Zeitplanung für komplexe wissensbasierte Arbeitsabläufe

Oliver Graeff

Diplomarbeit
Universität Kaiserslautern
September 1995

Betreuung:
Prof. Dr. Michael M. Richter
Dr. Frank Maurer

Erklärung

Hiermit versichere ich, daß ich diese Arbeit selbständig angefertigt und keine anderen Hilfsmittel, als die von mir angegebenen, verwendet habe.

Kaiserslautern, den 11. September 1995

Oliver Graeff

Inhaltsverzeichnis

Abbildungsverzeichnis

Nichts wird jemals im Rahmen des Terminplans oder Kostenvoranschlags gebaut.

Cheop's Gesetz in: Arthur Bloch, Murphy's Gesetz

Kapitel 1

Einleitung und Überblick

Das Thema dieser Arbeit ist die Zeitplanung in der Entwicklungsumgebung CoMo-Kit. Dieses einführende Kapitel beschreibt die Motivation, aus der sich die Aufgabenstellung zur Entwicklung einer Zeitplanungkomponente ableitet. Außerdem wird ein Überblick über diese Arbeit gegeben.

1.1 Einleitung

Bei vielen Arten von Entwicklungs- und Produktionsprozessen spielen Informationen eine immer größere Rolle. Sowohl die Informationsbeschaffung als auch die Bereitstellung der zu den Arbeitsabläufen benötigten Daten sind entscheidende Faktoren. Es muß festgelegt werden, welche Daten für welche Aufgaben benötigt werden und welche Ergebnisse die Lösung dieser Aufgaben liefert.

Sind bei einem verteilten System mehrere Personen beteiligt, so können die Aktivitäten auch parallel ausgeführt werden. Arbeitsabläufe müssen koordiniert und der Informationsfluß gesteuert werden. Die gleichzeitige Ausführung von Prozessen führt zu besonderen Anforderungen an die Zeitplanung. Beachtung erfordern vor allem die Abhängigkeiten zwischen den einzelnen (Teil-)Aufgaben [20].

Bei komplexen Aufgabestellungen fällt umfangreiches Wissen an, das erfaßt und gespeichert werden muß. Es werden Modelle von Arbeitsabläufen erstellt, um das vorhandene Wissen ablegen und wiederverwenden zu können.

Die Ablaufstrukturen der Informationsflüsse müssen beschrieben, koordiniert, geplant und weiterentwickelt werden. Auch während der Projektabwicklung kann es nötig werden, das Modell zu ändern, wenn z.B. bestimmte Teilaufgaben nicht gelöst werden können. Ziel ist es, solche Modelle mit einer Zeitplanungskomponente zu erfassen, um die einzelnen Aktivitäten zeitlich festzulegen.

In der Praxis von Entwicklungsprojekten bestehen häufig feste Termine für die Projektfertigstellung. Ein Ziel der Terminplanung ist die Verteilung des Aufwandes in dem zur Verfügung stehenden Zeitrahmen. Die erfolgreiche Allokation der Mittel resultiert in einer zeitlichen Festlegung der einzelnen Teilaufgaben, die das Projektziel erfüllt. Ein solcher korrekter Terminplan ist oft wichtiger als ein eingehaltener Kostenrahmen. Höhere Kosten können durch Preiserhöhungen oder längere Amortisationszeiten wieder wettgemacht werden. Ein überschrittener Zeitplan jedoch kann schwerwiegende Konsequenzen haben: Die Marktposition eines Unternehmens kann geschwächt werden, die Kunden werden unzufrieden und die internen Kosten steigen wegen des erhöhten Aufwandes [20].

Diese Überlegungen zeigen, daß der zeitlichen Planung und Kontrolle vernetzter Arbeitsabläufe eine große Bedeutung zukommt. Daraus ergibt sich das Thema dieser Diplomarbeit, deren Aufgabenstellung im folgenden Abschnitt vorgestellt wird.

1.2 Aufgabenstellung

CoMo-Kit [16] ist eine Entwicklungsumgebung für verteilte wissensbasierte Informationssysteme. Komplexe Aufgaben werden dargestellt, analysiert und in Teilaufgaben zerlegt. Dabei entsteht ein Informationsflußgraph, der bei der Projektabwicklung abgearbeitet wird.

Das Ziel dieser Arbeit ist die Erweiterung von CoMo-Kit um eine Zeitplanungskomponente. Man möchte Informationen über die Dauer der einzelnen Teilaufgaben gewinnen und diese zu Dauern der übergeordneten abstrakten Aufgaben abschätzen. Auf diese Weise kann sich ein Manager über die voraussichtliche Gesamtdauer des Projektes und dessen Fortgang informieren, ähnlich wie dies bei Meilensteinen möglich ist.

Dazu sollen frühestmögliche und spätestmögliche Anfangs- und Endzeitpunkte der Vorgänge berechnet werden. Die Differenz zwischen frühesten und spätesten Zeitpunkten gibt dem Projektmanager eine Aussage über den Dispositionsspielraum der einzelnen Vorgänge. Ist kein Spielraum vorhanden, kann man erkennen, daß dieser Vorgang die Projektdauer determiniert und eventuell für Verzögerungen verantwortlich ist.

Um diese Anforderungen zu erfüllen habe ich im Rahmen dieser Diplomarbeit das Verfahren der Allgemeine Hierarchischen Netzplantechnik entwickelt und als Zeitplanungskomponente in das System CoMo-Kit integriert.

Abgrenzung zu anderen Arbeiten

Um diese Diplomarbeit von anderen Arbeiten abzugrenzen, werden die folgenden zugrundeliegenden Annahmen herausgestellt:

- Es wird davon ausgegangen, daß die zur Projektdurchführung benötigten Ressourcen in unbeschränkter Menge zur Verfügung stehen. In zahlreichen Arbeiten werden die Probleme behandelt, die durch knappe Ressourcen entstehen [12]. Ihre Allokation wird unter den Gesichtspunkten der minimalen Kosten oder der minimalen Projektdauer betrachtet.

- Die Vorgangsdauern der Aufgaben werden in dieser Arbeit als gegeben angesehen. Im Rahmen von CoMo-Kit gibt es zur Zeit keine Erfahrungen über die Dauer der modellierten Vorgänge, sodaß keine Informationen über die stochastische Verteilung der Werte vorhanden ist. Sind diese Erfahrungen vorhanden, können sie in anderen Berechnungsverfahren ausgenutzt werden [12, 27, 30, 25].

  Weitere Informationen können in Form von Risikofaktoren in die Berechnung einfließen. Sie beschreiben Faktoren, die sich auf beliebige Attribute des Netzplanes auswirken. Damit kann die kausale Abhängigkeit einer Vorgangsdauer von äußeren Einflüssen durch eine Regel abgebildet werden. Erweiterungen der Netzplantechnik um deklaratives Wissen sind in [29] beschrieben.

- Um den Netzplan zu vereinfachen können unabhängige Teilnetze zu einzelnen Knoten reduziert werden. Einige Verfahren wenden diese Technik an, um zu abstrakteren Netzplänen zu gelangen, in denen die Zeitanalyse vereinfacht werden kann [14, 15, 22, 18, 27, 2]. Dabei werden feine Netzpläne derart zu gröberen zusammengefaßt, daß die Zeitanalyse von der feinen Ebene korrekt auf die grobe Ebene abgebildet wird.

  Bei Como-Kit findet dagegen eine zunehmende Verfeinerung der abstrakten Aufgaben statt. Der Benutzer achtet dabei nicht auf die Zusammenhänge zwischen den Abstraktionebenen, die eine konsistente Zeitplanung ermöglichen würden. Die Teilnetze können nicht problemlos reduziert und nicht unabhängig voneinander betrachtet werden.

Detailkenntnisse des Systems CoMo-Kit sind für das Verständnis dieser Arbeit nicht notwendig. In Kapitel 3 wird eine kurze Einführung gegeben.

1.3 Überblick

Dies ist eine Übersicht über die Kapitel dieser Arbeit.

Kapitel 2: Das zweiten Kapitel stellt die traditionellen Netzplantechniken vor. Diese hauptsächlich in den 50'er und 60'er Jahren entwickelten Verfahren bilden die Basis, auf die die Allgemeine Hierarchische Netzplantechnik aufbaut.

Zunächst werden allgemeine Grundlagen der Netzplantechnik vorgestellt, die alle Verfahren gemeinsam kennzeichnen. Die folgenden Abschnitte beschreiben die einzelnen Verfahren, die nach drei Grundkonzepten zusammengefaßt sind:

1. Die deterministischen Verfahren, die von einer festgelegten Struktur des Projektes ausgehen,

2. die entscheidungsbasierten Techniken, die verschiedene alternative Durchführungsmöglichkeiten abbilden können und

3. die hierarchischen Netzpläne, die die Hierarchie einer verfeinerten Projektzerlegung beschreiben.

Bei jedem Verfahren sind die Darstellungsmächtigkeit und die Prinzipien der Zeitanalyse angegeben.

Kapitel 3: Dieses Kapitel gibt eine kurze Einführung in das Modellierungswerkzeug CoMo-Kit. Insbesondere wird auf das konzeptuelle Modell eingegangen, das die Grundstruktur für die Zeitanalyse darstellt. Die Beriffe 'Aufgabe' und 'Methode' werden vorgestellt, da sie die zentrale Rolle bei der Abbildung von konzeptuellen Modellen auf Netzpläne spielen.

Außerdem wird auf den Interpreter von CoMo-Kit eingegangen. Während der Projektabwicklung stellt er Informationen zur Verfügung, die von der Zeitplanungskomponente dazu benutzt werden, die Zeitinformationen zu aktualisieren.

Kapitel 4: In diesem Kapitel werden die Mängel genannt, die die traditionellen Verfahren im Hinblick auf CoMo-Kit haben. Keines der bekannten Verfahren läßt sich direkt auf konzeptuelle Modelle anwenden. CoMo-Kit läßt eine beliebige Zerlegung der Aufgabe in Teilaufgaben zu und ermöglicht umfangreiche Abhängigkeitsbeziehungen zwischen allen Teilen des Modells. Diese Möglichkeiten werden von den traditionellen Verfahren nicht erfaßt.

Kapitel 5: Das von mir entwickelte Verfahren der Allgemeinen Hierarchischen Netzplantechnik wird in diesem Kapitel behandelt. Zusammen mit dem folgenden Kapitel enthält es die zentralen Resultate dieser Diplomarbeit.
Es ist beschrieben, wie ein konzeptuelles Modell auf einen Netzplan abgebildet wird und in welchen Teilen des Modells die Zeitpropagierung stattfindet. Die Menge der in den Knoten abgebildeten Informationen wird erweitert, um zusätzliche Zeitdaten zu erfassen. Anschließend beschreibt das Kapitel die eigentliche Zeitanalyse. Für jeden Vorgang soll ein zeitlicher Dispositionsspielraum festgelegt werden. Dazu sind nacheinander drei Phasen durchzuführen:

1. Die progressive Zeitanalyse zur Ermittlung der frühestmöglichen zeitlichen Lage der einzelnen Vorgänge,

2. die retrograde Zeitanalyse zur Ermittlung der spätestmöglichen zeitlichen Lage und

3. die Verallgemeinerung der so berechneten Zeitinformationen auf abstraktere Ebenen des konzeptuellen Modells.

Kapitel 6: Dieses Kapitel beschreibt Erweiterungen der Netzplankomponente, um die Signale des Interpreters während der Projektabwicklung zu verarbeiten.

Außerdem wird eine größere Flexibilität bei der Angabe von Vorgangsdauern zugelassen.

Kapitel 7: Die konkrete Realisierung der Zeitplanungskomponente ist Thema dieses Kapitels. An einem Beispiel wird die Anwendung veranschaulicht.

Kapitel 8: Das letzte Kapitel faßt die Resultate zusammen und gibt einen Ausblick auf mögliche Erweiterungen und weitere Anwendungen beim Zeitmanagement.

Kapitel 2

Traditionelle Netzplantechnik

In diesem Kapitel werden die traditionellen Verfahren der Netzplantechnik vorgestellt. Sie wurden seit den 50'er und 60'er Jahren entwickelt, um eine flexible Zeitplanung in komplexen Projekten zu ermöglichen.

In den ersten beiden Abschnitten werden die Verwendung von Netzplänen und ihre grundlegenden Konzepte erläutert. Die einzelnen Verfahren der Netzplantechnik unterscheiden sich einerseits durch die verwendete Darstellungsart des Projektnetzplanes und andererseits durch die Berechnungsmethodik. Die Methoden berücksichtigen ein unterschiedliches Maß an Unsicherheiten beim Projektablauf und beim Bedarf der einzelnen Vorgänge an Zeit, Kosten und Betriebsmitteln [9].

Um einen Überblick über die traditionellen Verfahren zu geben, werden in den darauf folgenden drei Abschnitten die wichtigsten vorgestellt. Sie sind dabei in deterministische, entscheidungsbasierte und hierarchische Netzplantechniken unterteilt. Die charakteristischen Eigenschaften der Verfahren werden gegenübergestellt, um sie von der Allgemeinen Hierarchischen Netzplantechnik abzugrenzen.

Zusätzliche Erweiterungen der Netzplantechnik beziehen sich auf die Einbeziehung von Kosten- und Kapazitätsgesichtspunkten. Diese Aspekte sind bei CoMo-Kit zur Zeit noch nicht berücksichtigt. Deshalb beschränkt sich diese Arbeit auf die Zeitaspekte.

2.1 Planung und Kontrolle mit Netzplänen

Jeder Ablauf muß in irgendeiner Form vor seiner Durchführung geplant werden. Unter **Planung** wird in dieser Arbeit die Ordnung, Vorbereitung oder gedankliche Vorwegnahme von Aktivitäten verstanden, um Aufgaben und Ziele sicher und ohne Umwege zu erreichen [25].

Komplexe Abläufe, im folgenden Projekt genannt, erfordern eine besonders sorgfältige Planung. Ein **Projekt** im Sinne der Netzplantechnik bezeichnet ein zeitlich, räumlich und sachlich begrenztes komplexes Arbeitsvorhaben, bei dem durch den Einsatz von Gütern und Arbeitskräften eine bestimmte Zielsetzung (Aufgabe) zu erreichen ist [25].

Dazu ist eine exakte graphische und rechnerunterstützte Planungs-, Überwachungs- und Steuerungsmethode erforderlich, die den Projektmanager unterstützt. Ein älteres Planungsverfahren ist z.B. der Gantt-Chart: Auf der horizontalen Achse wird die Zeit abgetragen und auf der vertikalen Achse werden die Arbeitsvorgänge aufgezeichnet. Abläufe sind anschaulich durch horizontale Balken dargestellt. Es werden dabei nur zeitliche, jedoch keine kausalen Abhängigkeiten verwaltet. Da Gantt-Charts den eigentliche Planungsvorgang deshalb nur mangelhaft unterstützen wird, dienen sie heute lediglich der Ergebnisdarstellung.

In den 50'er Jahren wurde die Notwendigkeit eines neuen Planungsverfahrens erkannt. Es sollte die folgenden Anforderungen erfüllen, die die bis dahin bekannten Verfahren nicht berücksichtigten [10, 25]:

- Übersichtliche Darstellung eines komplexen Projektablaufs (inkl. Vorgänge und deren Abhängigkeiten)
- Ermittlung der kritischen Stellen, der Zeitreserven und der Optimierungsmöglichkeiten
- schnelle Konsequenzenanalyse bei auftretenden Schwierigkeiten oder Situationsveränderungen
- Flexibilität bei Änderungen
- Einsatzmöglichkeit für EDV
- Wirtschaftlichkeit

Durch die Einführung der Netzplantechnik konnte die Darstellung um die kausalen Zusammenhänge erweitert werden. Damit konnten mehr Informationen über die Struktur eines Projektes in den Netzplänen abgebildet werden. Die sogenannte **Strukturanalyse** beinhaltet die Untersuchung der Reihenfolge und der gegenseitigen Beziehungen aller Vorgänge. Sie bildet die Grundlage für die anschließend durchgeführte **Zeitanalyse**.

