

Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse

Sven Schwarz, Andreas Abecker, Heiko Maus, Michael Sintek

Deutsches Forschungszentrum für Künstliche Intelligenz
– DFKI GmbH –
Postfach 2080, D-67608 Kaiserslautern
Phone: +49-631-205-3451, Fax: +49-631-205-3210
(schwarz,aabecker,maus,sintek@dfki.uni-kl.de)

Zusammenfassung Wissensmanagement zielt darauf ab, die Durchführung wissensintensiver Geschäftsprozesse im Unternehmen effizienter und effektiver zu machen. Während Software-Unterstützung für das WM momentan vorwiegend auf Kommunikations- und Kollaborations-Unterstützung bzw. verbessertes Informationsmanagement baut, war ein gezielter Geschäftsprozess- bzw. Workflow-Support für wissensintensive Tätigkeiten bisher kaum Gegenstand der Betrachtung. Ausgehend von einer Zusammenfassung der Merkmale gegenwärtiger Workflowmanagementsysteme und einer Charakterisierung wissensintensiver Aktivitäten, leiten wir in diesem Papier Anforderungen für eine Prozessmanagement-Unterstützung für den Wissensarbeiter ab, skizzieren einige existierende Arbeiten in diese Richtung und diskutieren unsere momentane Konzeption zur softwaretechnischen Umsetzung der gewünschten Funktionalitäten.

1 Ansatz und Ziele von Workflowmanagement

Heutige Unternehmen müssen sich dem globalen Wettbewerb stellen, ständig die Geschäftskosten senken und schnellstmöglich neue Dienstleistungen und Produkte anbieten. Dies erfordert ein konstantes Überdenken, Optimieren und Modifizieren der Geschäftsabläufe und eine effektive Gestaltung des Informationsflusses. Workflow(Wf)-Technologie erleichtert dies mittels Methoden und Software zur Unterstützung von (i) Geschäftsprozess-Modellierung zur Speicherung der Geschäftsabläufe als Wf-Spezifikation, (ii) Geschäftsprozess-Reengineering zur Optimierung der Geschäftsabläufe und (iii) Wf-Automation (Generierung von Wf-Implementationen) zur Durchführung und Kontrolle der Geschäftsabläufe.

Die Zielsetzung der Wf-Automation durch ein Workflow Management System (WfMS) wird i.a. als *“Provide the right information and the right tools to the right receiver at the right time.”* beschrieben, was direkt zu den folgenden vier Fragen führt (zitiert nach [14]):

- WAS ist die richtige Information?
- WIE wird die Information bearbeitet / geliefert?
- WER bekommt die Information bzw. WER bearbeitet eine Information?
- MIT WELCHEN TOOLS werden die Informationen bearbeitet?

Dabei wird also neben der Datenmodellierung und der Beschreibung der in einen Prozess involvierten Applikationen bei der zweiten (und teilweise der dritten) Frage die interessante Problematik der Beschreibung der Prozesslogik eines Vorganges angesprochen.

In konventionellen WfMSen wird die Durchführung eines Wfs mittels eines strengen Ablaufplanes – ähnlich einem Flussdiagramm – vollzogen. Die Prozesslogik wird somit mehr oder weniger auf das Abarbeiten dieses festen Ablaufplanes beschränkt. Der Ablaufplan wird *vor* der Durchführung modelliert wie von der Workflow Management Coalition (WfMC) definiert.

Das gesamte Wf-Management zerfällt somit in zwei Phasen: eine *Definitionsphase*, in der Prozess-Definitionen erstellt und bearbeitet werden, und eine *Ausführungsphase*, in der Prozesse instanziiert, ausgeführt und gegebenenfalls kontrolliert werden.¹

Eine genauere Beschreibung des Zusammenhangs zwischen Prozess-*Definitionen* wird im “Terminology & Glossary” Dokument² beschrieben. Zunächst wird pro Geschäftsprozessstyp eine Prozess-Definition erstellt. Für einen spezifischen aktuellen Geschäftsprozess wird diese Prozess-Definition dann instanziiert und ausgeführt. Änderungen werden traditionell nicht vorgesehen. Der Ablauf muss a-priori bekannt und in einer Prozess-Definition modelliert sein. Wenn überhaupt Änderungen möglich sind, dann nur bei der Prozess-Definition, was dann allerdings eine Änderung *aller* Instanzen zur Folge hat.

Es zeigte sich schnell, dass der konventionelle Ansatz zu starr war, um den Anforderungen der Realität bezüglich Dynamik und Flexibilität gerecht zu werden. Sich verändernde Rahmenbedingungen im Umfeld von Geschäftsprozessen erfordern die lang- und mittelfristige Evolution von Modellen. Weiterhin verlangen Ausnahmesituationen, Fehler und unvorhersehbare Ereignisse nach flexibler Handhabung auf Ebene der Prozessinstanzen. Es resultierte die Aufspaltung in drei grundsätzliche Herangehensweisen:

– **Ad-hoc Workflow**

I.a. existiert keine allgemeine Prozess-Definition, sondern bei Prozessbeginn kann der Bearbeiter eine Instanz durch Erzeugen einer elektronischen Umlaufmappe mit einer eingebauten Repräsentation der Routing-Logik erzeugen. Änderungen dieses Routings sind dem Instanzenbesitzer, allen am Umlauf-Beteiligten, oder definierten Beteiligten erlaubt.

– **Flexibler Workflow**

Erstellen einer Prozess-Definition vor der Ausführung, aber mit der Möglichkeit, die Prozesse und ihre Modelle während der Ausführung zu modifizieren. Folgende Ansätze lassen sich finden (siehe [18]):

- Die Prozess-Definition enthält offene Punkte (black boxes), deren genaue Ausgestaltung durch Nachmodellieren auf Instanzenebene zur Laufzeit geschieht (*open point approach*).
- Die Prozessdefinition enthält Beschreibungen bestimmter möglicher Ausnahmesituationen und Fehlerfälle (Ressourcen liegen nicht vor, ein Mitarbeiter nimmt eine Aktivität nicht an, ...) mit definierten Behebungsprozeduren (Verschieben der Aktivität, Delegation an anderen Mitarbeiter, Prozessänderungserlaubnis für bestimmte Personen, ...).

¹ *Wf Reference Model*, Seite 7 [<http://www.aiim.org/wfmc/standards/index.htm>]

² *Terminology & Glossary*, Seite 7 [<http://www.aiim.org/wfmc/mainframe.htm>]

- Bestimmte oder alle Prozessbeteiligte haben die Möglichkeit, auf Instanzen- bzw. Definitionsebene in Problemsituationen gewisse Änderungen mit einem vordefinierten Operatorvorrat (streiche Aktivität, ändere Reihenfolge, überspringe, ...) durchzuführen (*meta model approach*). Bei Änderungen der Definition muss die Konsistenz eventuell anderer laufender Instanzen sichergestellt werden.
- **Strukturierter Workflow**
Erstellen von vordefinierten Prozessen in reaktiv unveränderbaren Umgebungen mit relativ strenger Ausdruckskraft. Die Prozess-Modelle können nur als ganzes verändert werden. Einzelne laufende Prozesse sind nicht modifizierbar. Diese Definition entspricht dem klassischen Workflow-Konzept.

Nachdem wir soweit Grundansatz und aktuelle Hauptausprägungen der Wf-Idee kennengelernt haben, wollen wir im nächsten Abschnitt 2 eine Typisierung von Geschäftsprozessen und ihrer Teilaufgaben betrachten, sowie im darauffolgenden Abschnitt 3 entsprechende Merkmale und Elemente der Wissensarbeit untersuchen. In Abschnitt 4 wird die Eignung gewisser Geschäftsprozessklassen für die Unterstützung durch WfMSe beschrieben und die Klasse der wissensintensiven Geschäftsprozesse (WiGPe) eingeführt. Es folgt eine Gegenüberstellung der oben eingeführten drei Grundausprägungen des Wf-Supports in Abschnitt 5, denen wir wünschenswerte Eigenschaften einer Prozessunterstützung für Wissensarbeiter gegenüberstellen. Zur Umsetzung dieser Eigenschaften stellen wir in Abschnitt 6 einige Basis-Designentscheidungen für ein WfMS zur Unterstützung WiGPe vor und diskutieren einige Punkte zur weiteren Implementierung. Abschnitt 7 skizziert einige aktuelle verwandte Arbeiten, die Teile unseres Ansatzes bereits in irgendeiner Weise umsetzen, während wir schließlich in Abschnitt 8 kurz zusammenfassen.

2 Geschäftsprozess- und Aufgabentypen

Bevor man Anforderungen an die Rechnerunterstützung durch ein WfMS stellt, ist es zweckmäßig, zunächst einmal die Geschäftsprozesse selbst zu charakterisieren. Picot [25] klassifiziert Geschäftsprozesse anhand von 5 Kriterien, die er als "Prozessvariablen" bezeichnet (Abbildung 1).

