen, ergänzen Sie Felder, die Details über die Anfrage, mögliche Dringlichkeitsstufe usw. enthalten).

Wenn dieses mit Leichtigkeit getan wurde, haben wir ein neues Konferenzsystem, daß auf einen spezifischen Typ von Kollaboration hin zugeschnitten wurde. Diese Anwendungen nutzen alle Vorteile der darunterliegenden Lotus Notes Plattformdienste wie Messaging, Object Store, Replikation, Sicherungssysteme und Anwendungsumgebung. Die Notes Technologie dient als eine flexible Groupware Plattform oder Rahmenbedingung, innerhalb dessen Groupware Anwendungen erstellt und eingesetzt werden können.

Lotus Notes kombiniert durch seine Konzeption drei kraftvolle Gruppenarbeit unterstützende Technologien - messaging, verteilte Object Stores, und eine Entwicklungsumgebung für rich text Anwendungen - um die Architektur einzelner Komponenten für eine Vielzahl von Groupware Anwendungen hin zu formen. Diese Integration verteilter Datenbanken in ein Groupwaresystem ist bislang einzigartig.

Es unterstützt maßgeschneiderte Endbenutzerwerkzeuge, um einfache Änderungen an Datenstrukturen und Datenbanksichten vornehmen zu können. Es unterstützt eine Vielzahl von API's für Entwickler, die alternative UIs schaffen wollten, die auch von dem Datenbankmodell unterstützt werden sollen.

Die Implementierung der Technologie der verteilten Datenbanken ist ebenfalls ein einzigartiger Vorteil - auch gegenüber anderen Groupware Produkten. Lotus Notes robuste Replikation speziell ausgewählter Netzwerke ermöglicht auch dann den Einsatz, wenn nur wenige Organisationen eine vollständige Netzwerkanbindung innerhalb ihrer Firma erreicht haben. Es berücksichtigt damit außerdem den wachsenden Bedarf an getrennten Nutzungsmöglichkeiten von Workstations und Notebook Computern.

Eine vollständige Industrie von unabhängigen Softwareverkäufern, Systemingenieuren, Anwendungsentwicklern und Beratern hat sich rund um Notes entwickelt. Lotus kultiviert weiterhin diese Gemeinschaft, um sicherzustellen, daß der Kunde über genügend Ressourcen zur Installation, Entwicklung, Einsatz und Führen von Lotus-basierten Anwendungen verfügt.