Mit Hilfe der Zeitplanung der Netzplantechnik werden jedem einzelnen Arbeitsvorgang Anfangs- und Endzeitpunkte zugeordnet. Zeitliche Spielräume sind abgebildet, indem diese Zeitpunkte in den Schranken [frühestmöglicher Zeitpunkt, spätestmöglicher Zeitpunkt] angegeben werden. Eine Komponente zur Zeitkontrolle vergleicht während der Projektabwicklung die tatsächlichen Zeitpunkte mit den geplanten Zeitpunkten. Wird dabei z.B. erkannt, daß das Projekt zu spät fertig werden würde, weil etwa der tatsächliche Endzeitpunkt hinter dem geplanten Endzeitpunkt liegt, so muß umgeplant werden.

Die in der Netzplantechnik abgebildeten kausalen Abhängigkeiten werden benutzt, um die Umplanung zu unterstützen: Der Planer hat einen Überblick über den gesamten Projektablauf und kann Kürzungen der Projektdauer vornehmen. Man berechnet den 'kritischen Weg', d.h. den zeitlängsten Weg durch das Netz, der die Mindestdauer des Projektes bestimmt. Die Vorgänge auf diesem Pfad lassen sich kürzen, indem z.B. mehr Mittel eingesetzt werden, um die Vorgänge zu beschleunigen.

Durch die Anwendung der Netzplantechnik können also effizient und komfortabel komplexe Projekte präziser als mit Gantt-Charts und ähnlichen Verfahren geplant werden. Insbesondere durch Computerunterstützung gelingt es, bei umfangreichen Projekten die Lücke zwischen Plan und Realität zu verkleinern.

2.2 Grundlagen der Netzplantechnik

Im ersten Teil dieses Abschnitts werden die grundlegenden Konzepte und die Gemeinsamkeiten der verschiedenen Netzplantechniken erläutert. Diese Gemeinsamkeiten betreffen die verwendeten Begriffe und die Grundelemente der Netzpläne. Im folgenden Teil sind die verschiedenen Darstellungsformen vorgestellt. Anschließend wird die Projektanalyse erläutert, die sich in zwei Phasen vollzieht: 1. der Strukturanalyse und 2. der Zeitanalyse [30].

2.2.1 Grundkonzepte der Netzplantechnik

Die grundlegende Begriffe der Netzplantechnik sind:

Projekt: Als Projekt wird das zu planende und auszuführende Vorhaben bezeichnet. Charakteristisch für die betrachteten Projekte sind die hohe Komplexität und die vielen miteinander verbundenen Arbeitsgänge. Dadurch wird eine exakte und graphisch repräsentierbare Planungs- und Steuerungsmethode erforderlich.

Ereignis: Ein Ereignis beschreibt das Eintreten eines definierten Zustandes im Ablauf eines Projektes. Ereignisse haben keine zeitliche Ausdehnung und beschreiben Zeitpunkte, zu denen Vorgänge beginnen oder enden können.

Vorgang (auch: Aktivität, Tätigkeit): Als Vorgänge werden die Teilaufgaben bezeichnet, in die ein Projekt zerfällt. Dabei kann es sich sowohl um Vorgänge handeln, die eine menschliche Tätigkeit beinhalten, als auch um solche, die automatisch abgearbeitet werden oder nur eine Wartezeit repräsentieren. Entscheidend für die Netzplantechnik sind hier die definierten Anfangs- und Endzeitpunkte des Vorgangs, die zusätzlich noch nach frühestmöglichem und spätestmöglichem Zeitpunkt unterschieden werden. Es ergeben sich also vier Termine, die bei jedem Vorgang i angegeben sind [1]:

[1] In dieser Arbeit wird angenommen, daß die Knoten eines Netzplanes in einer beliebigen Reihenfolge durchnummeriert sind.

FAZ_i: Frühester Anfangszeitpunkt des Vorgangs i
FEZ_i: Frühester Endzeitpunkt des Vorgangs i
SAZ_i: Spätester Anfangszeitpunkt des Vorgangs i
SEZ_i: Spätester Endzeitpunkt des Vorgangs i

Vorgangsdauer: Die Vorgangsdauer d_i bezeichnet die zeitliche Ausdehnung des Vorgangs i und bildet die Grundlage für die Zeitplanung. Die Vorgangsdauern sind naturgemäß Schätzwerte, denen im günstigsten Fall umfangreiche Erfahrungen aus der Vergangenheit zugrunde liegen. Verbindliche Angaben sind im Planungsstadium aber in keinem Fall möglich.

Eine zusätzliche Schwierigkeit besteht darin, daß die Vorgangsdauer von der jeweils eingesetzten Kapazität abhängig und damit variabel ist. Um die Vorgangsdauern möglichst exakt zu schätzen, müssen daher die für diese Tätigkeit eingesetzten Mittel festgelegt sein. Durch z.B. einen höheren Arbeitskräfteeinsatz kann oft eine Beschleunigung des Vorgangs erreicht werden [1]. Sind hier sehr große Spielräume gegeben, so könnte man die Vorgangsdauer auch als Funktion des Aufwandes (z.B. 'Personen-Tage') oder sogar als Funktion des Kosten angeben.

Außerdem sollten die Zeitschätzungen unabhängig von einschränkenden Terminvorstellungen gemacht werden. Andernfalls besteht die Gefahr, daß sich die Schätzungen an den Terminvorstellungen orientieren und dadurch Verzerrungen und Fehler hervorgerufen werden [25].

In dieser Arbeit sehen wir die Vorgangsdauern, d.h. die Differenz zwischen Anfangs- und Endzeitpunkt eines Vorgangs, als gegeben an.

Netzplan: Ein Netzplan ist ein bewerteter azyklischer zusammenhängender gerichteter Graph [25]. Ein Netzplan besteht also aus Knoten, die durch Pfeile miteinander verbunden sind. Die Pfeile können mit Vorgangsdauern bewertet sein. Im folgenden Abschnitt wird erläutert, wie die Elemente eines Projektes (Vorgänge, Ereignisse und deren Abhängigkeiten) durch Elemente eines Netzplanes dargestellt werden.

Startknoten: Ein Startknoten eines Netzplanes ist ein Knoten, der keinen Vorgänger hat.

Endknoten: Ein Endknoten eines Netzplanes ist ein Knoten, der keinen Nachfolger hat.

geschlossener Netzplan: Ein Netzplan ist geschlossen, wenn er genau einen Startknoten und einen Endknoten hat.

offener Netzplan: Ein Netzplan ist offen, wenn er mehrere Start- und/oder mehrere Endknoten hat.

2.2.2 Darstellung der Netzpläne

Der Netzplan ist die Darstellung der einzelnen Vorgänge des Planungsvorhabens und ihrer Abhängigkeiten. Dies kann in tabellarischer Form geschehen; wegen der besseren Übersichtlichkeit wird jedoch meist die graphische Darstellung verwendet. Die Knoten eines Netzplanes werden meist als Rechtecke gezeichnet. In Anlehnung an Gantt-Charts läßt sich in die Rechtecke ein zeitlicher Verlauf von der linken zur rechten Begrenzungslinie interpretieren.

Die Elemente eines Projektes, also die Vorgänge und die Ereignisse, müssen den Elementen eines Netzplanes, also den Knoten und den Kanten, zugeordnet werden. Dafür gibt es zwei Möglichkeiten [30, 25]:

- Knotendarstellung: In dieser Darstellung werden die Vorgänge durch Knoten dargestellt. Ereignisse werden nicht explizit abgebildet. Die gerichteten Kanten geben dann Zusammenhänge zwischen den Vorgängen an. Die Arten von möglichen Zusammenhänge werden im Abschnitt 2.2.3 definiert.
- Pfeildarstellung: In dieser Darstellung werden die Vorgänge durch die gerichteten Kanten und die Ereignisse durch die Knoten dargestellt.

Eine weitere Unterscheidung bezüglich der Darstellung betrifft die Hervorhebung bestimmter Aspekte des Netzplanes.

- Vorgangsorientierte Netzpläne: Ist man insbesondere an der Beschreibung der Vorgänge eines Projektes in Bezug auf deren Inhalt und Dauer interessiert, so ergeben sich die Ereignisse automatisch als Anfangs- bzw. Endzeitpunkte der Vorgänge. Sie werden implizit durch die linke und rechte Begrenzungslinie des Knotens bzw. durch den Ursprung und das Ziel einer gerichteten Kante dargestellt. In diesem Sinne treten die Ereignisse bei der graphischen Darstellung in den Hintergrund.

  In der allgemeinen Praxis und auch in dieser Arbeit werden vorgangsorientierte Netzpläne benutzt.
- Ereignisorientierte Netzpläne: Möchte man dagegen vor allem die Ereignisse bezüglich ihres Inhalts beschreiben oder definieren, so bleiben die Vorgänge, die zu den einzelnen Ereignissen führen, im wesentlichen undefiniert. Diese deutlichere Darstellung der Ereignisse (bzw. Meilensteine oder Kontrollpunkte) eines Projektes eignet sich für die Bedürfnisse der Projektleitung. Sie ist vor allem an dem Projektfortschritt und den erreichten Meilensteinen interessiert.

Im deutschen Sprachgebrauch hat man die möglichen Darstellungs- und Orientierungsarten zu drei Netzplantypen kombiniert, die bei der Vorstellung der einzelnen Verfahren verwendet werden:

Vorgangsknotennetz: ein vorgangsorientierter Netzplan in Knotendarstellung

Vorgangspfeilnetz: ein vorgangsorientierter Netzplan in Pfeildarstellung

Ereignisknotennetz: ein ereignisorientierter Netzplan in Pfeildarstellung

Heute werden in der Praxis weitestgehend Vorgangsknotennetze verwendet. Auch für CoMo-Kit mit seiner Darstellung von Aufgaben und Datenflüssen eignet sich dieser Netzplantyp. In dieser Arbeit werden deshalb ausschließlich Vorgangsknotennetze betrachtet.

2.2.3 Strukturanalyse

Die Strukturanalyse beinhaltet die Untersuchung der Reihenfolge und der gegenseitigen Beziehungen aller Vorgänge. Dazu muß die Menge aller Vorgänge erstellt werden. Dabei ist ein solcher Detaillierungsgrad der Vorgänge zu wählen, der der Problemstellung angepaßt ist. Feinere Aufgliederungen können eine genauere und flexiblere Planung ermöglichen, erfordern aber eine aufwendigere Strukturanalyse.

Für jeden Vorgang sind die folgenden Fragen zu beantworten:

- Welche Vorgänge sind unmittelbar **vor** dem betrachteten Vorgang abzuschließen (Vorgänger)?
- Welche Vorgänge können unmittelbar **nach** dem betrachteten Vorgang begonnen werden (Nachfolger)?
- Welche Vorgänge können **gleichzeitig** zu dem betrachteten Vorgang durchgeführt werden, d.h. sind von dem betrachteten Vorgang unabhängig?
- Welche Vorgänge können sich mit anderen zeitlich überschneiden?

Technologische und ökonomische Abhängigkeiten geben die Antworten zu diesen Fragen. Dies führt zu zeitlichen Abhängigkeiten, die bei Vorgangsknotennetzen als Pfeile zwischen den Vorgängen dargestellt werden.

Die vier wichtigsten Abhängigkeitsbeziehungen, die durch die Kanten des Netzplanes repräsentiert werden, sind:

Start-Start-Beziehung: Der Anfangszeitpunkt des Vorgängers ist mit dem Anfangszeitpunkt der Nachfolgers verbunden.

Start-Ende-Beziehung: Der Anfangszeitpunkt des Vorgängers ist mit dem Endzeitpunkt der Nachfolgers verbunden.

Ende-Start-Beziehung: Der Endzeitpunkt des Vorgängers ist mit dem Anfangszeitpunkt der Nachfolgers verbunden.

Ende-Ende-Beziehung: Der Endzeitpunkt des Vorgängers ist mit dem Endzeitpunkt der Nachfolgers verbunden.

Neben diesen einfachen Beziehungen gibt es bei den einzelnen Verfahren noch zusätzliche Beziehungstypen: Kanten können mit einer Zeitdauer gewichtet werden. Dieser Wert beschreibt die Dauer, die zwischen dem Ursprungs- und dem Zielereignis der Kante liegt. Zusätzlich können Kanten unterschieden werden, bei denen dieser zeitliche Abstand einen Minimal- oder Maximalwert darstellen soll. Außerdem wurden spezielle Kanten eingeführt, die eine direkte Verbindung zum Projektanfang symbolisieren.

Diese Kantentypen erhöhen zwar die Darstellungsmächtigkeit der Netzpläne, erfordern aber auch einen erhöhten Rechenaufwand bei der Zeitanalyse. Gewöhnlich werden von den einzelnen Netzplantechniken die Arten der zulässigen Beziehungen eingeschränkt. Zur Verdeutlichung der Darstellung in dieser Arbeit wird nur die wichtigste Beziehung, die Ende-Start-Beziehung verwandt: Die Kanten führen vom Endzeitpunkt eines Vorgangs (d.h. von dessen rechter Begrenzungslinie) zum Startzeitpunkt eines nachfolgenden Vorgangs (d.h. zu dessen linker Begrenzungslinie).

Anhand der Ergebnisse dieser Strukturanalyse läßt sich dann ein Netzplan in seiner graphischen Darstellung erstellen. Ein einfacher Netzplan ist in Abbildung 2.1 abgebildet (Er enthält bereits die Zeitinformationen aus dem folgenden Abschnitt).

2.2.4 Zeitanalyse

Mit Hilfe der erstellten Netzplanstruktur soll nun die Zeitanalyse durchgeführt werden. Zunächst muß jedem Vorgang eine Vorgangsdauer zugeordnet werden (vergleiche Abschnitt 2.2.1). Auf dieser Basis soll berechnet werden, inwieweit die bei der Strukturanalyse ermittelten Beziehungen Konsequenzen für den zeitlichen Ablauf haben. Die zeitliche Lage der einzelnen Vorgänge und damit die Dauer des gesamten Projektes werden ermittelt.