Mit Hilfe dieser Prozessvariablen lassen sich drei idealtypische Prozesstypen (vgl. auch Abbildung 2-links), die Picot [25] folgendermaßen beschreiben:

- **Routineprozess:** Diesen Prozesstyp findet man typischerweise im Rechnungswesen, wo oft mehrere Menschen gemeinsam standardisierte Abläufe abarbeiten. Der Prozess hat eine klare, flache Struktur und behält diese auch weitestgehend auf Dauer.
- **Regelprozess:** Im Gegensatz zum Routineprozess bleibt die Struktur zwar noch kontrollierbar, ist aber häufigen Änderungen von Seiten der Mitarbeiter unterworfen. Der Ablauf eines Prozessmodells ist nicht mehr determiniert, wohl aber der *individueller* Prozesse. Beispiel: Auftragswesen des Anlagenbaus
- **Einmaliger Prozess:** Hier handelt es sich um typischen Vertreter von Management- oder Projektaufgaben. Weder Prozessablauf noch Kommunikationspartner sind konkret bestimmbar. Somit wird der Ablauf meist sehr *kommunikationsintensiv* durch einzelne Mitarbeiter oder Teams *individuell* durchgeführt. Offensichtlich sind diese Art von Prozessen kaum planbar und somit genauso wenig automatisierbar.

| Prozessvariable | Indikatoren |
|---------------------------|--|
| Komplexität | Zahl der Teilaufgaben Anordnung der Teilaufgaben (sequentiell, parallel) Abhängigkeiten/Rückkoppelungsbedarf der Teilaufgaben Rollen der in den Prozess involvierten Mitarbeiter |
| Grad der Veränderlichkeit | Wiederholungshäufigkeit ohne Strukturveränderungen Planbarkeit der Kommunikation während der Informationsbeschaffung Offenheit des Prozessergebnisses Änderungsanfall bedingt durch organisationsinterne/-externe Anforderungen |
| Detaillierungsgrad | Möglichkeit der Zerlegung des Gesamtprozesses in einfache Teilschritte Eindeutigkeit des erforderlichen Inputs, der Transformationsschritte und des Outputs |
| Grad der Arbeitsteilung | Anzahl der am Prozess beteiligten Mitarbeiter Koordinationsbedarf des Gesamtprozesses |
| Interprozessverflechtung | Schnittstellen zu anderen Prozessen Gemeinsame Datennutzung mit anderen Prozessen Prozesshierarchie (Beitrag des Prozesses zu über-, unter- oder nebengeordneten Prozessen) |

Abbildung 1. Kriterien und Indikatoren von Geschäftsprozessen, wie von [25] vorgestellt.

Ein Prozess setzt sich aus einer Vielzahl von Teilaufgaben zusammen. [25] teilt diese (Teil-) Aufgaben ebenfalls in drei Fälle ein (zur Charakterisierung eines Prozesses nach seinem Aufgabentyp nimmt man den im Prozess überwiegend vorherrschenden Typ):

- Der **Routinefall** ist gekennzeichnet durch seine hohe Planbarkeit bei niedriger Komplexität. Der Informationsbedarf ist ebenso gut bestimmt wie der Lösungsweg. Zur Lösung werden viele Hilfsmittel und genau festgelegte Kooperationspartner herangezogen, wobei die Kommunikationsintensität eher niedrig bleibt. Die Arbeit ist sehr stark dokumentorientiert.
- Beim **Sachbezogener Fall** erhöht sich bereits die Komplexität, während sich die Planbarkeit verringert und der Lösungsweg nunmehr nicht immer geregelt erscheint. Der Informationsbedarf und die Kooperationspartner sind nun nicht mehr eindeutig bestimmbar, weshalb auch das Arbeiten mit Dokumenten oft nicht mehr ausreicht.
- Der **Einzelfall** stellt für eine maschinelle Unterstützung eine Katastrophe dar, da hier hohe Komplexität mit niedriger Planbarkeit gepaart auftritt. Der Informationsbedarf sowie der Lösungsweg sind gänzlich unbestimmt. Die Kooperationspartner sind ebenfalls nicht festgelegt – nichtsdestotrotz handelt es hierbei aber um eine sehr kommunikationsintensive Arbeit unter Zuhilfenahme von wenigen Hilfsmitteln. Die Arbeit ist nicht mehr dokumentenorientiert.

Bevor wir auf die Eignung gewisser Prozesstyp-Aufgabentyp-Kombinationen für die WfMS-Unterstützung eingehen, wollen wir nun den Begriff des WiGPes etwas beleuchten.

3 Wissensintensive Geschäftsprozesse und -aufgaben

Selbstverständlich sind alle betrieblichen Tätigkeiten zu einem gewissen Anteil mit Wissen verbunden und erfordern mehr oder weniger Erfahrung im jeweiligen Tätigkeitsbereich.

Nichtsdestoweniger macht es Sinn, sich beim Start einer WM-Initiative im Unternehmen zuerst einmal auf einen (oder wenige) wesentliche Kernprozesse im Zentrum der betrieblichen Leistungserstellung zu konzentrieren, die in hohem Maße Fachwissen, trainiertes Entscheidungsvermögen, Informationsintegration aus vielen Quellen, Erfahrung in ähnlichen Fällen, etc. erfordern, so dass Wissen hier zum kritischen Erfolgsfaktor wird;³ Prozesse, deren Wissenorientierung oft geprägt ist durch Gebundenheit an wenige erfahrene Personen, Verteiltheit dieser Experten über große Entfernungen, Flüchtigkeit von Expertise und Entscheidungsbegründungen bei Mitarbeiterfluktuation und wegen mangelnder Dokumentation oder Dokumentierbarkeit usw.

Die Konzentration auf solche Prozesse bzw. Aufgaben in Prozessen lässt sich beispielsweise dadurch rechtfertigen, dass bei diesen Tätigkeiten oft der Leidensdruck genügend groß ist, um eine grundsätzliche Veränderung der Arbeitsweise angehen zu können; dass oft verhältnismäßig wenige Mitarbeiter involviert sind, mit einem relativ hohen Ausbildungsstand, was die Erfolgsaussichten eines Change Programms erhöht; dass solche Aufgaben häufig Gegenstand von Qualitätssicherungs- oder Zertifizierungsmaßnahmen sind, so dass man sich ohnehin einmal grundsätzlich mit ihnen beschäftigen muss; oder dass durch Qualitätsverbesserungen im Ergebnis, Fehlervermeidung oder Effizienzsteigerungen bei der Durchführung solcher Aufgaben durch ihre Schlüsselstellung im Betrieb große Hebelwirkungen erzielt werden können (beispielsweise kann keine Initiative zur Verringerung von Kodierfehlern bei der Software-Erstellung jemals so große positive Effekte zeitigen, wie die Verbesserung der Kommunikation und Dokumentation von Entwurfsentscheidungen in der Grobdesignphase des Softwareentwicklungsprozesses, wo schon kleine Fehler zu großen Effekten in der kompletten Produktentwicklung führen können).

Im Zuge unserer eigenen Industrieerfahrungen [20] zeigte sich die Notwendigkeit für WM-Funktionalitäten bei Anwendungsproblemstellungen wie technischen Entwurfsaufgaben / Produktentwicklung, bzw. Konfiguration von Anlagen, bei anderen Autoren wurden Anwendungsszenarien bei der Entscheidungsunterstützung [33], der strategischen Planung [13] oder im kreativen Design [16,29] untersucht. Ein weites Feld der Anwendung von WM, auch mit einer deutlichen Vorreiterrolle in der kommerziellen Nutzung, ist der Bereich Projektdurchführung / Projektmanagement bei Consulting-Aufgaben [23,8].

All diese Aufgabenbereiche sind keine typischen Anwendungsszenarien für Geschäftsprozess- oder WfMSe, man vermisst die wohlstrukturierten Abläufe, die klar erkennbaren, wiederholbaren Dokument- und Kommunikationsflüsse. Dennoch lässt sich die Notwendigkeit von WM-Maßnahmen unmittelbar einsehen; WM seinerseits kann aber auf vielfältige Weise synergetisch mit GPMSen und WFMSen verbunden werden (siehe [6]). Es stellt sich somit die Frage, ob sich die obengenannten Bereiche grundsätzlich jedem Prozessmanagement oder jeder Prozessunterstützung entziehen. Unseres Erachtens lässt sich dazu zumindest einmal folgendes beobachten:

1. Auch hochgradig unstrukturierte Aufgaben wie das kreative Design sind häufig eingebettet in durchaus konventionelle Prozesse, die zum Beispiel Rahmenvorgaben machen, Deadlines festsetzen, Kommunikationspartner festlegen, Dokumentationsformate und Ausgabedokumente vorschreiben, Input-Dokumente liefern usw. Die wissensintensive

³ Der initiale Fokus auf zentrale Aufgaben wird z.B. in der Know-Net Methode [9] in Stufe 1 'Strategische WM-Planung' als *make the KM business case* beschrieben, oder spiegelt sich in der CommonKADS-Methode [7] in der Machbarkeitsanalyse (*Worksheet OM-5: feasibility decision document*) wider.

Aufgabe taucht in diesem Fall als Black Box in einem gewöhnlichen Geschäftsprozess auf.

Es stellt sich die Frage, ob die Einbettung in einen umfassenden konventionellen Workflow nicht zu Verbesserungen führen kann. Im KnowMore-Projekt [3] beispielsweise wird zur Unterstützung solcher Aufgaben eine aktive Informationsbereitstellung aus dem Organisationsgedächtnis eingesetzt, welche Kontextparameter des Workflows zur Verbesserung des Information Retrieval benutzt.

2. Auch umgekehrt kann es vorkommen, dass innerhalb eines komplexen Ad-hoc-Prozesses, der sich selber der a-priori-Planung und Ablaufprogrammierung entzieht, durchaus stark strukturierte, eher konventionell angehbare Teilaufgaben oder Prozessanteile auftauchen. Man denke beispielsweise an das Einholen amtlicher Genehmigungen innerhalb eines großen Bau- oder Raumplanungsvorhabens, oder auch nur an das Arrangieren eines Meetings (Terminabsprachen, Einladungen, Agenda, Raum- und Hardwarereservierung, Catering, Reiseplanung, Unterkunft, Protokollführung, ...) innerhalb eines komplexen kollaborativen Prozesses. Hier entstehen konventionelle Wf-Bruchstücke innerhalb eines schwächer strukturierten umgebenden Prozesses.

In einer solchen Situation kann wiederum der umliegende schwach strukturierte Prozessanteil Input für den stark strukturierten liefern, bzw. seinen Output weiternutzen; ferner kommen natürlich die bekannten Wf-Nutzeffekte für den stark strukturierten Prozessanteil zum Tragen.

Man sieht also, dass der Einsatz von Wf-Technologien auch im Umfeld wissensintensiver Aufgaben durchaus Sinn macht. Bevor wir auf mögliche konkrete Ausprägungen der Unterstützung näher eingehen, wollen wir noch etwas auf die Charakteristika wissensintensiver Arbeit eingehen, wie sie in der Literatur diskutiert werden.

Rittels Begriff der ‘Wicked Problems‘

Schon in den 70er Jahren charakterisierte Rittel (siehe [27,28] und [13]) ‘Wissensarbeit‘ als primär durch den Umgang mit *wicked problems* (‘harte Problemstellungen‘, im Gegensatz zu *tame problems*, einfachen Problemen) geprägt. Einige Merkmale von wicked problems:

Problembeschreibung: Schon die klare Problemformulierung ist schwierig und i.a. nicht ohne Dissens aller beteiligter Parteien zu erreichen, z.B. hinsichtlich des adäquaten Abstraktionsniveaus zur Aufgabenbeschreibung. Im Kern läuft die Aufgabenstellung häufig auf ein komplexes System von Anliegen (Themen, Diskussionspunkten, Entscheidungsvariablen) und Randbedingungen (Restriktionen, Optimierungsgrößen, Constraints) hinaus. Dabei sind die Bedingungen bisweilen stark veränderlich über die Zeit hinweg, und auch Anliegen können sich ändern.

Problembearbeitung: Normalerweise sind viele Parteien mit ihren Interessen involviert, so dass die Lösungsfindung eher ein Aushandeln zur Erreichung eines akzeptablen Kompromisses ist als ein Problemlösen. Lösungen müssen mithin eher konstruiert oder entdeckt werden, als aus einer gegebenen Lösungsmenge ausgewählt. Häufig gibt es nicht richtige oder falsche Lösungen, sondern mehr oder weniger gute, oft sogar mit moralischen oder politischen Einflussfaktoren bei der Bewertung der Lösungsgüte. Folglich hat man oft keine klaren Erfolgskriterien, sondern beendet die Lösungsfindung, wenn keine Ressourcen mehr für die weitere Diskussion da sind.

Schon diese erste Grobcharakterisierung macht deutlich, dass ein Software-Support für solche Tätigkeiten eng mit Kommunikations- und Kollaborationssystemen gekoppelt sein sollte, dass die Dokumentation von Abläufen, Planungen und Entscheidungen mit ihren Ursachen nützlich wäre, dass eine Planungsunterstützung als ‘Nachdenkhilfe’ bei der Strukturierung des Problem- bzw. Lösungsraumes vonnöten ist, und dass eine optimierte Informationslogistik hilfreich ist, beispielsweise zum kontinuierlichen Monitoring externer Informationsquellen, um z.B. auf veränderte Rahmenbedingungen sofort aufmerksam zu werden.

Bearbeitung wissensintensiver Aufgaben

Aufbauend auf den Darstellungen von Kidd [19] und Buckingham Shum [29] – ähnliche Darstellungen findet man beispielsweise auch bei Davenport [17] – fassen wir typische Merkmale bei der Bearbeitung wissensintensiver Aufgaben zusammen:

Unvorhersehbare Verhaltensmuster: Vielfältige und stark ad-hoc entstehende Verhaltensmuster und Vorgehensweisen sind typisch. Dies schließt den Einsatz konventioneller Wf-Ansätze aus und verlangt nach individuell situativ definierbaren Prozessmustern. Diese würden immerhin Nachvollziehbarkeit unterstützen, könnten in ähnlichen Situationen zur Planung herangezogen werden und könnten auf einem gewissen Abstraktionsniveau zur Definition von sehr groben Standardvorgehensmustern bzw. zumindest Checklisten dienen.

Kommunikationsorientiert: Hoch variable Kommunikationsnetzwerke mit vielfältigem Einsatz von Medien sind typisch. Fachliche Inhalte sind wichtiger bei der Auswahl von Kooperationspartnern als starre betriebliche Aufbauorganisation. Wissen ist massiv in ausgetauschten Dokumenten, Mails, Entwürfen, etc. enthalten. Dies erfordert die Integration von Prozessmanagement und Kommunikationskanälen sowie die inhaltsorientierte Erschließung von Dokumenten und auch E-Mails für das Informationsmanagement. Mächtige Yellow-Page-Systeme sollten die Basis für die Zusammenstellung virtueller Teams sein.

Interdisziplinär: In zunehmendem Maße erfordert die Bearbeitung komplexer wissensintensiver Aufgaben die Expertise vieler Fachrichtungen und Unternehmensbereiche. Dies erfordert die schnelle Erfassbarkeit des aktuellen Prozessstatus für neu hinzukommende Team-Mitglieder sowie den einfachen Zugriff auf Hintergrund-Informationsmaterial zum Briefing in einem Thema noch unerfahrener Mitarbeiter/innen.

Informationslastig: Die – oftmals auch nur oberflächliche – Verarbeitung großer Mengen von Information und Dokumenten ist typisch. Automatische inhaltsorientierte Anbindung an Informations-, insbesondere Push-Dienste wären hier von Vorteil, ebenso die kontextuelle, automatisch indexierte Ablage von Information.

Argumentationsbasiert: Das Vorgehen ist ein kontinuierliches Argumentieren und Verhandeln entlang von Schlüsselangelegenheiten, Kernfragen, Handlungsoptionen und Optimierungskriterien. Ein System sollte es zumindest ermöglichen, die eigene Arbeit entlang solcher Grundkategorien zu planen, organisieren und auch zu dokumentieren. Dies würde u.a. die schrittweise Top-Down-Verfeinerung von Aktivitäten, oder auch die Terminierung von Aktionen umfassen.

Iterativ: Da der Problembearbeitungsprozess von Haus aus iterativ und inkrementell ist, muss eine Prozessbeschreibungssprache natürlich Zykel und ggf. spätere Verfeinerun-

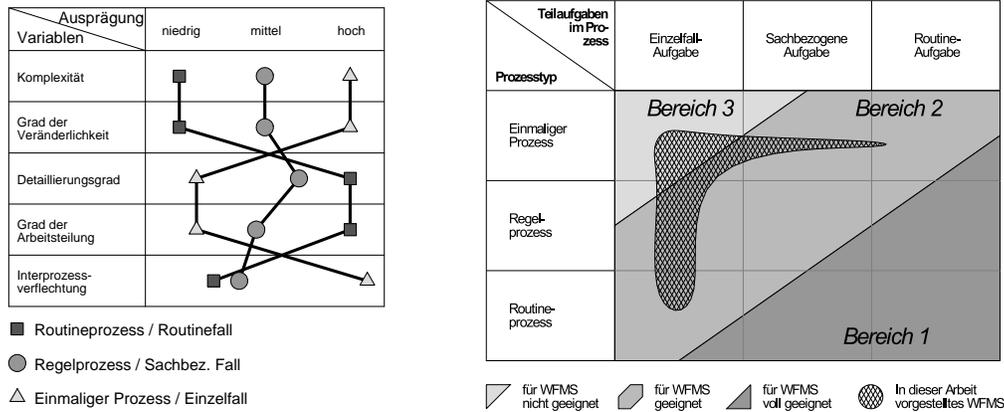


Abbildung 2. Links: Picots Charakterisierung verschiedener Geschäftsprozessstypen. Rechts: Eignung von Prozess-/Aufgabentypen für WfMSe [25].

gen früherer Prozessschritte erlauben. Der Zugriff auf Dokumente und Prozessabläufe in früheren Durchläufen wäre hier hilfreich.