Zunächst werden die sogenannten **Plantermine** in das Netz eingetragen. Sie sind von außen vorgegebene Termine, insbesondere der Projektstart und eventuell ein Fertigstellungstermin. Auch einzelne Vorgänge können duch äußere Einflüsse terminlich

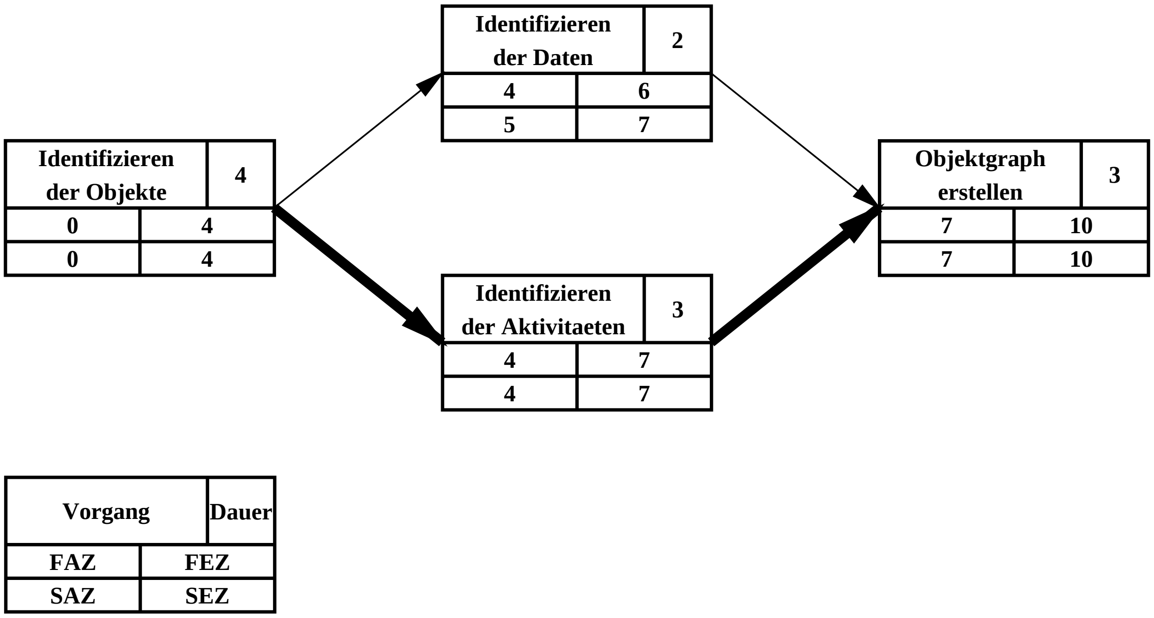

Abbildung 2.1: Netzplan: Objektorientierte Systemarchitektur entwickeln

schon im Voraus festgelegt sein. Diese Plantermine werden in das Netz eingetragen und so bei der Zeitanalyse berücksichtigt. Starttermine werden z.B. in alle Startknoten eingetragen.

Progressive Zeitanalyse

Nachdem alle Plantermine festgelegt sind, wird die progressive Zeitanalyse durchgeführt. Durch diese Vorwärtsrechnung werden die frühesten Start- und Endzeitpunkte der Vorgänge ermittelt. Dazu werden die Termine durch das Netz entlang der Kanten propagiert. Sie werden in den Knoten verrechnet und dann an die nachfolgenden Knoten weitergereicht.

Die Berechnung wird am Beispiel eines Knotens i dargestellt:

FAZ_i: Der früheste Anfangstermin kann durch einen Plantermin festgelegt sein. Besteht noch eine Wahlmöglichkeit, wird er anhand der ankommenden Termine aller Vorgänger von i berechnet: Die ankommenden Kanten von den Vorgängern bedeuten, daß alle diese Vorgänge abgeschlossen sein müssen, bevor der Vorgang i gestartet werden kann. Also ergibt sich FAZ_i als das Maximum der einlaufenden FEZ_j aller Vorgänger j von i.

FEZ_i: Wenn der Vorgang i frühestens zum Zeitpunkt FAZ_i starten kann, so ergibt sich sein frühester Endzeitpunkt FEZ_i als $FAZ_i + d_i$. Dieser Endzeitpunkt wird an alle Nachfolger von Knoten i weiterpropagiert.

Diese Vorgehensweise wiederholt sich für jeden Knoten, dessen gesamte Eingangskanten schon mit Terminen belegt sind. Schließlich sind alle frühesten Termine des Projektes ermittelt.

Retrograde Zeitanalyse

Zur Initialisierung dieser Rückwärtsrechnung muß das Projektende festgelegt werden. Das kann extern geschehen; häufig wird jedoch wie folgt vorgegangen: Alle Vorgänge ohne Nachfolger sind Endknoten für dieses Projekt. Die FEZ_i von allen diesen Endknoten i werden verglichen und das Maximum wird als SEZ_i in alle Endknoten i des Netzes eingetragen. Das Projekt soll spätestens dann beendet sein, wann es auch frühestens beendet werden kann.

Die retrograde Zeitanalyse berechnet die spätesten Start- und Endzeitpunkte und erfolgt analog zur progressiven Zeitanalyse in umgekehrter Richtung: Termine werden von den Nachfolgern des Knotens i empfangen, verrechnet und an alle Vorgänger weiterpropagiert.

Die Berechnung wird am Beispiel eines Knotens i dargestellt:

SEZ_i: Der späteste Endtermin steht entweder schon fest (z.B. ein Plantermin) oder wird anhand der ankommenden Termine aller Nachfolger berechnet: Die ausgehenden Kanten zu Nachfolgern bedeuten, daß Vorgang i spätestens dann beendet sein muß, wenn der erste seiner Nachfolger beginnen muß. Also ergibt sich SEZ_i als das Minimum der einlaufenden SAZ_j aller Nachfolger j von i.

SAZ_i: Wenn der Vorgang i spätestens zum Zeitpunkt SEZ_i beendet sein muß, so ergibt sich sein spätester Startzeitpunkt SAZ_i als $SEZ_i - d_i$. Dieser Startzeitpunkt wird an alle Vorgänger von Knoten i weiterpropagiert.

Diese Vorgehensweise wiederholt sich für jeden Knoten, dessen gesamte Ausgangskanten schon mit Terminen belegt sind. Schließlich sind alle spätesten Termine des Projektes ermittelt.

Die Zeitrechnung läßt sich im folgendem Algorithmus Propagate_Time zusammenfassen, der in einem PASCAL-ähnlichen Pseudocode angegeben ist.

Propagate_Time

Eingabe: Menge von Knoten und Kanten, Termine für den Projektstart und das Projektende

Ausgabe: Belegung der Knoten mit den frühesten und spätesten Anfangs- und Endzeitpunkten

```
Fuer alle Knoten i ohne Vorgaenger tue
   FAZ(i) := Projektstarttermin
Wiederhole
   Waehle einen Knoten i, dessen Eingangskanten alle mit
      einem FEZ-Termin belegt sind.
   FAZ(i) := max (FEZ(j)) fuer alle Vorgaenger j von i
   FEZ(i) := FAZ(i) + d(i)
   Propagiere FEZ(i) an alle Nachfolger von i
bis alle Knoten ausgewaehlt wurden

Projektendtermin := max (FEZ(i)) fuer alle Endknoten i des Netzes

Fuer alle Knoten i ohne Nachfolger tue
   SEZ(i) := Projektendtermin
Wiederhole
   Waehle einen Knoten i, dessen Ausgangskanten alle mit
      einem SAZ-Termin belegt sind
   SEZ(i) := min (SAZ(j)) fuer alle Nachfolger j von i
   SAZ(i) := SEZ(i) - d(i)
   Propagiere SAZ(i) an alle Vorgaenger von i
bis alle Knoten ausgewaehlt wurden
```

Nach dieser Zeitrechnung stehen für jeden Vorgang i die vier Zeitpunkte $FAZ_i, FEZ_i, SAZ_i, SEZ_i$ fest. In Abbildung 2.1 wurde die Zeitanalyse bereits durchgeführt.

Aus dem Netzplan können jedoch noch zusätzliche Informationen herausgelesen werden: Die meisten Vorgänge weisen eine Differenz zwischen den frühesten und spätesten Zeitpunkten auf. Diese Differenz beschreibt eine zeitliche Verschiebungsmöglichkeit des Vorgangs, die den gesamten Projektablauf nicht beeinflußt.

Kritischer Pfad: Als kritischen Pfad bezeichnet man diejenige Vorgangskette, in der die Vorgänge keine zeitliche Verschiebungsmöglichkeit haben. Das bedeutet, daß die maximal für einen Vorgang zur Verfügung stehende Zeit gleich dessen Dauer ist. Bei diesen Vorgängen stimmen also die frühesten und spätesten Termine (d.h. Anfangs- und Endzeitpunkt) überein.

In Abbildung 2.1 ist der kritische Pfad durch dicke Pfeile hervorgehoben.

Pufferzeiten: Die Differenz zwischen dem frühesten und dem spätesten Anfangszeitpunkt eines Vorgangs beschreibt dem Dispositionsspielraum. Innerhalb dieses Zeitraumes kann der Anfang des Vorgangs verschoben werden, ohne die minimal mögliche Projektdauer zu beeinflussen. Um diesen zeitlichen Spielraum differenzierter zu beschreiben, wird er in (sich teilweise überschneidende) kleinere Zeiträume unterteilt, die Pufferzeiten genannt werden:

Pufferzeiten von Vorgängen sind Zeitspannen, um die der Anfang eines Vorgangs gegenüber einem definiertem Zeitpunkt verschoben werden kann bei bestimmter Beeinflussung der zeitlichen Bewegungsmöglichkeiten der umgebenden Vorgänge [1]

Gesamtpuffer: Der Gesamtpuffer beschreibt die Zeitspanne zwischen der frühesten und spätesten Lage eines Vorgangs. Der Gesamtpuffer des Vorgangs 'Identifizieren der Daten' in Abbildung 2.1 beträgt 1 Zeiteinheit.

Freier Puffer: Der freie Puffer beschreibt diejenige Zeitspanne, um die ein Vorgang aus seiner frühesten Lage verschoben werden kann, ohne die früheste Lage der nachfolgenden Knoten zu verändern. Ist der freie Puffer klein, kann ein Vorgang bei frühester Lage seiner umgebenden Vorgänge nur wenig verschoben werden.

Unabhängiger Puffer: Diejenige Zeitspanne, in der ein Vorgang verschoben werden kann, wenn sich alle Vorgänger in der spätesten Lage und alle Nachfolger in der frühesten Lage befinden. Bei Einhaltung des unabhängigen Puffers wird durch eine Verschiebung des Vorgangs der maximale Bewegungsspielraum für keinen der anderen Vorgänge eingeschränkt.

Die folgenden drei Abschnitte enthalten einen Überblick über die traditionellen Verfahren der Netzplantechnik. Zunächst werden die deterministischen Verfahren vorgestellt, die von einer starren Projektstruktur ausgehen. Die anschließend beschriebenen entscheidungsbasierten Netzplantechniken können auch mehrere alternative Durchführungsmöglichkeiten für ein Projekt abbilden. Diesen beiden Arten ist gemeinsam, daß sie eine flache Vorgangsstruktur verwenden. Möchte man bei komplexen Plänen einzelne Teilpläne zusammenfassen, kann man dagegen in einer hierarchischen Struktur mehrere Abstraktionsstufen einführen. Diese hierarchische Netzplanung wird in einem eigenen Abschnitt erläutert.

2.3 Deterministische Verfahren

Die deterministischen Verfahren der Netzplantechnik gehen von einer streng determinierten Struktur des Projektes aus. Das bedeutet, daß die im Verlauf eines Projektes eintretenden Ereignisse und die auszuführenden Tätigkeiten schon im Voraus festgelegt werden können. Dies äußert sich darin, daß **alle** vom Knoten wegführenden Kanten realisiert werden müssen [30] und man keine Wahlmöglichkeit bezüglich des Projektfortgangs hat. Die Ausgangskanten sind durch eine UND-Verknüpfung verbunden.

Um die einzelnen Verfahren voneinander abzugrenzen, werden die jeweils zugelassenen Knotentypen genauer beschrieben. Je nach Kantenverknüpfung bei ankommenden und ausgehenden Kanten wird unterschieden zwischen 'UND-UND-Knoten', 'UND-XODER-Knoten', 'XODER-UND-Knoten' und 'XODER-XODER-Knoten'. Bei den deterministischen Verfahren werden konjunktive Knotentypen (d.h. UND-UND-Knoten und XODER-UND-Knoten) verwandt.

Eine andere Eigenschaft der deterministischen Verfahren ist, daß die Vorgangsdauern deterministisch festliegen.

2.3.1 Methode des Kritischen Pfades (Critical Path Method CPM)

Dies ist (neben den Verfahren der Abschnitte 2.3.2 und 2.3.3) eine der drei Methoden, die in den Jahren 1957/58 den neuen Weg der Planungstechnik beschritten. Sie wurde in den Firmen du Pont und Remington Rand zur Planung der Wartungs- und Umstellungsarbeiten in der chemischen Industrie entwickelt [13].

CPM verwendet Vorgangspfeilnetze und gestattet bei der Netzplanstruktur nur Ende-Start-Beziehungen. Zur Darstellung von Vorgängen werden ausschließlich UND-UND-Knoten verwandt. Als Vorgangsdauer wird die wahrscheinlichste Dauer verwendet.

2.3.2 Program Evaluation and Review Technique (PERT)

Dieses Verfahren wurde 1956 in den USA zur Überwachung militärischer Forschungs- und Entwicklungsprojekte entwickelt [24, 25, 30]. Es verwendet Ereignisknotennetze, bei denen die Pfeile die Reihenfolge der Ereignisse angeben. Auch hier werden UND-UND-Knoten benutzt. Das Besondere an PERT ist die Bestimmung der Vorgangsdauern: Es wird berücksichtigt, daß die geschätzten Zeiten bestenfalls Mittelwerte sind und aufgrund statistischer Untersuchungen gefunden werden.

Nimmt man eine Beta-Verteilung der Zufallsvariablen 'Vorgangsdauer' an, so kann man deren Mittelwert t und Varianz σ^2 aus den 3 Werten optimistische (d_o) / wahrscheinlichste (d_w) / pessimistische (d_p) Vorgangsdauer berechnen [30]:

$$t = \frac{d_o + 6d_w + d_p}{6}$$

$$\sigma^2 = \left(\frac{d_p - d_o}{6}\right)^2$$

Die Zeitanalyse erfolgt wie bei CPM unter Verwendung des Mittelwerts t. Die errechneten Ergebniszeiten sind dann normalverteilte Erwartungswerte mit entsprechender Varianz.

Das PERT-Verfahren ist wegen des höheren Aufwandes nur dann sinnvoll, wenn die Wahrscheinlichkeitsinformationen auch tatsächlich vorliegen und sie als Ausgabe von PERT auch benutzt werden.

2.3.3 Metra-Potential-Methode (MPM)

Dieses Verfahren wurde 1958 von Bernard Roy entwickelt [27, 30]. MPM verwendet Vorgangsknotennetze, wobei die Kanten mit Dauern d gewichtet sind. Vier verschiedene Start-Start-Beziehungen sind zugelassen:

1. Vorgang B kann frühestens d Zeiteinheiten nach dem Beginn von Vorgang A starten.
2. Vorgang B kann frühestens d Zeiteinheiten nach dem Projektstart beginnen.
3. Vorgang B muß spätestens d Zeiteinheiten nach dem Beginn von Vorgang A starten.
4. Vorgang B muß spätestens d Zeiteinheiten nach dem Projektstart beginnen.

Eine Besonderheit ist die sogenannte Bündelbedingung, bei der ein Vorgang bereits beginnen kann, wenn erst einer seiner Vorgänger beendet ist und die anderen noch andauern. Bei MPM können also zusätzlich XODER-UND-Knoten verwandt werden.

Die Zeitrechnung ist bei MPM komplizierter als bei CPM, da zur Berechnung mehrere Iterationen notwendig sind, kommt aber zum gleichen Ergebnis. Die Netzpläne sind wegen der zahlreichen Ausdruckmöglichkeiten nicht sehr anschaulich.

2.3.4 Precedence Diagramming Methode (PDM)

PDM [11] verwendet ebenfalls Vorgangsknotennetzpläne. Diese Methode wurde 1964 entwickelt und ist eine sinnfälligere und verständlichere Version von MPM. Die eingeschränkten Darstellungsmöglichkeiten lassen jedoch nur Beziehungen vom Typ 1 der MPM-Methode zu.