4 Eignung von Prozess-/Aufgabentypen für WfMSe

Eine Untersuchung *aller* Kombinationsmöglichkeiten von Prozess- und Aufgabentypen liefert Hinweise auf die Eignung dieser Kombinationen für die Unterstützung durch ein WfMS. Picot [25] schlägt eine Einordnung dieser Kombinationen in drei Eignungsstufen vor (von nicht geeignet bis voll geeignet, siehe Abbildung 2-rechts). Dabei wurde das ursprüngliche Diagramm leicht modifiziert und zeigt unsere Einordnung des WfMSs zur Unterstützung WiGPe. Wie aus dem letzten Kapitel leicht nachvollziehbar, siedeln wir WiGPe primär im Bereich der Einzelfall-Aufgaben an. Fließende Übergänge zu Sach- und Routineaufgaben sowie in Regel- und Routineprozesse sind natürlich zu beobachten, wobei die Variabilität noch eher auf der Achse der Prozesscharakteristika als bei den Aufgabencharakteristika zu sehen ist, weil man wohl nur schwerlich aus reinen Routineaufgaben einen wissensintensiven Prozess wird zusammensetzen können.

Dabei kann Bereich 1 eine volle Eignung attestiert werden. Sowohl der Gesamtprozess als auch die Teilaufgaben können nach festen Regeln abgearbeitet werden. Eine Rechnerunterstützung zu diesem Bereich kann hervorragend in Form eines *transaktionsorientierten (konventionellen, strukturierten) Workflows* erfolgen.

Innerhalb des zweiten Bereiches kann den Mitarbeitern immer noch Unterstützung in Form von Dokumenten-Retrieval, E-Mail, Routineprüfungen etc. bereitgestellt werden, während wichtige Entscheidungen und der genaue Ablauf des Prozesses hauptsächlich in der direkten Verantwortung der Mitarbeiter selbst liegen. *Ad-hoc* und *Flexible Workflows* beispielsweise unterstützen diesen Bereich.

In Bereich 3 ist ein komplettes WfMS nur schwerlich anzusiedeln, da sowohl die Aufgabe als auch der Ablauf des Prozesses kaum vorherbestimmt werden können und fast gänzlich dem Mitarbeiter überlassen sind. Eine Rechnerunterstützung in diesem Bereich wird sich

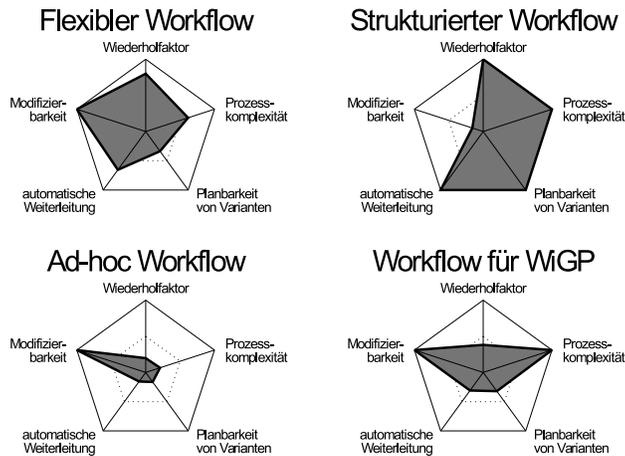


Abbildung 3. Eigenschaften von Workflow-Typen, inkl. der Unterstützung wissensintensiver Geschäftsprozesse.

daher hauptsächlich mit Informationsmanagement, Kommunikationsunterstützung und individuellen Planungs- und Projektierungsaufgaben begnügen. Die Funktion dieser Rechnerunterstützung entspricht weitestgehend einem Assistenten, der den Benutzer bei der Arbeit lediglich unterstützt, anstatt durch zwingende Vorschläge das Einhalten eines bestimmten Ablaufplans zu gewährleisten. In diesem Bereich ist die Unterstützung WiGPe vorwiegend anzusiedeln.

Im FRODO-Projekt [1] streben wir eine Unterstützung zur Planung und Durchführung sachlich komplexer Geschäftsprozesse mit geringer Häufigkeit in Form eines WfMSs an. Darüber hinaus sollen bereits durchgeführte Prozesse als Informationsquelle nutzbar gemacht werden, um damit zukünftige Geschäftsprozesse leichter planen und durchführen zu können. Die Unterstützung zielt also zunächst auf die Planung, Durchführung und Archivierung einmaliger Prozesse ab, allerdings mit dem Hintergedanken, dass archivierte, abgeschlossene Prozesse zu zukünftigen Prozessen zumindest *ähnlich* genug sind, dass eine Wiederverwendung nach leichten Modifikationen dennoch möglich ist.

5 Eigenschaften eines Workflow-Supports für WiGPe

Abbildung 3 beschreibt die drei in Kapitel 1 dargestellten Wf-Typen in fünf Dimensionen, wie sie von COI⁴ vorgeschlagen wurden. Die Abbildung wurde um ein weiteres Diagramm ergänzt, welches wünschenswerte Eigenschaften eines WfMS zur Unterstützung WiGPe skizziert. Der folgende Abschnitt 5 wird auf diesen vierten Wf-Typ und seine Schwerpunkte eingehen. Zunächst stellen wir die fünf Wf-Charakteristika vor:

Modifizierbarkeit. Wie oft, wie komfortabel und wie stark sind Änderungen möglich?

Hier geht auch ein, ob *jeder* Benutzer Änderungen vornehmen kann oder bloß bestimmte Administratoren.

⁴ <http://www.coi.de>

Wiederholfaktor. Wie oft können die Workflows wiederverwendet werden?

Müssen die Workflows bei der Wiederverwendung zuerst angepasst werden, verringert sich der Wiederholfaktor.

Prozesskomplexität. Können komplexe Arbeitsprozesse adäquat modelliert werden?

Ist die Modellierung beliebig verfeinerbar?

Sind komplexe Sachverhalte durch ausdrucksstarke Beschreibungen komfortabel modellierbar?

Planbarkeit von Varianten. In wie weit ist eine Wf-*Variante* auf der Basis eines anderen Workflows generierbar? Hier geht es nicht um die Möglichkeit, bereits bestehende Workflows zu verändern. Vielmehr geht es um die Frage, ob *zum Zeitpunkt der Erstellung* eines neuen Workflows, dieser auf der Basis eines anderen Workflows generiert werden kann und dieser strukturell tatsächlich als Variante geführt wird.

Automatische Weiterleitung. Werden den Benutzern *automatisch* Aufgaben zugeteilt?

Sind diese Aufgaben zwingend zu bearbeiten, oder stellen sie für den Benutzer bloß Vorschläge dar?

Unterstützung wissensintensiver Geschäftsprozesse

Unseres Erachtens sollten die Schwerpunkte zur Unterstützung WiGPe folgendermaßen verteilt sein:

Modifizierbarkeit: hoch. Jeder Benutzer kann *jederzeit alles* modifizieren und verfeinern. Das WfMS ist flexibel und dynamisch an neue entstehende Situationen anpassbar. Außerdem ermöglicht erst eine hohe Modifizierbarkeit, dass Prozesse zunächst grob vormodelliert werden und diese dann während der Durchführung nachmodelliert werden können. Dies entspricht dem ‘open point‘ Ansatz zur Flexibilität und macht nur Sinn, weil wir von hochqualifizierten ‘Power Usern‘ ausgehen können.

Wiederholfaktor: gering. Generell ist das WfMS zur Planung und Durchführung von individuellen Geschäftsprozessen gedacht. Daher ist die (exakte) Wiederholung von Prozessen nur von zweitrangiger Bedeutung. Dennoch kann ein gewisser Wiederholungsfaktor durch die Überführung von Prozess(teilen) in Modellen erreicht werden. Diese Modelle können sowohl bei der Generierung als auch während der Durchführung instanziiert in Prozesse eingebaut werden. Somit wird das Wiederverwenden von Prozess(teilen), die sich in der Vergangenheit bewährt haben, ermöglicht.

Prozesskomplexität: hoch. Sowohl komplexe Prozesse als auch tiefergehende Modellierung von Einzelfall-Aufgaben sollen unterstützt werden. Das WfMS bietet eine komplexe Modellierung erstens durch beliebige Top-Down-Verfeinerungsmöglichkeit und zweitens durch eine ausdrucksmächtige Sprache zur Beschreibung der Prozesslogik.

Planbarkeit von Varianten: gering. Das Erstellen von Wf-Varianten ist aufgrund der Fokussierung auf individuelle Prozesse von geringfügiger Bedeutung.

Nichtsdestotrotz stellt das Erstellen von Varianten mithilfe von Prozess-(teil-)Modellen kein Problem dar. Diese (Teil-)Modelle werden zwar instanziiert, die Änderungen werden aber danach direkt auf den Instanzen durchgeführt und nicht etwa auf den Ursprungsmodellen. Allerdings handelt sich anschaulich bei diesen “Varianten” eher um Workflows, die “ähnlich” zu dem ursprünglichen Workflow sind. Nach zu erwartenden häufigen Modifikationen wird der neue Workflow unter Umständen nur noch entfernt an den ursprünglichen erinnern.

Automatische Weiterleitung: gering. Das WfMS unterbreitet den Benutzern stets nur Vorschläge. Die Benutzer haben die volle Entscheidungsgewalt bezüglich der Prozessabläufe. Aufgrund der Natur der unterstützten Einzelfall-Aufgaben könnte das System in den meisten Fällen auch gar keine eigenständigen Entscheidungen treffen oder gar selbsttätig handeln. Das System wird den Benutzer agieren lassen und stellt somit ein Assistenz-System dar.

Es wird demnach direkt ersichtlich, dass die Schwerpunkte hier auf die Prozesskomplexität sowie die Flexibilität und Dynamik während der Durchführung gelegt wurden. Diese Charakterzüge verleihen dem Wf-Konzept die Fähigkeit, auch komplexe Projekte spontan modellieren und durchführen zu können, wobei jedoch mehr Strukturierungshilfen als beim reinen Ad-hoc-Workflow gegeben sind.

Diese groben Zielsetzungen führten zur Ableitung technischer Anforderungen an den Wf-Support, die wir im nächsten Kapitel darstellen.

6 Technischer Ansatz zur Unterstützung WiGPe

Im DFKI-Projekt KnowMore [2,3,4] wurde bereits der Ansatz verfolgt, eine Wissensmanagement-Unterstützung aktiv an ein WfMS anzubinden. Dies wurde erreicht durch kontextsensitives Bereitstellen von Informationen zu wissensintensiven Aktivitäten innerhalb eines klassischen Workflows. Ein Fokus des FRODO-Projekts [1] besteht nun in der Fortsetzung des KnowMore-Gedankens, allerdings mit dem Bestreben, den Übergang von der Koordination zur Kollaboration fließender zu gestalten (vgl. [31]). Da wissensintensive Aufgaben von Natur aus schwer planbar und vorhersehbar sind (siehe Abschnitt 3), könnte eine komfortable Unterstützung auf folgenden Design-Entscheidungen fußen:

Assistenz-Funktionalität. Das System soll den Benutzer bei seiner Wissensarbeit adäquat unterstützen, d.h. bei Planungs- und Organisations-, Kommunikations- und Verhandlungstätigkeiten. Das System soll dabei Vorschläge statt Befehle präsentieren.

Verzahnung von Modellierung und Ausführung. Die Modellierung sollte auch parallel zur Prozessausführung stattfinden können. (s.a. [24]).

Lazy/Late Modeling. Modellierung soll unvollständig begonnen und sukzessive verfeinert werden können. (s.a. [24]).

Dynamische hierarchische Dekomposition. Jede Aufgabe soll während der Durchführung rekursiv in Unteraufgaben aufgeteilt werden können. (s.a. [15]).

Ausdrucksmächtige Prozesslogik. Zur Spezifikation von Ablaufconstraints sollen komfortable Mittel zur Verfügung gestellt werden.

kontext-sensitive WM-Anbindung. Eine solche Anbindung wurde (für *klassische* Workflows) bereits in KnowMore realisiert. Jedoch die teilweise unterspezifizierte und sich entwickelnde Modellierung einer wissensintensiven Aktivität fordert besondere Aufmerksamkeit, da aus dieser nur wenig Hinweise auf relevante Informationen gezogen werden kann bzw. sich Hinweise erst zur Laufzeit ergeben.

Erste Experimente zur Umsetzung dieser Ideen wurden im Rahmen einer Diplomarbeit gemacht [30]. Ideen zur weiteren technischen Ausgestaltung folgen in den nächsten Unterkapiteln. Im darauffolgenden Kapitel besprechen wir dann kurz einige relevante Arbeiten anderer Gruppen.

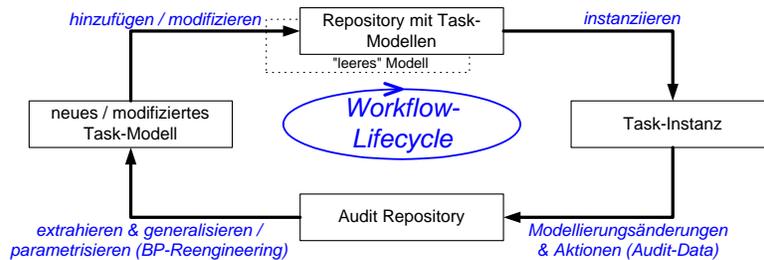


Abbildung4. Der Wf-Lifecycle.

6.1 Modelle und Instanzen

Aufgrund der hohen Flexibilitätsansprüche erscheint es wenig sinnvoll, den Lebenslauf eines Workflows (bzw. einer Wf-Instanz) in traditioneller Weise zu übernehmen. Statt wie gewohnt Wf-Modelle zu modellieren und diese dann für die durchzuführenden Aufgaben zu instanzieren (als Wf-Instanz), soll ein leicht modifizierter Weg beschritten werden:

Die Workflows werden zwar weiterhin durch Instanzieren von Wf-Modellen gebildet. Jedoch bleibt die Bindung zum ursprünglichen Wf-Modell nur sehr schwach. Änderungen werden direkt auf den Workflows und nicht auf Modellen ausgeführt, wodurch das *ursprüngliche* Modell im Extremfall nicht mehr mit der aktuellen Modellierung der Wf-Instanz verglichen werden kann. Neu zu modellierende Geschäftsprozesse basieren anfangs auf einem "leeren" Wf-Modell und werden danach direkt auf der erzeugten Instanz modelliert.

Um bereits modellierte Prozesse für zukünftige Prozesse wiederverwenden zu können, werden konsequent alle aktuell vorliegenden Modellierungen von Wf-Instanzen nebst entsprechenden Modifikationsprotokollen in einer Datenbank (Audit Repository) gehalten. In einer Reengineering-Phase wird zu einem gegebenen Zeitpunkt (durch einen versierten Anwender des WfMSs) eine Wf-Instanz zu einem Wf-Modell konvertiert. Es liegt nahe, dass dieser Vorgang einiges an Fachwissen voraussetzt, da hier Generalisierungen und vernünftige Parametrisierungen gefragt sind.

Das so erhaltene Wf-Modell beschreibt eine (generische) Standardvorgehensweise zu einer bestimmten Aufgabe und kann nun wiederum dem WfMS als neues auswählbares Modell zur Verfügung gestellt werden. Bei Bedarf wird dieses Modell instanziiert, gegebenenfalls auf neue Bedürfnisse angepasst, eventuell wieder parametrisiert und erneut in modifizierter Form dem System zur Verfügung gestellt, so dass nach vielfachem Durchlaufen dieses Zyklus nach und nach eine brauchbare Menge von Modellen entsteht. Dieser angesprochene Zyklus findet sich in Abbildung 4 wieder und beschreibt optisch exakt den gleichen Wf-Lebenszyklus, wie man es vom traditionellen Wf-Paradigma her kennt. Ein grundlegender Unterschied besteht jedoch darin, dass die Modellierung (i) auch *zur Laufzeit* und (ii) auf den *Wf-Instanzen* geschieht statt nur *a-priori* und/oder auf *Modellen*.

Da Prozesse oft Teile eines größeren Projekts und ebenso oft Teile eines Projekts wiederum Projekte sind, liegt es nahe, dass wir nicht bloß ganze Prozesse generalisieren und parametrisiert als Modelle für zukünftige Prozess-Instanzen zur Verfügung stellen, sondern

auch Teile davon. Gleiches gilt für die (instanziierte) Wiederverwendung von Prozess-Modellen für spezielle Teilaufgaben innerhalb einer Prozess-Instanz.

6.2 Prozesslogik

WiGPe lassen sich, aufgrund ihrer hohen Dynamik und schweren Planbarkeit, schlecht mittels traditioneller Prozesslogik beschreiben und ausführen. Es liegt daher nahe, sowohl die Spezifikation als auch die Art der Ausführung von WiGPe auf deren Beschaffenheit und Bedürfnisse hin anzupassen.