2.4 Entscheidungsbasierte Verfahren

Bei dieser Klasse von Netzplan-Verfahren ist die Struktur des Projektes nicht von vorneherein festgelegt: Welche der Vorgänge auszuführen sind und welche der Ereignisse eintreten, wird erst während des Projektablaufs bestimmt. Die starre Form der konjunktiven Knotentypen der deterministischen Verfahren wird aufgelockert. Es müssen nicht mehr alle ausgehenden Kanten realisiert werden. Es sind die disjunktiven Knotentypen UND-XODER-Knoten und XODER-XODER-Knoten zulässig. [29] beschreibt die Abbildung von solchen Entscheidungsnetzplänen auf UND-ODER-Graphen.

Es wird zusätzlich die Frage gestellt, auf welchem Weg das Projektziel erreichbar ist und wieviel Zeit der jeweilige Projektablauf erfordert.

2.4.1 Decision Boxes

Zusätzlich zu den konjunktiven Knotentypen werden hier Entscheidungsknoten vom Typ UND-XODER in den Netzwerken zugelassen [4]. An einem solchen Entscheidungsknoten muß eine Entscheidung für **eine** der Alternativen getroffen werden. Die übrigen ausgehenden Kanten und deren Teilbäume werden nicht mehr betrachtet.

Benötigte man bisher für jede Alternative ein separates Netzwerk, so läßt sich nun der gesamte Ablauf in einem einzigen Netz abbilden. Es wird allerdings kein Verfahren zur Zeitanalyse angeboten.

2.4.2 Generalized Activity Networks (GAN)

Bei GAN-Netzwerken [6, 5, 30] wird ein zusätzlicher Freiheitgrad miteinbezogen: es werden auch disjunktive Knoten**ein**gänge zugelassen. Insgesamt gibt es sechs Knotentypen [30]:

1. XODER-UND-Knoten
2. ODER-UND-Knoten
3. UND-UND-Knoten
4. XODER-XODER-Knoten
5. ODER-XODER-Knoten
6. UND-XODER-Knoten

Wenn die zu treffenden Entscheidungen keine echten Entscheidungen sind, sondern Versuchs- oder Testergebnisse, können den Kanten auch Wahrscheinlichkeiten für deren Realisierung zugeordnet werden. Außerdem können Rückkopplungen und rekursive Prozesse erfaßt werden.

2.4.3 Graphical Evaluation and Review Technique (GERT)

Dieses Verfahren [21] stellt eine Kombination von GAN und PERT dar: Es wird ein Indeterminismus bezüglich der Struktur erfaßt und außerdem Unsicherheiten in der Zeitschätzung mit Hilfe von Betaverteilungen erfaßt. Die sechs Knotentypen entsprechen den GAN-Knoten, wobei jede Kante durch zwei Parameter charakterisiert wird [30]:

- Der Parameter p gibt die Wahrscheinlichkeit für die Realisierung dieses Weges an.
- Der Parameter t gibt für den Fall der Ausführung die erforderliche Zeit in Form einer Zufallsvariablen an.

Die Zeitanalyse in GERT-Netzwerken wird ausschließlich für XODER-XODER-Knoten durchgeführt.

2.5 Hierachische Netzpläne

Das Charakteristikum der hierarchischen Netzplanverfahren liegt darin, daß die Verknüpfung zweier Knoten nicht mehr die zeitliche Abfolge, sondern die Hierarchie einer verfeinerten Projektzerlegung beschreibt. Um diesen Unterschied zu verdeutlichen, heißt dieser Abschnitt 'Hierarchische Netzpläne'.

Diese Verfahren wurde u.a. von [25, 30] vorgestellt, insbesondere wird hier das Verfahren von Ludwig Hennicke [9] zugrundegelegt, der diese Netzpläne als vertikale Netzpläne bezeichnet.

Das Ziel dieser neuen Darstellungsart ist einerseits eine Abbildung verschiedener Alternativen und andererseits die Beibehaltung deterministischer Netzplanberechnungen. Es wird sich jedoch zeigen, daß eine deterministische Netzplanberechnung erst dann durchgeführt werden kann, wenn man sich schon an jedem Entscheidungsknoten für eine der Alternativen entschieden hat.

2.5.1 Netzplanelemente

Die Knoten des Netzplangraphen beschreiben (komplexe) Vorgänge; die Kanten beschreiben die Hierarchien einer verfeinerten Projektzerlegung. Es werden drei Knotentypen eingeführt:

Subnetzplanknoten: Dieser Knoten repräsentiert einen geschlossenen Subnetzplan mit genau einem Start- und einem Endknoten. Der Subnetzplanknoten ist in folgender Weise mit dem Subnetzplan verbunden: Kann der Subnetzplanknoten ausgeführt werden, so gilt dies auch für den Anfangsknoten des Subnetzplanes. Ist der Endknoten des Subnetzplanes abgeschlossen, so ist auch der Subnetzplanknoten abgeschlossen.

In der graphischen Darstellung werden die Vorgänge des Subnetzplanes durch UND-Kanten als Söhne des Subnetzplanknotens eingetragen. Die eigentliche Struktur des Subnetzplanes wird nicht dargestellt, sondern implizit im Subnetzplanknoten abgebildet.

Subnetzplanknoten werden durch Rechtecke mit doppelten Seitenlinien dargestellt.

Entscheidungsknoten: Der Entscheidungsknoten repräsentiert eine von mehreren verschiedenen Alternativen einer Projektdurchführung [9]. Dazu werden die Knoten, die die einzelnen Alternativen repräsentieren, durch XODER-Kanten mit dem Entscheidungsknoten verbunden. Entscheidungsknoten haben im Gegensatz zu den beiden anderen Knotentypen keinen eigenen Zeitbedarf.

Entscheidungsknoten werden als auf der Spitze stehende Rauten abgebildet.

Vorgangsknoten: Diese Knoten repräsentieren 'atomare' Vorgänge und beinhalten eine Angabe der Vorgangsdauer. Vorgangsknoten werden als Kreise dargestellt.

Abbildung 2.2 stellt einen hierarchischen Netzplan für den Vorgang 'Systemarchitektur entwickeln' dar.

2.5.2 Zeitanalyse

Das Ziel der Zeitanalyse ist die Errechnung der frühesten und spätesten Anfangs- und Endzeitpunkte der Netzplanelemente. Zu Beginn werden die externen Terminvorgaben (Plantermine) in den Wurzelknoten, der das gesamte Projekt repräsentiert,

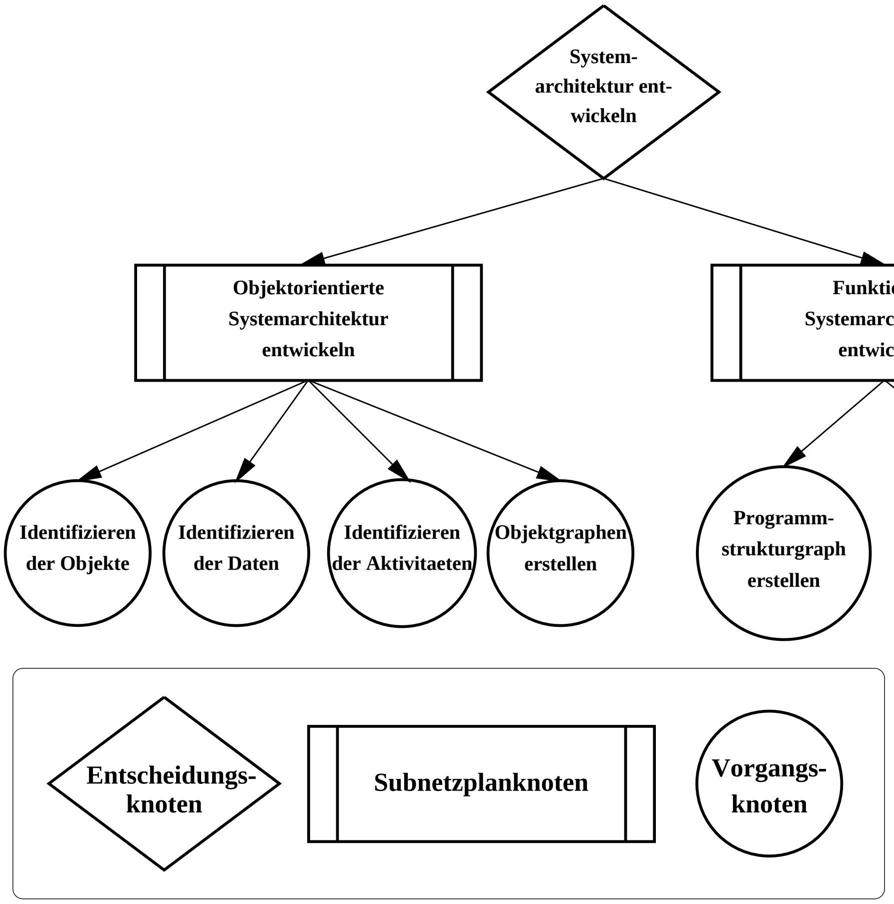

Abbildung 2.2: Hierarchischer Netzplan: Systemarchitektur entwickeln

eingetragen. Anschließend werden diese Termine durch den Baum in Richtung der Blätter propagiert. Dabei unterscheidet sich die Berechnung für die drei Knotentypen:

- **Subnetzplanknoten:**
  Bekommt ein Subnetzplanknoten einen frühesten Anfangstermin zugeteilt, so bedeutet dies, daß der Startknoten des durch ihn repräsentierten Subnetzplanes ebenfalls diesen Termin zugeordnet bekommt. Dieser Termin wird durch die übliche progressive Zeitrechnung durch den Subnetzplan propagiert. Wird dem Endknoten ein FEZ-Termin zugeordnet, gilt dieser ebenfalls für den Subnetzplanknoten. Die retrograde Zeitrechnung verläuft analog in umgekehrter Richtung.

  Abbildung 2.3 stellt die Zeitpropagierung zwischen Subnetzplanknoten und Subnetzplan graphisch dar. Die Struktur des Subnetzplanes ist hier explizit veranschaulicht.

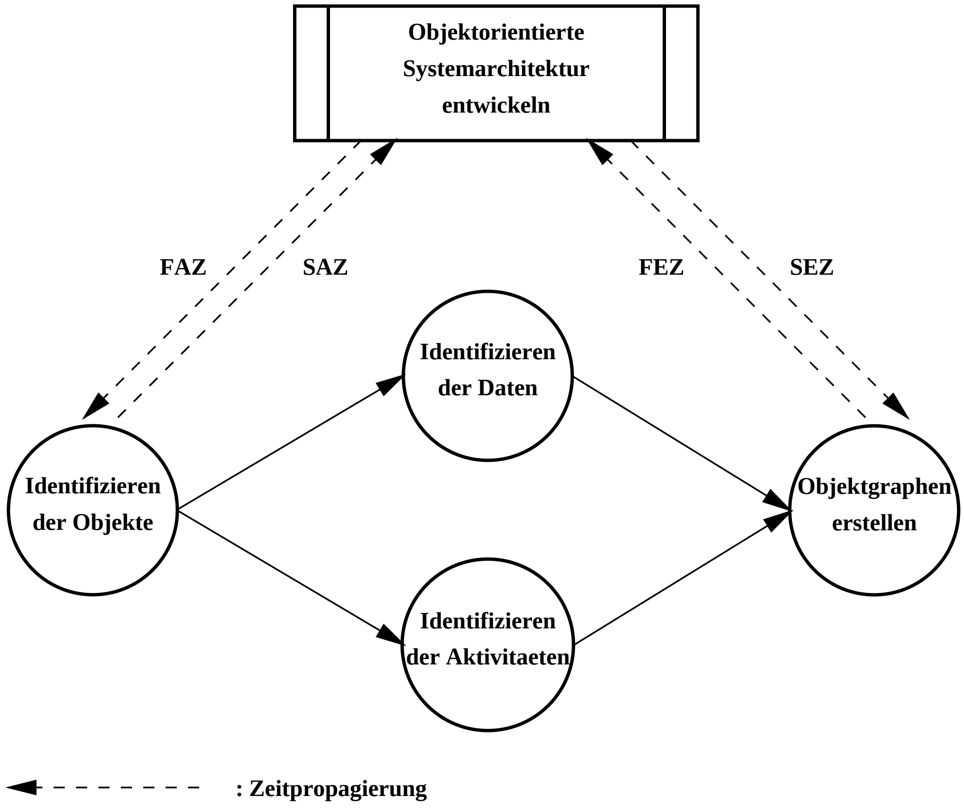

Abbildung 2.3: Zeitpropagierung bei Subnetzplanknoten

- **Entscheidungsknoten:**
  Der FAZ- (aus der Vorwärtsanalyse) und der SEZ-Termin (aus der Rückwärtsanalyse) werden vom Entscheidungsknoten an alle Durchführungsalternativen weitergereicht. Die übliche Berechnung findet in jedem alternativen Netzplan statt. Die Ergebnisse der Zeitpropagierung wird nur von derjenigen Alternative an den Entscheidungsknoten hochgereicht, die die tatsächlich ausgewählte Alternative darstellt. Dieses Verfahren kann also mehrere Alternativen zwar darstellen, nicht jedoch in die Zeitanalyse miteinbeziehen. Es wird sich zeigen, daß das Verfahren der Allgemeine Hierarchischen Netzplantechnik (vgl. Kapitel 5) auch mehrere Alternativen zeitlich abschätzen kann.

  Abbildung 2.4 stellt die Zeitpropagierung im Entscheidungsknoten graphisch dar.

- **Vorgangsknotenknoten:**
  Die Zeitpropagierung durch Vorgangsknoten erfolgt wie bei den übrigen Verfahren dieses Kapitels: Die progressive Zeitanalyse berechnet das Maximum der Werte auf den einlaufenden Kanten, die retrograde Analyse das Minimum der Werte auf den ausgehenden Kanten.

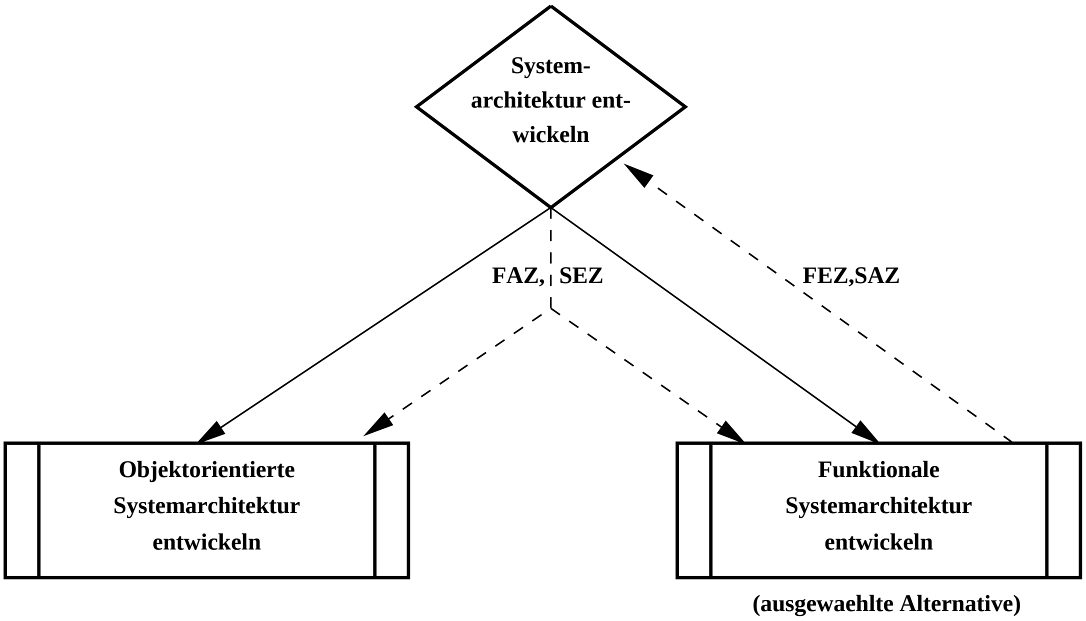

Abbildung 2.4: Zeitpropagierung bei Entscheidungsknoten

Kapitel 3

CoMo-Kit

In diesem Kapitel werden die wichtigsten Elemente des Modellierungswerkzeuges CoMo-Kit (Conceptual Model Construction Kit) kurz vorgestellt, soweit sie für das Verständnis dieser Arbeit benötigt werden. Im ersten Abschnitt werden die Ziele und die wesentlichen Funktionen vorgestellt. Die zentralen Elemente von CoMo-Kit sind in den beiden folgenden Abschnitten erläutert: Die statische Wissensbasis von CoMo-Kit, das sogenannte konzeptuelle Modell und die Ausführungskomponente. Für weitergehende Informationen wird auf [16, 3, 23] verwiesen.