Bei WiGPe handelt es sich oft um Einzelfall-Aufgaben, wie sie auf Seite 4 beschrieben wurden. Diese Aufgaben sind typischerweise schwer planbar und von komplexer Natur. Daher soll bei der Modellierung einer solchen Aufgabe stets die Möglichkeit bestehen, beliebig tiefe Verfeinerungen einer Aufgabe in Unteraufgaben zu erlauben, was in einer hierarchischen Aufgabenstruktur resultiert.

Die Komplexität wissensintensiver Aufgaben schlägt sich allerdings auch auf das Vorkommen und Interagieren von oft sehr vielen Unteraufgaben nieder, weshalb eine zentral gehaltene, flache, beispielsweise flussdiagrammähnliche Strukturierung der Unteraufgaben (untereinander) von vornherein auszuschließen ist. Stattdessen erscheint es zweckmäßiger, die Beziehung zwischen Aufgaben eben in diesen Aufgaben selbst abzulegen, was bedeutet, dass man *nicht* die Reihenfolge von Aufgaben mittels einer direkten zeitlichen Ordnungsrelation (in planer Sichtweise) modelliert. Vielmehr werden in den Aufgaben selbst Vorgänger und Nachfolger in Form von Ablauf-*Constraints* spezifiziert. Die konkrete Abarbeitungsreihenfolge wird aus diesen Constraints dann inferiert. Zur Spezifikation und Steuerung der Ablaufconstraints können in den Aufgaben Attribute definiert werden, deren Werte in die Auswertung der Constraints eingehen. Die Constraints können somit aus Termen über Aufgabenzustände bestehen, wobei die Aufgabenzustände mittels der gespeicherten Attributwerte abgefragt werden können.

Es wird eine Reihe verschiedener Ablaufconstraints angeboten – mindestens die traditionellen Konstrukte wie *and-/or-fork* und *and-/or-join*. Darüber hinaus werden auch spezielle Konstrukte angeboten, die dynamische Anfragen an den aktuellen Kontext erlauben (bspw. die Verfügbarkeit bestimmter Ressourcen). Außerdem werden noch zeitliche Bedingungen wie “muss spätestens am 31.12.2001 abgeschlossen sein” angeboten, also zeitliche Constraints zur Spezifikation von Deadlines, Dauer und frühestmöglichem Beginn einer Aufgabe.

6.3 Ausführung

Die Geschäftsprozesse liegen also in Form von hierarchisch strukturierten Aufgaben vor, deren Ablaufbeziehung untereinander durch Ablaufconstraints spezifiziert ist. Diese Constraints werden zur Laufzeit ausgewertet um dem Benutzer aktuell durchführbare Aufgabenangebote zu unterbreiten. Gegebenenfalls kann sogar eine zukünftige Abarbeitungsreihenfolge von nachfolgenden Aufgaben inferiert und vorgeschlagen werden. Wohlgemerkt handelt es sich hierbei jedoch bloß um Vorschläge, nicht um Vorschriften.

Aufgrund seiner menschlichen Kompetenz oder gerade neu erworbenen Informationen kann der Benutzer diesen Vorschlag ignorieren und andere Entscheidungen treffen. Idealerweise würde der Benutzer eine begründbare Entscheidungsänderung damit untermauern,

dass er die entsprechende Aufgabe mit den zur Änderung geführten Faktoren nachmodelliert, so dass bei erneuter Inferenz der Ablaufconstraints nun der “richtige” Vorschlag vom System kommt.

In [30] haben wir die Ablaufbeziehungen verschiedener Aufgaben bereits durch Constraints in Form von Termen über Aufgabenattributen spezifiziert. Zu gegebenem Zeitpunkt wurden alle diese Constraints mittels eines Constraint-Solvers ausgewertet und damit sowohl eine *aktuelle* Menge von durchführbaren Aufgaben als auch eine *mögliche* Ablaufreihenfolge zukünftiger Nachfolge-Aufgaben inferiert. Letztere wurden dem Benutzer präsentiert, indem ein entsprechendes Gantt-Diagramm berechnet wurde, das dem Benutzer einen übersichtlichen Vorschlag zur günstigen, zukünftigen Ablaufreihenfolge in grafischer Form unterbreitet.

6.4 Engine und Plattform

Aufgrund hoher Anforderung an die Flexibilität und Dynamik des Systems bietet sich das Verwenden einer Agentenplattform für das “Enactment” des WfMS an. Ein weiterer Grund, autonome Softwareagenten ins Spiel zu bringen, ist die Tatsache, dass der Ablauf des gesamten Workflows nicht zentral (z.B. in Form eines Flussdiagrammes) vorliegt, sondern vielmehr verteilt innerhalb der einzelnen Aktivitäten der Prozesse. WiGPe bestehen aus vielen konkurrierenden Aktivitäten, deren idealer Ablauf nur schwer aus globaler Sicht erschlossen werden kann. Das Verwenden einer Agentenplattform zur Realisierung des kompletten WfMSs ermöglicht eine natürliche Abbildung dieser Aktivitäten auf eben solche (autonomen) konkurrierenden Agenten. Ferner erleichtert das Agentenparadigma eine modulare Entwicklung und Erweiterung des WfMSs.

7 Existierende Ansätze und Systeme

7.1 AIS Workware Demonstrator (AWD)

Bei dem AIS Workware Demonstrator [12,11] handelt es sich um ein Forschungsprojekt verschiedener norwegischer Forschungs- und Anwendungspartner, dessen Konzept nahezu identisch zu dem in dieser Arbeit vorgestellten Ansatz motiviert wird.

Ziele:

- Das System soll für normale Endbenutzer handhabbar sein.
- Integration von Planung und Ausführung: Prozessmodelle sind veränderbar und die Ausführung von unfertigen Modellen ist möglich.
- Es werden Prozess-*Instanzen* modelliert, nicht Prozess-*Definitionen*. Vielmehr werden die Instanzen später archiviert und können in Schablonen umgewandelt werden.
- Hierarchische Dekomposition: Jedes “Work item” (wissensintensive Aktivität) kann in mehrere “Sub Work items” dekomponiert werden.
- Deskriptive Modellierung statt präskriptiver Modellierung.
- Modellierung mittels APM (Action Port Model) [11].
- Archivieren von alten Varianten von Workflows mit Speicherung von Bemerkungen bezüglich der Änderungen. Dies ermöglicht ein Dokumentieren gemachter Erfahrungen von alten Wf-Phasen bis hin zu aktuellen Wf-Instanzen und -schablonen.

Bewertung: Die Zielsetzungen bei der Entwicklung des AIS Workware Demonstrator ist offensichtlich *sehr* ähnlich zu dem in dieser Arbeit vorgestellten Konzept. Dabei ist allerdings die Prozesslogik einfacher als in unserem Konzept. So sind keine komplexen Terme zur Ablaufsteuerung möglich, da lediglich die Operationen AND, OR und XOR bereitgestellt werden. Ferner ist beim AWD keine generische WM-Anbindung vorgesehen. Die Anbindung zielt stattdessen eher auf Informations- und Ressourcenmanagement ab. Perspektivisch interessant ist das Speichern von Erfahrungen während der Prozessausführung, wie auch das Speichern der alten Versionen (Entwicklungsphasen) eines Workflows. Dies könnte auch für die zukünftige Konzeptualisierung im FRODO-Projekt von Interesse sein.

7.2 Continuous Planning and Execution Framework (CPEF)

CPEF [22,10] realisiert eine intelligente Planung, Durchführung, Kontrolle und *Fehlerbehebung* von kontinuierlichen, komplexen Planungsaufgaben in unvorhersehbaren, dynamischen Umgebungen. Bei dem System handelt es sich um ein Forschungsprojekt von SRI International. Das Anwendungsszenario für dieses System sind militärische Luftschlachten.

Ziele:

- Integration von Wf-Management- und KI-Planungs-Komponenten. Dabei definieren die KI-Planungs-Komponenten eine intelligente, zielorientierte, ereignisgesteuerte, reaktive Kontrolle von Prozess-Aktivitäten.
- Die Komponenten agieren verteilt auf einer Multiagentenplattform. Dies ermöglicht eine fließende, einheitliche Integration der reaktiven Komponenten in das umgebende Wf-Konzept.
- Die Planung und Fehlerhebung wird durch Verwenden eines “Hierarchical Task Networks” (HTN) realisiert. Dies ermöglicht den flexiblen Aufbau von komplexen, detaillierten Plänen.
- Das System unterscheidet eine *direkte* und *indirekte Ausführung*. Als direkte Ausführung werden Aktivitäten bezeichnet, die vom System selbst unternommen werden können, während die indirekten Ausführungen alle diejenigen Aktivitäten bezeichnen, deren Ausführung vom System nur überwacht werden können. Bei letzteren handelt es sich um wissensintensive Aktivitäten, die typischerweise stark spontaner, dynamischer und (softwaretechnisch) unkontrollierbarer Natur sind.
- Pläne können aus archivierten Komponenten einer sogenannten “Prozedurbibliothek” zusammengestellt werden.

Bewertung: Bei CPEF lassen sich konzeptionelle Parallelen zu dem in dieser Arbeit vorgestellten Konzept entdecken, allerdings folgt man mit CPEF eher dem Drang nach einer Multiagenten-Architektur und der damit verbundenen Komponentenorientierung. Die fließende, einheitliche Integration von reaktiven Komponenten wird dadurch erleichtert. Ein Fokus von CPEF ist die Analyse und Behandlung von Planungsfehlern und die resultierende Umplanung. Dazu werden sowohl die Planung als auch die Behandlung von Planungsfehlern mittels HTN-Technologie (*hierarchical task networks*, eine Planungstechnik der Künstlichen Intelligenz) realisiert.

7.3 InConcert Wf

Bei InConcert [21] handelt es sich um ein Produkt einer Tochterfirma von Xerox, das eine Flexible-Wf-Architektur realisiert. Das System ist als aktuelle Version "InConcert 2000" ein lauffähiges, kommerzielles Produkt, das sich bereits auf dem Markt etabliert hat.

Ziele:

- Objektorientiertes Datenmodell, beliebige Verfeinerungsstufen durch hierarchische Dekomposition.
- Steuerungs- und Speicherungsmöglichkeiten innerhalb einer Aktivität durch "Attribute" und Referenzen auf externe Daten.
- Benutzer sind einer Menge von "User-Pools" zugeordnet, die den "Rollen" im klassischen Wf-Konzept entsprechen. Diese Rollen können für die Abarbeitung einer Aktivität gefordert werden.
- Vorbedingungen in den Aktivitäten steuern den Prozessablauf.
- Anbindung an Standardprogramme wie MS Project, MS Exchange etc.
- Bereitstellung der Dienste des WfMSs in Form einer Programmierschnittstelle (API).
- Client-Server-Architektur mit zentraler Datenhaltung im Server. Die Datenhaltung findet wiederum durch Kommunikation mit einem Datenbank-Server statt.
- Eine Prozess-Instanz kann von der zugehörigen Prozess-Definition abweichen. Allerdings ist dabei nur die Aktivität änderbar, nicht aber der Prozessablauf.
- Jede modellierte Prozess-Definition steht direkt als wiederbenutzbare Schablone zur Verfügung, und ebenso kann jede Prozess-Instanz als Basis neuer Prozesse benutzt werden.

Bewertung: Trotz der ähnlichen Ziele dieses Konzepts im Vergleich zu dem in dieser Arbeit vorgestellten Konzept, werden die Benutzer des InConcert-Systems, wie im klassischen Sinne, in zwei grobe Lager aufgespalten: diejenigen, welche die Aktivitäten einfach nur ausführen und diejenigen, welche die Ausführung der Aktivitäten ändern und kontrollieren können. Dies widerspricht dem großen Schwerpunkt in dieser Arbeit, dass *jeder* Benutzer des Systems seine Arbeit so frei wie möglich ausführen kann, so dass ihn das Benutzen des Systems hinsichtlich seiner Entscheidungsfreiheit weder einschränkt noch hindert. Des Weiteren ist eine WM-Unterstützung weder vorgesehen noch geplant.

7.4 OpenWater

OpenWater [36,35] ist ein von IBM entwickelter Forschungsprototyp. Motiviert wird die Entwicklung dieser Applikation durch Identifizieren der Schwächen traditioneller Workflows für Ad-hoc-Situationen und Ausnahmen. Eine Analyse vieler WiGPe ist aufwendig (teuer) und oft nicht machbar. Trotzdem wird dem "Prozesswissen" ein wichtiger Stellenwert bescheinigt. Ähnlich wie bei Wargitsch [34] werden daher Geschäftsprozesse als wesentlicher Bestandteil des organisatorischen Lernens auf verschiedenen Abstraktionsebenen erachtet. Das System besteht in Form einer Java-Applikation. Eine Anbindung an eine Datenbank ermöglicht eine zentrale Datenhaltung. Eine Zusammenarbeit von IBM und Lotus verspricht eine eventuelle Anbindung von OpenWater an Lotus Notes.

Ziele:

- Workflows müssen leicht modifizierbar sein — auch für Endbenutzer.
- Entwickle Wf-Definitions-Methode für Prozessevolution und Ausnahmenbehandlung ohne Notwendigkeit, formale Modelle zu bearbeiten.
- Deskriptiv statt präskriptiv: “Wf-Support” statt “Wf-Management”. Bei OpenWater entscheidet der Benutzer selbst über den Prozessablauf. Das Routing von Arbeitsanweisungen basiert auf einem “Electronic Circulation Folder” und verläuft konkret per E-Mail-Kommunikation.
- Dieses Routing soll durch Lernalgorithmen automatisiert und verbessert werden. Beispielsweise wird ein bestimmtes Routing vom System vorgeschlagen, falls der Nachfolger von früher bekannt ist.
- “Entdecke” Prozesse aus der Beobachtung von Ausführungen.
- Integriere das organisatorische Lernen aus Prozessausführungen.

Bewertung: Die Konzeption dieses Systems ist sehr stark ad-hoc-orientiert. Die Prozesslogik geht nur unwesentlich über ein vom Benutzer festgelegtes Routing von Prozessen hinaus. Eine ausdrucksmächtigere Steuerungsmöglichkeit des Ablaufs, wie beispielsweise die von uns angedeutete Vorbedingungssprache, wäre zu begrüßen.

Die Struktur von Prozessen und Daten ist flach realisiert und erlaubt somit weder eine hierarchische Gliederung noch eine Top-Down-Planung.

7.5 Plan Management Agent (PMA)

Der Plan Management Agent [26,32] zielt auf die Unterstützung *eines* Benutzers bei der Verwaltung einer potentiell großen und komplexen Menge von Plänen ab. Das System wurde von Martha E. Pollack und Ioannis Tsamardinos (University of Pittsburgh), sowie John F. Harty (University of Maryland) vorgestellt. Das System wird in Lisp (für Windows) implementiert, ist bereits lauffähig, befindet sich aber derzeit noch im Entwicklungsstatus.

Ziele:

- Einsetzen von KI-Technologie zur Modellierung von “Plänen”.
- Reasoning über ausdrucksstarke Pläne.
- Ausdrucksmächtige zeitliche Constraints, die Ausdrucksmöglichkeiten für zeitliche Ungewissheit beinhalten.
- Berechnung von Kosten für (1) die Erstellung eines neuen Planes und (2) das Ausführen eines Planes bezüglich des aktuellen Kontextes.
- Der Benutzer wird über nahende Deadlines informiert. Dabei wird beachtet, inwieweit der Zeitplan des Benutzers in Zukunft gefüllt ist. Im Falle eines zur Zeit unterbesetzten und in Zukunft dicht besetzten Zeitplanes schlägt das System bereits früh Aktionen vor, um die stark asymmetrisch vorliegende Arbeitsdichte etwas zu glätten.
- Dem Benutzer werden vom PMA Vorschläge unterbreitet. Die Entscheidung wird letztendlich jedoch stets dem Benutzer überlassen.
- Konflikte zwischen verschiedenen Plänen werden vom System erkannt. Das System schlägt dem Benutzer Lösungsmöglichkeiten zur Behandlung des Konfliktes vor, schafft aber selbstständig keine Abhilfe.
- Der Benutzer kann jederzeit Änderungen an den Plänen vornehmen.
- Bereits eingegebene Pläne oder Teile davon können in einer “Activity Library” zur Verfügung gestellt werden, um daraus neue Pläne zusammenzubauen.

Bewertung: Das System ist als Planungsunterstützung für einen einzelnen Benutzer konzipiert. Eine Kooperation mit anderen Mitarbeitern steht somit nicht im Vordergrund und wird auch nicht weiter beachtet. Gleiches gilt auch für eine WM-Anbindung.

Das System siedelt sich zwischen einfachen Kalender- / Erinnerungs-Tools und WfMSen an. Es sollen benutzerorientierte Pläne komfortabel und flexibel erstellt werden können. Dabei wird ein großer Schwerpunkt auf das Modellieren des zeitlichen Rahmens einer Aktivität (innerhalb eines Planes) gelegt.

Die *grundsätzliche* Konzeption von PMA wird bereits von der in dieser Arbeit vorgestellten subsummiert. Interessant sind aber die Betrachtung der kleinen Details, wie beispielsweise die Kostenabschätzung für die Durchführung eines Planes. Ferner ist im Ausblick des Dokuments [26] die Rede von einer Erfolgsabschätzung für ein Projekt unter Beachtung des aktuellen Zustands und der anstehenden Deadlines.