3.1 Aufgaben und Funktionen von CoMo-Kit

Das Werkzeug CoMo-Kit dient zur Modellierung und Abwicklung von kooperativen Designprozessen [16]. Aus elementaren Basisstrukturen lassen sich komplexe Modelle aufbauen, die einen verteilten Entwurfsprozeß beschreiben. Mit Hilfe verschiedener Modellierungswerkzeuge lassen sich diese Modelle erzeugen, ansehen und abändern. Die einzelnen Objekte können mit spezifischen Editoren bearbeitet werden.

CoMo-Kit verfügt über eine Interpreterkomponente [3, 23], die die entwickelten Modelle operationalisieren und damit validieren kann. CoMo-Kit wird als Client/Server-System implementiert: Der Interpreter bildet den Server, der die Funktionen der Clienten, also der einzelnen Agenten, steuert und die dynamischen Informationen verwaltet.

Ziele, die durch die Anwendung von CoMo-Kit erreicht werden können, sind:

- Ermöglichung einer verteilten Problemlösung
- Entwicklung eines besseren Verständnisses der zu bearbeitenden Aufgabe und der benötigten Kooperationsprozesse
- Simulation der Problemlösung und Prototyping

CoMo-Kit wird in zwei Domänen angewandt und weiterentwickelt: In der Bebauungsplanung [17] und im Bereich des Software-Engineering. Im Sonderforschungsbereich 501 'Entwicklung großer Systeme mit generischen Methoden' wird CoMo-Kit zur Planung und Abwicklung von Softwareentwurfsprozessen eingesetzt.

3.2 Das konzeptuelle Modell

Da CoMo-Kit die Entwicklung wissensbasierter Systeme unterstützt, ordnet sich das System in den Rahmen des Knowledge-Engineering ein. Eine Methode der Wissenserhebung wird als modellbasierte Wissensakquisition bezeichnet [16]. Das Ziel ist der sukzessive Aufbau eines Modells der behandelten Domäne. Es wird in dem KADS-Ansatz [28] zur modellbasierten Wissensakquisition als **konzeptuelles Modell** bezeichnet. Es beschreibt die Wissenstypen der Domäne, die Verarbeitungsmethoden und damit die Vorgehensweise bei der Lösung einer Aufgabe.

Ein konzeptuelles Modell besteht aus folgenden Basisobjekten:

Aufgaben: Aufgabe beschreiben Ziele, die durch die Problemlösung erreicht werden sollen. Eine hierarchische Anordnung von Aufgabe und deren Teilaufgaben beschreiben eine verfeinerte Problemzerlegung. Es wird unterschieden zwischen:

- komplexen Aufgaben, die in Teilaufgaben weiter zerlegt werden und
- atomaren Aufgaben, die unteilbare Probleme darstellen. Sie können von einem Bearbeiter gelöst werden, der dabei entsprechende Wertzuweisungen vornimmt.

Methoden: Eine Methode beschreibt einen von mehreren alternativen Lösungsmöglichkeiten für eine komplexe Aufgabe. Alle Methoden für eine Aufgabe bilden die Konfliktmenge.

- Komplexe Methoden beschreiben die Zerlegung einer Aufgabe in Teilaufgaben. Sie definieren die Ein- und Ausgaben der Teilaufgaben und den Informationsfluß zwischen ihnen.
- Atomare Aufgaben beschreiben Wertzuweisungen für atomare Aufgaben

Aufgaben und Methoden zusammen bilden einen UND-ODER-Baum, wie er in Abbildung 7.1 dargestellt ist. Dabei sind Aufgaben als Kreise und Methoden als Rechtecke dargestellt.

Konzepte: Konzepte erfassen die Daten der Domäne, die in Form von Ein- und Ausgaben der Aufgaben vorkommen. Konzeptklassen definieren eine Wissensstruktur der Konzepte mit verschiedenen Attributen. Instanzen dieser Klassen belegen Attribute mit Werten. Das Wissen in den Konzeptinstanzen wird in Problemfall- und Problemlösewissen unterschieden: **Problemfallwissen** ist schon bei der Modellierung bekannt. Es wird durch vollständig instantiierte Konzeptinstanzen abgebildet. **Problemlösewissen** wird erst während der Abwicklung erzeugt. Diese Konzeptinstanzen beschreiben veränderliches Wissen, das als (Teil-) Ergebnisse des Problemlöseprozesses benutzt wird.

formale Parameter: Sie verknüpfen Konzeptklassen mit Aufgaben. Konkrete Ausprägungen der Konzeptklassen, die während der Laufzeit als Ausgabe einer Aufgabe entstehen, werden als formaler Parameter an die folgende Aufgabe weitergereicht.

Agenten: Agenten sind die Bearbeiter der Aufgaben. Es können sowohl menschliche Bearbeiter als auch Computeragenten vorkommen, die automatisch bestimmte Teilaufgabe lösen. Bei komplexen Aufgaben verteilen Agenten die Teilaufgaben und überwachen deren Ausführung. Atomare Aufgaben werden abgearbeitet und als Ergebnis Wertzuweisungen vorgenommen.

Diese Basisobjekte sind durch gerichtete Kanten verbunden, die die Zerlegungsstruktur (bei der Aufgaben-Methoden-Hierarchie) oder den Informationsfluß (zwischen Konzeptinstanzen, formalen Parametern und Aufgaben beschreiben.

3.3 Die Ausführungskomponente

Der Interpreter (auch 'Scheduler') dient dazu, den im konzeptuellen Modell abgebildeten Prozeß auszuführen. Das Ziel ist die Operationalisierung und damit eine Validierung des Modells. Es kann eine verteilte Problemlösung realisiert werden, bei der mehrere Agenten parallel arbeiten und durch den Interpreter koordiniert werden. Dabei hat der Interpreter folgende Aufgaben:

- Die **Aufgabenzerlegung** wird verwaltet. Bearbeitet ein Agent eine komplexe Aufgabe, so muß er sich für eine der alternativen Lösungsmethoden entscheiden. Die ausgewählte Methode legt die Zerlegung in Teilaufgaben fest. Der Agent delegiert jede dieser Teilaufgaben und hat damit die Aufgabe bearbeitet.

  Ist die Aufgabe atomar, wird durch die Bearbeitung eine Belegung der Ausgabeparameter der Aufgabe erreicht.

- Der Interpreter berücksichtigt bei der Abwicklung die **Informationsflußabhängigkeiten**, die durch die Informationsflußdiagramme gegeben sind. Formale Parameter treten als Ausgabe einer Aufgabe und als Eingabe der folgenden Aufgabe auf. Es werden deshalb nur diejenigen Aufgaben zur Bearbeitung freigegeben, deren Eingaben schon vorhanden sind.

- Die Ausführungskomponente verwaltet den **Zustand des Lösungsprozesses**. Jede Aufgabe hat einen Bearbeitungszustand, von dem der weitere Ablauf abhängt. Außerdem werden Entscheidungen protokolliert und Rückzüge von

Entscheidungen berücksichtigt. Dabei achtet der Interpreter auf die Konsistenz der gesamten Lösung.

Abbildung 3.1 zeigt das Zusammwirken der statischen Wissensbasis und der dynamischen Ausführungskomponente.

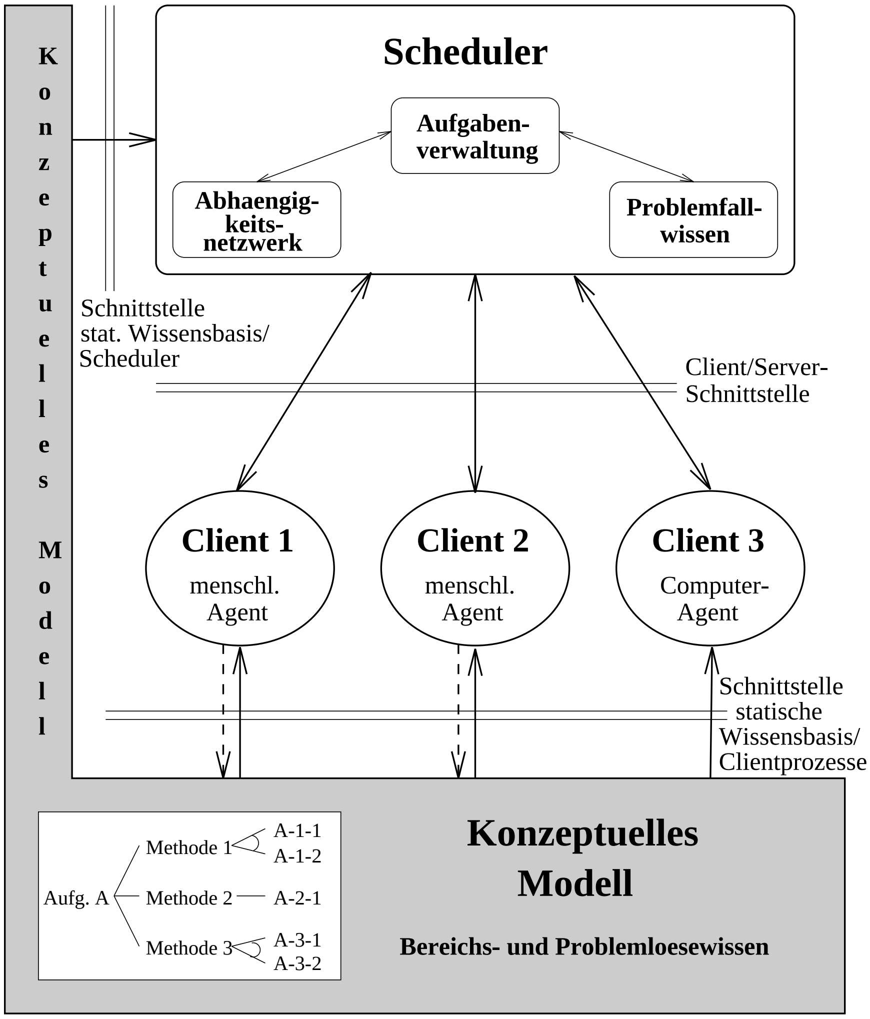

Abbildung 3.1: Konzeptuelles Modell und Scheduler bei CoMo-Kit

Kapitel 4

Mängel traditioneller Verfahren in Hinblick auf CoMo-Kit

In diesem Kapitel sind die Gründe zusammengestellt, wegen derer die traditionellen Netzplan-Verfahren für die Zeitplanung in CoMo-Kit nicht genutzt werden können.

4.1 Geschlossene Subnetzpläne

Die traditionellen Verfahren verwenden ausschließlich geschlossene Netzpläne. In Abbildung 4.1 ist ein konzeptuelles Modell von CoMo-Kit abgebildet, das einen offenen Subnetzplan enthält. Die zugehörigen Informationsflüsse sind in Abbildung 4.2 angegeben.

Das Verfahren der hierarchischen Netzpläne verlangt, daß die Subnetzpläne **geschlossene** Netzpläne sind, also genau ein Anfangs- und ein Endereignis haben. In diesem Fall jedoch hat der Knoten A2 einen **offenen** Subnetzplan. Die beiden Teilaufgaben A21 und A22 können zu unterschiedlichen Zeitpunkten starten und enden; sie sind voneinander unabhängig. Ein Algorithmus zur Zeitanalyse sollte in diesem Fall als Dauer des Vorgangs A den Wert 10 ausgeben. Für traditionelle Verfahren müßten die Enden des Subnetzplanes künstlich zusammengeführt werden. Die Dauer des Vorgangs A wäre 12.

Bei CoMo-Kit wird durch eine Aufgabe ein Ziel repräsentiert, das mit Hilfe einer Methode erreicht werden kann. Die Methode beinhaltet einen Informationsflußgraphen, der die Abarbeitung der einzelnen Teilaufgaben beschreibt. Dieser Informationsflußgraph kann beliebig viele Start- und Endknoten haben. In der Netzplantheorie herrschen verschiedene Meinungen darüber, ob man offene Subnetzpläne überhaupt zulassen sollte. Kann also ein Netzplan mit mehreren Anfangs- und Endereignissen überhaupt zu einem Vorgang zusammengefaßt werden?

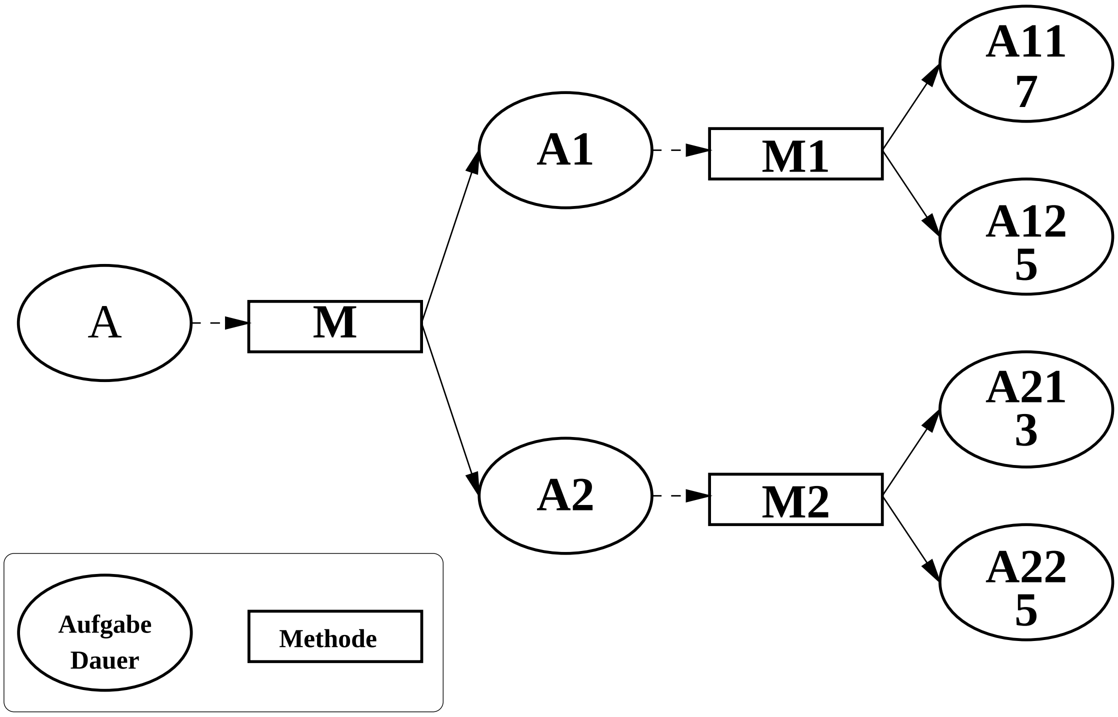

Abbildung 4.1: Konzeptuelles Modell

[19] argumentiert, daß der Netzwerkentwerfer durch den Zwang zu geschlossenen Netzplänen zu zusätzlicher, sinnvoller Disziplin gezwungen wird. Es kann unter Umständen logisch erscheinen, daß jedem Projekt und auch jedem Vorgang ein eindeutiger Start- und Endzeitpunkt zugeordnet werden kann.