8 Zusammenfassung

Wir wollen kurz die Kernaussagen unseres Diskussionsbeitrages zusammenfassen: Wie wir gesehen haben, sind unterschiedliche Geschäftsprozesse durchaus verschiedener Natur und daher mehr oder weniger geeignet für gewisse Arten des Software-Supports. Wir haben einige Merkmale der Wissensarbeit besprochen und erkannt, dass aufgrund der hohen Flexibilitätsanforderungen wegen starker Einzelfallorientierung und oft einmaliger Prozessabläufe konventionelle Wf-Ansätze zum Scheitern verdammt sind. Andererseits ist grundsätzlich ein Wf-Support durchaus wünschenswert, schon wegen der Einbettung in stärker strukturierte Teile und der Dienste eines WfMSs für ein angebundenes WM-System. Als Zielsetzung wurde somit identifiziert, Systemdienste ‘an der Schwelle zwischen Ad-hoc- und Flexible-Workflow‘ auszugestalten. Wichtige Charakteristika unseres Vorschlags sind das hierarchisch dekomponierende Modellieren auf Instanzen und das Speichern dieser Instanzen zur späteren Generalisierung zu generischen Modellen (= Standardvorgehensweisen) auf einem höheren Abstraktionsniveau, sowie die nahtlose Einbettung konventioneller, stark strukturierter Konstrukte zur inkrementellen Spezifikation von Prozesslogik, die aber darüberhinaus auch beliebige Terme über Attributwerten aller modellierten Entitäten sowie Möglichkeiten zur Spezifikation zeitlicher Plandaten enthält. Für die weiteren Implementierungsarbeiten werden wir Agentenplattformen zur Realisierung des WfMS bzw. der Wf-Engine evaluieren, um eine nahtlose Integration der Wf-Dienste mit den Informationsagenten des WM-Systems zu gewährleisten.

Danksagung: Die vorliegenden Ausführungen wurden mit finanzieller Unterstützung des Bundesministeriums für Bildung und Forschung (Projekt FRODO – A Framework for Distributed Organizational Memories, Förderkennzeichen 01 IW 901) und der Europäischen Kommission (Projekt DECOR – Delivery of Context-Sensitive Organizational Knowledge, FKZ IST-1999-13002) erarbeitet.

Literatur

1. A. Abecker, A. Bernardi, A. Dengel, L. van Elst, M. Malburg, M. Sintek, S. Tabor, A. Weigel, and C. Wenzel. FRODO: A Framework for Distributed Organizational Memories. Project Proposal, DFKI GmbH Kaiserslautern, 2000. URL: <http://www.dfki.uni-kl.de/frodo/>.

2. A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a Technology for Organizational Memories. *IEEE Intelligent Systems*, June 1998.
3. A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project. *Int. Journal on Information Systems Frontiers (ISF), Special Issue on Knowledge Management and Organizational Memory, Kluwer*, 2(3/4), 2000.
4. A. Abecker, A. Bernardi, H. Maus, M. Sintek, and C. Wenzel. Information Supply for Business Processes – Coupling Workflow with Document Analysis and Information Retrieval. *Knowledge-Based Systems, Special Issue on AI in Knowledge Management, Elsevier*, 13(5), 2000.
5. A. Abecker, St. Decker, K. Hinkelmann, and U. Reimer. Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises, Freiburg, Germany. Document D-97-03, DFKI GmbH, September 1997.
6. A. Abecker, H. Maus, and A. Bernardi. Software-Unterstützung für das Geschäftsprozeß-orientierte Wissensmanagement: Versuch eines Überblicks. In H.J. Müller, A. Abecker, and H. Maus, editors, *Workshop Geschäftsprozeßorientiertes Wissensmanagement auf der WM'2001, Baden-Baden*, March 2001.
7. H. Akkermans, P.-H. Speel, and A. Ratcliffe. Problem, Opportunity, and Feasibility Analysis for Knowledge Management: An Industrial Case Study. In B.R. Gaines, M.A. Musen, and R.C. Kremer, editors, *12th Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Canada*, 1999.
8. D. Apostolou and G. Mentzas. Managing corporate knowledge: A comparative analysis of experiences in consulting firms. In *Proc. Second Int. Conference on Practical Aspects of Knowledge Management, PAKM-98, Basel, Switzerland*, 1998.
9. D. Apostolou, G. Mentzas, R. Young, and A. Abecker. Consolidating the Product Versus Process Controversy in Knowledge Management: The Know-Net Approach. In J. Domingue, editor, *PAKeM 2000, 3. Int. Conference and Exhibition on The Practical Application of Knowledge Management, Manchester, UK*, April 2000.
10. P. M. Berry and K. L. Myers. Adaptive Process Management: An AI Perspective. In *Towards Adaptive Workflow Systems*, 1998.
11. S. Carlsen. Action Port Model: A Mixed Paradigm Conceptual Workflow Modeling Language. In *3rd IFCIS Conference on cooperative Information systems - CoopIs '98, New York*, 1998.
12. S. Carlsen and H. Jorgensen. Emergent Workflow: The AIS Workware Demonstrator. In *ACM CSCW '98 Conference Workshop: Towards Adaptive Workflow Systems, Seattle*, 1998.
13. E. J. Conklin and W. Weil. Wicked problems: Naming the pain in organizations. White Paper of Group Decision Support Systems, Inc., URL: <http://www.gdss.com/wicked.htm>, 1997.
14. Consulting for Office und Information Management GmbH (COI). URL: http://www.coi.de/coiupdate/info/infomaterialien_de.asp.
15. N. Craven and D. Mahling. Goals and Processes: A Task Basis for Projects and Workflows. Technical report, Department of Information Science, University of Pittsburgh, May 1995.
16. M. Daniel and S. Decker *et al.* ERBUS - Towards a Knowledge Management System for Designers. In [5], 1997.
17. Th. Davenport, S.L. Jarvenpaa, and M.C. Beers. Improving knowledge work processes. *Sloan Management Review*, Reprint Series 37(4), Summer, 1996.
18. Y. Han, A. Sheth, and Ch. Bussler. A Taxonomy of Adaptive Workflow Management. In *CSCW-98 Workshop: Towards Adaptive Workflow Systems*, 1998.
19. A. Kidd. The marks are on the knowledge worker. In U. M. Borghoff and R. Pareschi, editors, *Proc. ACM CHI94: Human Factors in Computing Systems, Boston, Mass*, pages 186–191. ACM Press, 1994.

20. O. Kühn and A. Abecker. Corporate memories for knowledge management in industrial practice: Prospects and challenges. In U. M. Borghoff and R. Pareschi, editors, *Information Technology for Knowledge Management*. Springer, 1998.
21. R. T. Marshak. InConcert Workflow. Patricia Seybold's Workgroup Computing Report, http://www.inconcert.com/press/articles/iar_seybold.html, 1997.
22. K. L. Myers. Towards a Framework for Continuous Planning and Execution. In *AAAI Fall Symposium on Distributed Continual Planning*, 1998.
23. D. O'Leary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, May/June 1998.
24. C. Petrie, S. Goldmann, and A. Raquet. Agent-Based Project Management. Technical report, Center for Design Research, Stanford University, 1999.
25. A. Picot and P. Rohrbach. Organisatorische Aspekte von Workflow-Management-Systemen. *Information Management* 1/95, 1995.
26. M. E. Pollack, I. Tsamardinos, and J. F. Horty. Adjustable Autonomy for a Plan Management Agent. In *AAAI Spring Symposium on Adjustable Autonomy*, Stanford, CA, March 1999.
27. H.W.J. Rittel. *Second Generation Design Methods*. 1972.
28. H.W.J. Rittel and M.M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4, 1973.
29. S. B. Shum. Negotiating the construction and reconstruction of organisational memories. *Journal of Universal Computer Science*, 3(8):899–928, 1997.
30. S. Schwarz. Schwach strukturierte Workflows für das Wissensmanagement in Unternehmen. Master's thesis, FB Informatik, Uni Kaiserslautern, August 2000.
31. A. Sheth. From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Coordination and Collaboration. In *IEEE: 8. Int. Workshop on Database and Expert Systems Applications, Toulouse, France*, 1997.
32. I. Tsamardinos, M. E. Pollack, and J. F. Horty. Merging Plans with Quantitative Temporal Constraints, Temporally Extended Actions, and Conditional Branches. In *Proc. 5. Int. Conference on AI Planning Systems, Breckenridge, CO*, April 2000.
33. R. van Kaathoven, M. A. Jeusfeld, M. Staudt, and U. Reimer. Organizational memory supported workflow management. In A.-W. Scheer and M. Nüttgens, editors, *Electronic Business Engineering – 4. Int. Tagung Wirtschaftsinformatik, Saarbrücken, Germany*. Physica Verlag, Heidelberg, 1999.
34. Ch. Wargitsch. WorkBrain: Merging Organizational Memory and Workflow Management Systems. Technical report, Bavarian Research Center for Knowledge-Based Systems (FOR-WISS), 1997.
35. K. Whittingham. OpenWater White Paper. Technical report, IBM Research Division, Zurich Research Laboratory, 1999.
36. K. Whittingham, H. Ludwig, and M. Stolze. An alternative approach to business process support. In *CSCW-98 Workshop: Towards Adaptive Workflow Systems*, 1998.