Auf der anderen Seite kann die Berechnung eines möglichst realitätsnahen Netzplanes eine Flexibilität bezüglich der Anfangs- und Endereignisse erfordern. Oft wird vorgeschlagen, die offenen Enden künstlich zusammenzuführen [27, 30]. Dadurch ergibt sich eine zu hohe minimale Projektdauer.

Auch Bemühungen, das Netzwerk umzuformen und zu vereinfachen, scheitern in diesem Fall. Es gibt in der Literatur [14, 15, 22, 18, 27, 2] zahlreiche Verfahren, um aus einem komplexen Netz ein einfaches reduziertes Netz zu erzeugen. Dabei werden ganze Teilnetzpläne durch einen einzigen Vorgang ersetzt. Diese reduzierten Netze müssen aber bezüglich der Zeitrechnung die gleiche Wirkung haben wie die ursprüglichen Netze. Daher ist allen Verfahren folgendes gemein:

Ein Ersatzvorgang erhält als Dauer die zeitliche Länge des längsten Verbindungsweges durch das originale Netz von einem bestimmten Eingangsknoten zu einem bestimmten Ausgangsknoten. Dabei müssen alle möglichen Verbindungswege berücksichtigt werden, sodaß ein solcher Ersatzvorgang für jede Kombinationen von Ein- und Ausgangsknoten erstellt werden müßte. Da dies im obigen Beispiel nicht durchgeführt werden kann, führen die Verfahren der Netzwerkreduktion hier nicht zum Erfolg.

4.2 Anordnungsbeziehungen

Bei hierarchischen Netzplänen sind Anordnungsbeziehungen zwischen Knoten aus verschiedenen Subnetzplänen verboten [9]. Die Informationen, die sich aus diesen Abhängigkeiten ergeben, können durch den Propagierungalgorithmus nicht auf die abstrakteren Ebenen hochgereicht werden. Es ist wünschenswert, daß beliebige Abhängigkeitsbeziehungen zwischen allen Teilen eines Netzplanes zugelassen sind. Dieses Abbildungsproblem läßt sich im Rahmen der traditionellen Verfahren nicht befriedigend lösen.

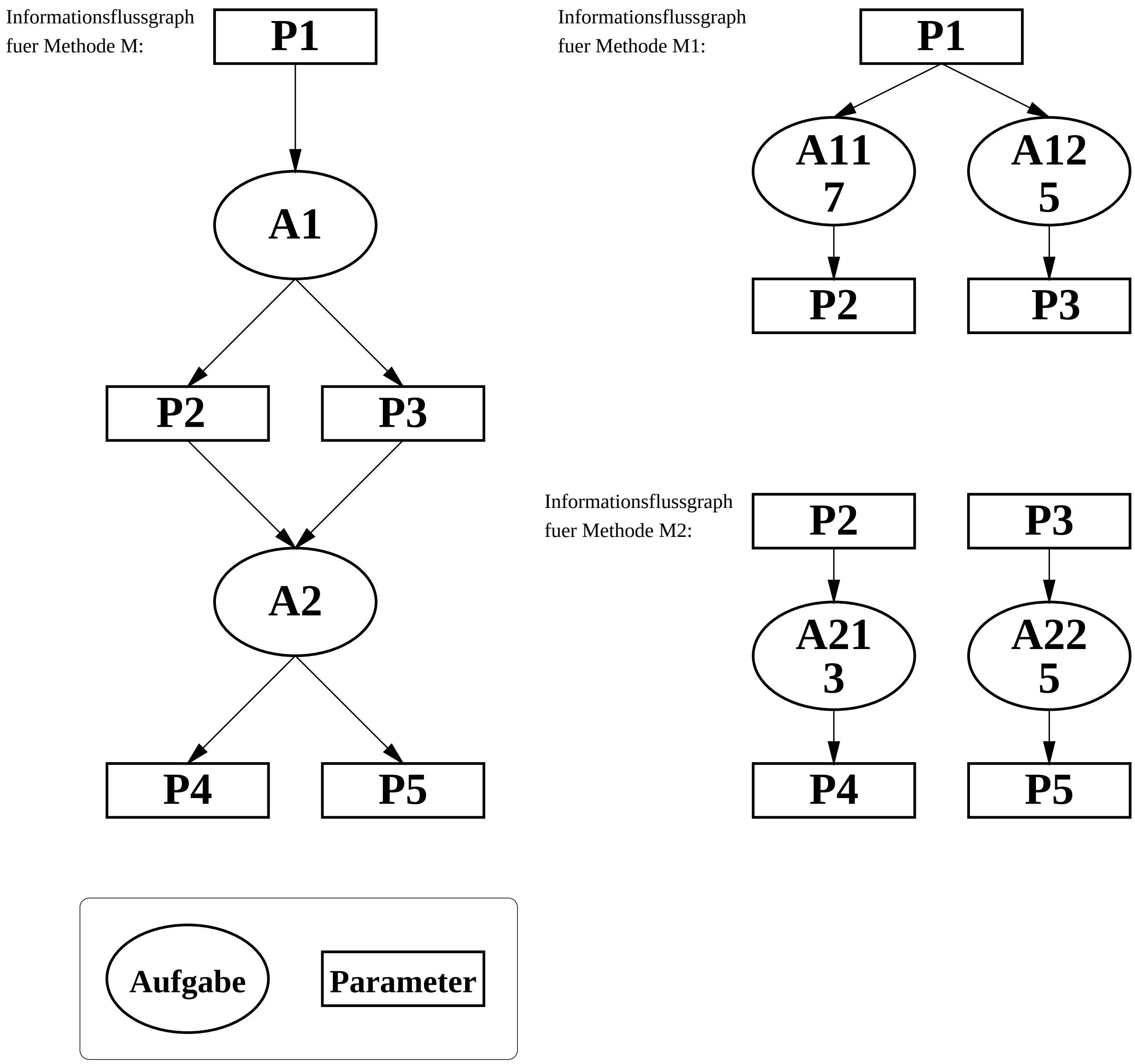

Abbildung 4.2: Informationsflußgraphen

Kapitel 5

Allgemeine Hierarchische Netzplantechnik in CoMo-Kit

Das Werkzeug CoMo-Kit erlaubt es, alternative Vorgehensweisen abzubilden. Im Abschnitt 2.4 wurde beschrieben, daß die konventionelle Netzplantechnik die Repräsentation von Entscheidungsalternativen zuläßt. Die Verfahren konnten sich in der praktischen Anwendung nicht durchsetzen, weil Algorithmen zur Auswertung solcher Entscheidungsnetzpläne strukturell zu aufwendig sind [29].

Durch die Einbeziehung von offenen Subnetzplänen und beliebigen Anordnungsbeziehungen ergeben sich zusätzliche Schwierigkeiten. Gesucht ist hier ein **allgemeines** Verfahren, das diese Netzplanelemente verarbeiten kann.

Für CoMo-Kit ist außerdem charakteristisch, daß die Grundlage des konzeptuellen Modells eine hierarchisch aufgebaute Aufgaben-Methoden-Struktur ist. Die Hierarchieebenen beschreiben dabei die Stufen einer verfeinerten Projektzerlegung. Diese Darstellungsform erfordert eine **hierarchische** Vorgehensweise bei der Netzplanung.

Das in dieser Arbeit entwickelte Verfahren der Allgemeinen Hierarchischen Netzplantechnik löst diese Probleme pragmatisch.

5.1 Zusammenhang zwischen konzeptuellem Modell und Netzplan

In Abschnitt 3.2 wurde das konzeptuelle Modell beschrieben, das Grundlage für die Zeitanalyse ist. Es beschreibt eine Menge von möglichen Durchführungsalternativen. In den Abschnitten 2.4 und 2.5 wurde aufgeführt, daß die Zeitanalyseverfahren der traditionellen Netzplantechniken nicht mehrere Durchführungsalternativen für ein Projekt miteinbeziehen können. Ziel ist es aber, allein anhand des konzeptuellen Modells, also schon vor dessen Operationalisierung und damit etwaigen Durchführungsentscheidungen, eine aussagekräftige Zeitabschätzung vornehmen zu können. Um die dazu benötigten Informationen zu ermitteln, müssen alle Möglichkeiten, das Projekt durchzuführen, in Betracht gezogen werden. Diese Möglichkeiten werden im folgenden als **Lösungen** bezeichnet.

Ziel der Allgemeine Hierarchischen Netzplantechnik ist es, für jede Lösung eine getrennte Zeitanalyse durchzuführen und anschließend die Ergebnisse zusammenzufassen. Dazu muß derjenige Teil des konzeptuellen Modells, der eine Lösung beschreibt, auf einen Netzplan abgebildet werden, um die übliche Netzplantechnik anwenden zu können:

- Die **Aufgaben** entsprechen den Vorgängen in einem Netzplan. Sie verbrauchen Zeit und sollen Anfangs- und Endzeitpunkte erhalten.

- **Methoden** beschreiben in der Aufgaben-Methoden-Hierarchie die Verzweigungspunkte für alternative Durchführungsmöglichkeiten. Sie werden nicht auf den Netzplan abgebildet, legen jedoch die einzelnen Lösungen fest und beschreiben, welche Teilaufgaben zu der jeweiligen Lösung gehören.

  Wichtig ist der Informationfluß zwischen den Teilaufgaben, der in jeder Methode angegeben ist. Pfade, auf denen Informationen fließen, führen von einer Aufgabe über einen oder mehrere formale Parameter zu der nächsten Aufgabe. Diese Pfade werden als Pfeile in den Netzplan eingetragen.

- Die **Konzeptinstanzen** und **formalen Parameter** stellen Produkte der Projektabwicklung dar. Sie haben keinen Einfluß auf die Zeitanalyse.

5.2 Lösungsmöglichkeiten

Um für jede Lösung den zugehörigen Netzplan bearbeiten zu können, müssen zunächst alle möglichen Lösungen generiert werden. Jeder Hyperpfad im Aufgaben-Methoden-Baum repräsentiert eine dieser Lösungen. Ein Hyperpfad eines UND-ODER-Baumes ist ein Teilgraph mit folgenden Eigenschaften:

- Für jeden UND-Knoten (hier: Methode) enthält er alle Nachfolger (hier alle Aufgaben, die zu dieser Methode gehören).

- Für jeden ODER-Knoten (hier: Aufgabe) enthält er genau einen Nachfolger (hier: diejenige Methode, die für diese Lösung ausgewählt wird).

Jede einzelne Lösung wird durch folgende Daten beschrieben:

- Die Menge aller Methoden, die bei der Abarbeitung des Projektes ausgewählt werden müssen, um zu diese Lösung zu gelangen. In diesem Fall ist das Tupel (methode, true) in der Lösungsbeschreibung enthalten.

- Die Menge aller Methoden, die bei der Abarbeitung des Projektes **nicht** ausgewählt werden dürfen, um zu diese Lösung zu gelangen. In diesem Fall ist das Tupel (methode, false) enthalten.

  Existiert für eine Methode methode_X weder ein Tupel (methode_X, true), noch (methode_X, false), so ist es für diese Lösung unerheblich, ob die Methode ausgewählt wurde oder nicht.

- Jede Lösung enthält außerdem den maximalen frühesten Endzeitpunkt aller Endknoten. Diese Zeitangabe stellt den frühesten Termin dar, zu dem das Projekt bei der Auswahl dieser bestimmten Methoden beendet werden kann. Dieser Termin wird als Ausgangspunkt für die retrograde Zeitrechnung benötigt.

- Jede Lösung ist mit IN oder OUT markiert. Diese Markierung zeigt an, ob die Lösung beim momentanen Zustand des Schedulers noch zur Verfügung steht. Denn nur, wenn der Agent noch zu dieser Lösung gelangen kann, muß sie bei der Zeitanalyse berücksichtigt werden.

Als Beispiel dient das konzeptuelle Modell aus den Abbildungen 4.1 und 4.2. Es hat nur eine Lösung, die beschrieben wird durch: {(M,true), (M1,true), (M2,true), (endTime,10), (status,IN)}.

5.3 Zeitanalyse

Dieser Abschnitt beschreibt die Zeitanalyse bei CoMo-Kit. Die Aufgaben erhalten eine zusätzliche Datenstruktur, um die Zeitinformationen abzubilden. Durch eine progressive Zeitanalyse werden die frühesten Zeitpunkte belegt; durch die retrograde Zeitanalyse die spätesten Zeitpunkte.

5.3.1 Zeitinformationen in Aufgaben

Im vorigen Abschnitt wurde eine mögliche Lösung als Hyperpfad im Aufgaben-Methoden-Baum repräsentiert. Der Hyperpfad beschreibt eine hierarchische Dekomposition der Wurzelaufgabe, bei der die atomaren Aufgaben die Blattknoten sind. Die atomaren Aufgaben sind diejenigen, die tatsächlich ausgeführt werden. Sie entsprechen den Vorgängen im Netzplan und ihnen muß eine Vorgangsdauer zugeordnet werden.

Um die Informationen der Zeitanalyse zu repräsentieren, erhält jede dieser Aufgaben eine zusätzliche Datenstruktur. Wie bei den konventionellen Netzplantechniken werden sowohl die frühesten als auch die spätesten Zeitpunkte betrachtet. Ziel dabei ist es, die Pufferzeit eines Vorgangs zu beurteilen und die Grenzen für eine Terminierung zu ermitteln, innerhalb derer die Gesamtprojektdauer nicht beeinflußt wird. Außerdem werden die Anfangs- und Endzeitpunkte betrachtet, da diese Ereignisse von einem Projektmanager eindeutig festzustellen sind.

Die folgenden vier Zeitpunkte werden in atomaren Aufgaben gespeichert:

FAZ: Frühester Anfangszeitpunkt

FEZ: Frühester Endzeitpunkt

SAZ: Spätester Anfangszeitpunkt

SEZ: Spätester Endzeitpunkt

Die Zeitpropagierung wird getrennt für die Blattknoten jedes Hyperpfades, der eine mögliche Lösung darstellt, durchgeführt. Es ist dabei ausreichend, nur die im momentanen Zustand des Schedulers noch möglichen Lösungen zu betrachten. Es werden in der Menge der Lösungen nur diejenigen betrachtet, die mit IN markiert sind. Die Vorgehensweise bei der Vorwärts- und Rückwärtsrechnung gleicht der bei den traditionellen Netzplanverfahren.

5.3.2 Progressive Zeitanalyse

Abschnitt 5.1 beschreibt die Abbildung des konzeptuellen Modells auf einen Netzplan. Alleine die Aufgaben verbrauchen Zeit und entsprechen deshalb den Vorgängen in der Netzplantechnik. Die Pfade zwischen den Aufgaben, auf denen Konzeptinstanzen und formale Parameter liegen, zeigen die zeitlichen Abhängigkeiten auf. Termine werden also durch die Konzeptinstanzen und formalen Parameter nur hindurchpropagiert, und dort nicht gespeichert oder verändert.

Um die Zeitpropagierung zu starten werden alle Anfangsknoten mit dem Termin des Projektstarts initialisiert. Knoten ohne Vorgänger sollen frühestens beim Projektstart beginnen können, also werden die FAZ-Termine aller Anfangsknoten des Netzes mit diesem Startdatum belegt. Durch das Eintragen dieser Plantermine wird die Vorwärtspropagierung gestartet.

Die Anfangsknoten, wie auch alle übrigen Knoten sind entweder Konzeptinstanzen, formale Parameter oder Aufgaben. Die Verarbeitung der Termine innerhalb dieser Knoten geschieht wie folgt:

Konzeptinstanzen: Konzeptinstanzen dienen zur Darstellung von unveränderlichem oder instantiiertem Wissen für den Problemlöseprozeß [23]. Der FAZ-Termin, der einer Konzeptinstanz zugeordnet wird, muß an diejenigen Aufgaben weiterpropagiert werden, die diese Konzeptinstanz als Eingabe verwenden. Dabei dürfen jedoch nur solche Aufgaben berücksichtigt werden, die Bestandteil der aktuellen Lösungsmenge sind. Weiterhin wird die Menge der betroffenen Aufgaben auf diejenigen eingeschränkt, die eine Angabe der Dauer gespeichert haben: Nur zu den Blättern des Aufgaben-Methoden-Baumes soll propagiert werden; die komplexen Aufgaben haben selbst keine eigene Vorgangsdauer. An die betroffenen Aufgaben wird der früheste Anfangstermin weiterpropagiert.

formale Parameter: Wie bei den Konzeptinstanzen muß hier der FAZ-Termin an die Aufgaben weiterpropagiert werden, die diesen formalen Parameter als Eingabe verwenden. Dabei dürfen wieder nur solche Aufgaben berücksichtigt werden, die Bestandteil der aktuellen Lösungsmenge sind und eine Angabe der Dauer haben, also Blattknoten sind.

Aufgaben: Liegen an allen Eingabekanten der Aufgabe Termine an, so kann auch hier die Zeitrechnung beginnen. Das Maximum der einlaufenden FAZ-Termine wird als früheste Anfangszeit in der Aufgabe gespeichert. Anschließend wird die Vorgangsdauer addiert und als frühester Endzeitpunkt gespeichert. Dieser Termin wird dann an alle Ausgabeknoten, d.h. formale Parameter, der Aufgabe weitergeschickt.

5.3.3 Retrograde Zeitanalyse

Die Initialisierung der Rückwärtsrechnung, d.h. die Belegung der SEZ-Termine aller Endknoten, geschieht durch die maximale FEZ-Zeit aller Endknoten der jeweiligen Lösung. Dieser Termin wurde bei der Vorwärtsrechnung ermittelt und wird als Endzeitpunkt für das gesamte Projekt angestrebt.

Die Propagierung erfolgt analog zur Vorwärtsrechnung mit den beiden Unterschieden:

- Konzeptinstanzen werden nicht betrachtet, weil sie nicht als Ausgabe einer Aufgabe vorkommen können.

- Anstatt die Vorgangsdauer zu dem FAZ-Termin zu addieren, um den FEZ-Termin zu erhalten, wird sie von dem SEZ-Termin subtrahiert, um den SAZ-Termin zu erhalten. Dieser Zeitpunkt wird dann an alle Eingabeparameter weiterpropagiert.

5.4 Verallgemeinerung der Zeitinformationen

Nachdem die vier Zeitpunkte FAZ, FEZ, SAZ, SEZ für die atomaren Aufgaben berechnet worden sind, sollen sie abstrahiert und in der Aufgaben-Methoden-Hierarchie in Richtung Wurzel hin verallgemeinert werden. Den komplexen Aufgabe sollen ähnliche Zeitwerte zugeordnet werden können wie den atomaren Aufgaben.

Die komplexen Aufgabe bekommen eine ähnliche Datenstruktur wie die atomaren. Eine Besonderheit dieses Verfahrens liegt in der Betrachtung der offenen Subnetzpläne (siehe Abschnitt 4.1). Offene Subnetzpläne können beliebig viele

Anfangs- und Endknoten haben, um eine möglichst natürliche Modellierung zuzulassen. Es wird damit der Tatsache Rechnung getragen, daß mit CoMo-Kit beliebige Informationsflußgraphen unter einer Methode zusammengefaßt werden können. Subnetzpläne haben keine eindeutigen Anfangs- und Endereignisse mehr, sondern jeweils mehrere Anfangs- und Endpunkte. In Abbildung 5.1 ist ein Teilnetzplan abgebildet mit je zwei Anfangs- und Endknoten.

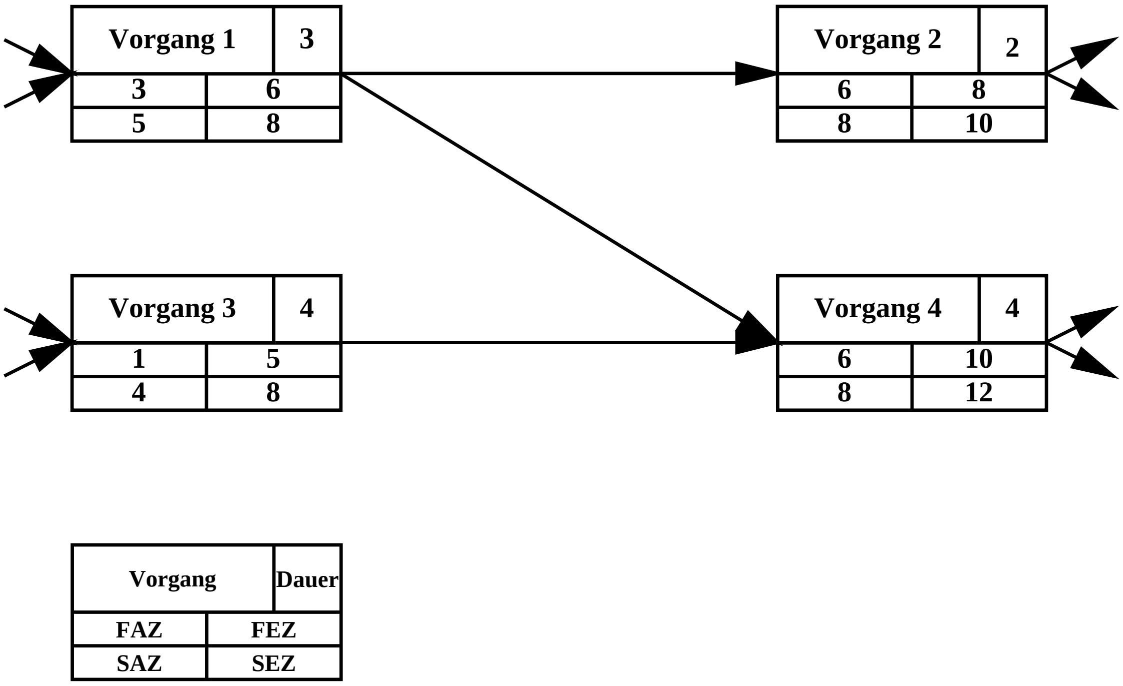

Abbildung 5.1: Offener Teilnetzplan

Für den Projektmanager besonders interessant ist dabei der Vorgang, der als erstes beginnt ('die erste Teilaufgabe kann beginnen') und der Vorgang, der als letztes beendet ist ('die letzte Teilaufgabe ist beendet').

Insgesamt ergeben sich also drei verschiedene Qualitäten der Zeitangaben:

1. frühester / spätester Zeitpunkt
2. Anfangs- / Endzeitpunkt
3. erste / letzte Teilaufgabe

Die Zeitpunkte werden mit Abkürzungen bezeichnet, die diese drei Qualitäten unterscheiden:

1. **F:** Wann kann ich frühestens (d.h. wenn die Methoden ausgewählt werden, die am ehesten fertig sind)
   S: Wann muß ich spätestens (d.h. wenn die Methoden ausgewählt werden, die als letztes fertig werden)
2. **A:** beginnen
   E: enden
3. **E:** mit der ersten Teilaufgabe des betrachteten Knotens?
   L: mit der letzten Teilaufgabe des betrachteten Knotens?

Von den acht Möglichkeiten sind insbesondere die folgenden fünf Zeitpunkte interessant und werden deshalb in der Datenstruktur der Aufgaben und Methoden abgebildet:

FAET: Frühester Anfangszeitpunkt der ersten Teilaufgabe

FEET: Frühester Endzeitpunkt der ersten Teilaufgabe

FELT: Frühester Endzeitpunkt der letzten Teilaufgabe

SALT: Spätester Anfangszeitpunkt der letzten Teilaufgabe

SELT: Spätester Endzeitpunkt der letzten Teilaufgabe

Betrachtet man einen Vorgang, der den in Abbildung 5.1 dargestellten Subnetzplan repräsentiert, so bekommt er die folgenden Zeitpunkte zugeordnet:

FAET: 1

FEET: 8

FELT: 10

SALT: 5

SELT: 12

Das Verfahren läßt sich leicht erweitern, um auch die übrigen Zeitpunkte zu erfassen.

Die Termine der atomaren Aufgaben sollen nun zu den abstrakteren Aufgaben verallgemeinert werden, damit auch auf abstrakter Ebene eine Zeitschätzung möglich ist. Dadurch werden auch Aussagen über das Projekt als Ganzes gemacht. Außerdem werden die Zeiten in den Methoden gespeichert, um dem Benutzer zusätzliche Informationen zur Verfügung zu stellen.

Bei diesem Verfahren sind zwei Arten von Abschätzungen zu berücksichtigen:

1. Damit bei dieser Technik (im Gegensatz zu Abschnitt 2.5) den verschiedenen Durchführungsalternativen Rechnung getragen werden kann, muß eine Abschätzung der Alternativen vorgenommen werden. Besonders interessant für den Projektmanager sind die kürzeste Projektdauer, die benötigt wird, wenn die jeweils 'schnellsten' Methoden ausgewählt werden und eine worst-case Abschätzung, die unter den ungünstigsten Bedingungen eintreffen kann. Bessere Abschätzungen (z.B. Erwartungswerte) lassen sich nicht vornehmen, da keine Informationen über die Auswahlwahrscheinlichkeit einer Methode verfügbar sind.

2. Eine andere Notwendigkeit für Abstraktionen ergibt sich aus der Verwendung von offenen Subnetzplänen: Es kann nicht mehr eindeutig von Start und Ende eines solchen Subnetzplanes gesprochen werden. Stattdessen muß man sich auf eine Abschätzung für dessen erste und letzte Teilaufgabe beschränken.

Diese Abstraktionsschritte werden bei der Verallgemeinerung der Zeitinformationen berücksichtigt. Dazu werden drei Fälle unterschieden: Die Verallgemeinerung der Termine

1. von atomaren Aufgaben mit Vorgangsdauer zu den übergeordneten Methoden,

2. von Methoden zu komplexen Aufgaben und

3. von komplexen Aufgaben zu übergeordneten Methoden.

Diese Schritte werden im Folgenden an Formeln verdeutlicht, wobei gilt:

$FAZ_i^s = Früheste\ Anfangszeit\ der\ atomaren\ Aufgabe\ i\ als\ Bestandteil\ der\ Lösung\ s$
$FAET_i = Früheste\ Anfangszeit\ der\ ersten\ Teilaufgabe\ des\ Knotens\ i\ (für\ Methoden\ und\ komplexe\ Aufgaben)$
$A_i = Menge\ aller\ Anfangsknoten\ des\ Subnetzes,\ das\ der\ Methode\ i\ direkt\ untergeordnet\ ist$
$E_i = Menge\ aller\ Endknoten\ des\ Subnetzes,\ das\ der\ Methode\ i\ direkt\ untergeordnet\ ist$
$S = Menge\ aller\ noch\ gültigen\ Lösungen$
$M_i = Menge\ aller\ alternativen\ Methoden\ zu\ Aufgabe\ \ i$

1. **Atomare Aufgabe $\longrightarrow$ Methode i**

   Durch diesen Schritt wird berücksichtigt, daß den atomaren Aufgaben mit Vorgangsdauer für jede Lösung neue Termine zugeordnet werden. Diese verschiedenen Termine sind bei diesem ersten Abstraktionsschritt durch das Minimum und das Maximum abzuschätzen.

$$\begin{aligned} FAET_i &= \min_{s\in S, j\in A_i}(FAZ_j^s) \\ FEET_i &= \min_{s\in S, j\in E_i}(FEZ_j^s) \\ FELT_i &= \min_{s\in S}(\max_{j\in E_i}(FEZ_j^s)) \\ SALT_i &= \max_{s\in S, j\in A_i}(SAZ_j^s) \\ SELT_i &= \max_{s\in S, j\in E_i}(SEZ_j^s) \end{aligned}$$

2. **Methode $\longrightarrow$ komplexe Aufgabe i**

   Hier wird die Abschätzung der 'schnellsten' und 'langsamsten' Methoden vorgenommen. Zur Berechnung der frühesten Zeitpunkte wird die Selektion der schnellsten Methode angenommen, für die spätesten Zeitpunkte die langsamste Methode.

$$\begin{aligned} FAET_i &= \min_{j\in M_i}(FAET_j) \\ FEET_i &= \min_{j\in M_i}(FEET_j) \\ FELT_i &= \min_{j\in M_i}(FELT_j) \\ SALT_i &= \max_{j\in M_i}(SALT_j) \\ SELT_i &= \max_{j\in M_i}(SELT_j) \end{aligned}$$

3. **Komplexe Aufgabe $\longrightarrow$ Methode i**

   Hier wird die Abschätzung bezüglich der ersten und letzten Teilaufgabe vorgenommen. Für die ersten Teilaufgaben wird das Minimum der betroffenen Aufgaben ermittelt, für die letzten Teilaufgaben das Maximum.

$$\begin{aligned} FAET_i &= \min_{j\in A_i}(FAET_j) \\ FEET_i &= \min_{j\in E_i}(FEET_j) \\ FELT_i &= \max_{j\in E_i}(FELT_j) \\ SALT_i &= \max_{j\in A_i}(SALT_j) \\ SELT_i &= \max_{j\in E_i}(SELT_j) \end{aligned}$$

Diese Schritte werden wiederholt ausgeführt, bis man bei der Wurzel des Aufgaben-Methoden-Baumes angelangt ist und somit dem gesamten Projekt die benötigten Zeitinformationen zugeordnet hat.

Kapitel 6

Erweiterungen der Allgemeinen Hierarchischen Netzplanung

6.1 Einfluß des Interpreters

Die Zeitanalyse wurde im ersten Abschnitt dieses Kapitels statisch betrachtet. Anhand des konzeptionellen Modells wird **vor** der Ausführung des Projektes die benötigte Zeit für seine Vollendung abgeschätzt. Dabei werden alle Hyperpfade berücksichtigt.

Wird ein Scheduler gestartet, beginnt die Projektabwicklung, während der Entscheidungen über den Lösungsweg getroffen werden. Die Agenten bekommen Aufgaben zugeteilt und entscheiden sich für Methoden, um diese Aufgaben zu lösen. Führen solche Entscheidungen nicht zum Ziel, können sie wieder zurückgenommen werden. Solche Informationen aus dem Scheduler werden benutzt, um die Zeitinformationen zu aktualisieren und dem neuen Projektzustand anzupassen. Dies ist nötig, damit die Zeitplanung weiterhin aktuell und konsistent mit dem Zustand des Schedulers ist. Es sollen genau diejenigen Lösungen in die Zeitplanung miteinfließen, die dem Benutzer noch zur Verfügung stehen. Die Zeitplanung während der Projektabwicklung berücksichtigt nur noch die aktuell zugelassenen Hyperpfade.

6.1.1 Treffen von Entscheidungen

Bekommt ein Agent eine komplexe Aufgabe zugeteilt, so muß er eine Methode zu ihrer Bearbeitung auswählen. Durch diese Festlegung auf eine der Methoden fallen die übrigen Alternativen weg und dürfen auch bei der Zeitanalyse nicht mehr berücksichtigt werden.

Jedesmal, wenn eine Methode ausgewählt wird, müssen bestimmte Lösungen deaktiviert werden. Es handelt sich dabei genau um diejenigen Lösungen, die die ausgewählte Methode verbieten, d.h. eine der anderen Alternativen benutzen. Diese Lösungen werden gesucht, mit OUT markiert und stehen dann für die Zeitpropagierung nicht mehr zur Verfügung. Durch diese Markierung ist erreicht, daß nicht mehr alle Lösungen neu berechnet werden müssen.

6.1.2 Rückzug von Entscheidungen

Der Anwender von CoMo-Kit trifft Entscheidungen unter bestimmten Rahmenbedingungen. Ändert sich diese Situation, so kann es nötig werden, eine Entscheidung wieder zurückzuziehen. CoMo-Kit stellt Mechanismen für derartige Rückzüge zur Verfügung [3].

Der Benutzer hat zwei Möglichkeiten, eine Entscheidung zurückzuziehen:

Retract: Durch die Entscheidung wurden alle übrigen Alternativen mit OUT markiert und bei den Zeitanalyse nicht mehr berücksichtigt. Der Rückzug dieser Entscheidung hat den Rückzug des gesamten Teilbaumes der betreffenden Methode zur Folge [3]. Anschließend stehen zur Lösung der Aufgabe wieder alle alternativen Methoden zur Verfügung; die entsprechenden Lösungen werden alle mit IN markiert.

Reject: Der Unterschied zu Retract besteht in der Verfügbarkeit der Methode nach dem Rückzug. Es besteht eine Notwendigkeit für den Rückzug [3]. Die zurückgezogene Methode kann nicht wieder für den Lösungsprozeß herangezogen werden. Die Lösungen zu allen übrigen Methoden werden wieder mit IN markiert; diejenigen, die zu der zurückgezogenen Methoden gehören, mit OUT.

Lösungen, die mit OUT markiert sind, können durch ein 'Unreject' wieder aktiviert werden.

6.2 Zeitanalyse auf abstrakter Ebene

Im vorigen Kapitel wurde atomaren Aufgaben eine Vorgangsdauer zugeordnet und die Zeitpropagierung fand auf der Ebene der atomaren Aufgaben statt. Eventuell ist jedoch die Aufgliederung der Teilaufgaben zu fein, um noch eine sinnvolle Zeitanalyse durchführen zu können. Wenn einer komplexen anstatt einer atomaren Aufgabe sinnvoll eine Vorgangsdauer zugeordnet werden kann, findet die Propagierung auf dieser höheren Hierarchieebene des Aufgaben-Methoden-Baumes statt. Zusätzlich wird Aufwand gespart, wenn die Teilbäume, die unter der betreffenden Aufgabe hängen, nicht mehr betrachtet werden müssen. Die Hyperpfade im Aufgaben-Methoden-Baum enden dann bei Knoten mit einer Angabe der Vorgangsdauer.

Konsistenzprüfung der Vorgangsdauern

Da bei der Berechnung alle alternativen Methoden und alle Teilaufgaben berücksichtigt werden sollen, muß gewährleistet sein, daß die Vorgangsdauer bei ausreichend vielen Vorgängen angegeben ist.

Hat eine Aufgabe eines bestimmten Abstraktionniveaus der Aufgaben-Methoden-Hierarchie eine Vorgangsdauer, so bedeutet das, daß die Zeitanalyse auf diesem Abstraktionsniveau stattfinden soll. Auf dem Pfad von der Wurzel bis zu diesem Knoten darf kein weiterer Vorgang eine Dauer haben. Dies würde bedeuten, daß sogar schon auf einer abstrakteren Ebene eine Dauer angegeben werden kann. Außerdem darf in dem Teilbaum, der unter diesem Vorgang hängt und durch ihn repräsentiert wird, ebenfalls keine Vorgangsdauer angegeben sein. Dies wäre eine überflüssige Information, da schon dem ganzen Teilbaum eine Dauer zugeordnet werden konnte.

Es läßt sich zusammenfassen, daß auf jedem Pfad von der Wurzel bis zu einem Blattknoten des Aufgaben-Methoden-Baumes genau eine Vorgangsdauer angegeben sein muß. Diese Tatsache wird von meinem Verfahren überprüft und bei Fehlern wird eine entsprechende Meldung ausgegeben.

Kapitel 7

Realisierung und Beispiel

Im ersten Abschnitt dieses Kapitels wird die konkrete Realisierung der Verwaltungsinformationen für die Lösungsmenge vorgestellt. Der zweite Abschnitt stellt die Vorgehensweise bei der Zeitplanung an einem Beispiel dar.

7.1 Realisierung der Lösungsmenge

In Abschnitt 5.2 wurde gezeigt, wie jede der Lösungen einen Hyperpfad im Aufgaben-Methoden-Baum repräsentiert. Der Scheduler hat eine Instanzen-Variable 'solutions', die die Menge aller gültigen Lösungsmöglichkeiten enthält. Jede einzelne Lösung enthält die Informationen, die in Abschnitt 5.2 angegeben sind.

Außerdem ist in solutions diejenige Lösung gekennzeichnet, die die schnellste Lösung darstellt. Sie wird dem Benutzer nach der Zeitpropagierung angezeigt, weil sie unter Zeitgesichtspunkten optimal ist.

Generierung der möglichen Lösungen

Die benötigten Verwaltungsinformationen in der Variable solutions werden durch den Algorithmus Generate_Solutions erzeugt. Der Algorithmus wird zunächst beschrieben und dann in einem PASCAL-ähnlichen Pseudo-Code notiert.

Für die Zeitpropagierung ist es nötig zu wissen, ob eine Aufgabe zu einer bestimmten Lösungsmöglichkeit gehört oder nicht. Um diese Information bereitzustellen, wird in jeder Aufgabe notiert, welche Methoden seit Projektbeginn ausgewählt werden mußten, damit diese Aufgabe Bestandteil der Lösung ist.

Dazu wird der Aufgaben-Methoden-Baum in Tiefensuche durchgegangen. Erreicht der Algorithmus eine Aufgabe, werden in ihr die Namen aller Methoden gespeichert, die auf dem Pfad von der Wurzel bis hierher gefunden wurden. Mit dieser Information kann später entschieden werden, ob die Aufgabe bei der Zeitpropagierung für eine bestimmt Lösung zu berücksichtigen ist oder nicht: Sind alle in der Aufgabe angegebenen Methoden in der Lösung mit 'true' gekennzeichnet, so ist die Aufgabe Bestandteil der Lösung.

Nach diesem Vorbereitungen kann die Variable solutions belegt werden. Erreicht der Algorithmus eine Aufgabe, muß berücksichtigt werden, daß mehrere Lösungen von dieser Aufgabe betroffen sind. Es stehen alternative Methoden zur Realisierung dieser Aufgabe zur Verfügung und alle betroffenen Lösungen müssen angepaßt und um die 'neu entdeckten' Alternativen erweitert werden. Stehen n alternative Methoden zur Verfügung, so muß die aktuelle Lösung n mal kopiert werden und jeweils um eine der neuen Methoden erweitert werden. Das bedeutet, daß die Lösung die Selektion der entsprechenden Methode erfordern muß (ausgewählteMethode,true) und alle übrigen Methoden verbieten muß ((sonstigeMethode_1,false),...,(sonstigeMethode_(n-1),false)). Die n neuen Lösungen ersetzen die ursprüngliche Lösung.

Diese Vorgehensweise ist im Algorithmus Generate_Solutions beschrieben:
Als m_1 bis m_n seien dabei die n Methoden bezeichnet, die zur Bearbeitung der Aufgabe i dienen können.

Generate_Solutions

Eingabe: alle Knoten eines Netzplanes

Ausgabe: Lösungsmenge solutions

$solutions\ :=\ \emptyset$
$for\ i\ :=\ alle\ Knoten\ des\ Netzplanes\ do$
$\quad if\ (solutions\ =\ \emptyset)$
$\quad\quad for\ j\ :=\ 1\ to\ n_i\ do$
$\quad\quad\quad solutions\ :=\ solutions \cup \{L_j\}$

$\quad else$
$\quad\quad begin$
$\quad\quad betroffene\ solutions_i\ :=\ \{s \in solutions \mid (m, true) \in s\ \ \forall m \in P_i\}$
$\quad\quad for\ s\ :=\ alle\ betroffenen\ solutions_i\ do$
$\quad\quad\quad begin$
$\quad\quad\quad for\ m\ :=\ 1\ to\ n_i\ do$
$\quad\quad\quad\quad solutions\ :=\ solutions \cup \{s \cup L_m\}$
$\quad\quad\quad solutions\ :=\ solution \setminus s$
$\quad\quad\quad end$
$\quad\quad end$

$wobei\ gilt:$

$$L_i\ :=\ \{(M_j, bool) \mid j = 1, 2, ..., n;\ bool = \begin{cases} true & \text{für } j = i \\ false & \text{sonst} \end{cases}\}$$

$P_i\ :=\ Menge\ aller\ Methoden\ des\ Pfades\ von\ der\ Wurzel\ bis\ zur\ Aufgabe\ i$

$M_i\ :=\ Menge\ aller\ alternativer\ Methoden\ zu\ Aufgabe\ i$
$n_i\ :=\ |M_i|$

7.2 Beispiel

In diesem Abschnitt sind die beschriebenen Vorgehensweisen an einem Beispiel erläutert. Dabei wird ein konzeptuelles Modell verwandt, dessen Aufgaben-Methoden-Hierarchie in Abbildung 7.1 angegeben ist.

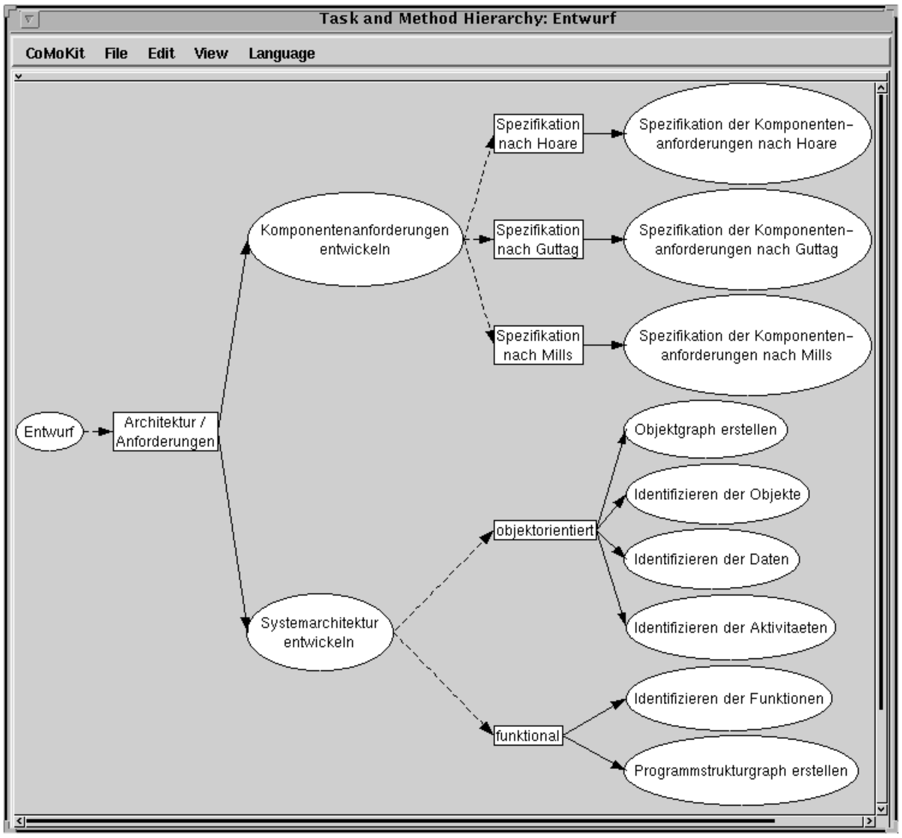

Abbildung 7.1: Konzeptuelles Modell 'Entwurf'

In den Abschnitten 5.3.1 und 6.2 ist beschrieben, weshalb bestimmten Aufgaben des konzeptuellen Modells eine Vorgangsdauer zugeordnet werden muß. Dies geschieht mit dem Menüpunkt 'Edit duration' im Edit-Menü des Fensters mit der Aufgaben-Methoden-Hierarchie. Abbildung 7.2 zeigt das Fenster zum Editieren der Vorgangsdauer.

Die geschätzte Dauer des Vorgangs 'Systemarchitektur entwickeln' in Tagen wird in das entsprechende Feld eingetragen. Wird 'no duration' angewählt, so bedeutet dies, daß die Dauer nicht zuverlässig geschätzt werden kann. Entweder muß dann auf einem niedrigerem Abstraktionsniveau allen Teilaufgaben der Aufgabe 'Systemarchitektur entwickeln' eine Dauer zugeordnet werden oder man ist sogar in der Lage, eine abstraktere Aufgabe abzuschätzen.

Anschließend kann die Zeitpropagierung gestartet werden, indem man den Menüpunkt 'propagate time' auswählt. Die Zeitanalyse wird durchgeführt und die ermittelten Zeitpunkte werden in den Knoten gespeichert. Das System zeigt an, welche Methoden ausgewählt werden müssen, um zu der schnellsten Lösung zu gelangen. Alle Fenster, die Aufgaben graphisch darstellen, können umgeschaltet

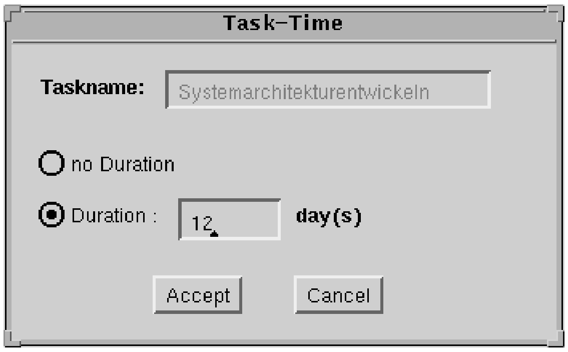

Abbildung 7.2: Editieren der Vorgangsdauer

werden, um die Zeitinformationen in den Aufgaben-Knoten zu veranschaulichen.

In Abbildung 7.3 ist die Aufgaben-Methoden-Hierarchie abgebildet. Im Beispiel sind nur zu den beiden Aufgaben 'Systemarchitektur entwickeln' und 'Komponentenanforderungen entwickeln' Vorgangsdauern angegeben. Sie müssen in dieser Reihenfolge ausgeführt werden, um die Methode 'Architektur / Anforderungen' zu realisieren. Auf dieser Abstraktionsebene der Aufgaben-Methoden-Hierarchie findet die Zeitanalyse statt. Bei den Aufgaben sind die Vorgangsdauern und die fünf Zeitpunkte FAET, FEET, FELT, SALT und SELT angegeben. Speziellere Aufgaben haben keine Zeitangaben.

Zu einer detaillierteren Zeitanalyse sind nun bei allen Teilaufgaben der komplexen Aufgabe 'Systemarchitektur entwickeln' Vorgangsdauern angegeben. Die resultierenden Termine im Datenfluß der Methode 'objektorientiert' sind in Abbildung 7.4 aufgeführt.

Wünscht der Anwender die ursprüngliche übersichtlichere Darstellung, so kann er sich die Zeitinformationen für eine einzelne Aufgabe oder Methode in einem getrennten Fenster anzeigen lassen. Ein Beispiel ist in Abbildung 7.5 angegeben.

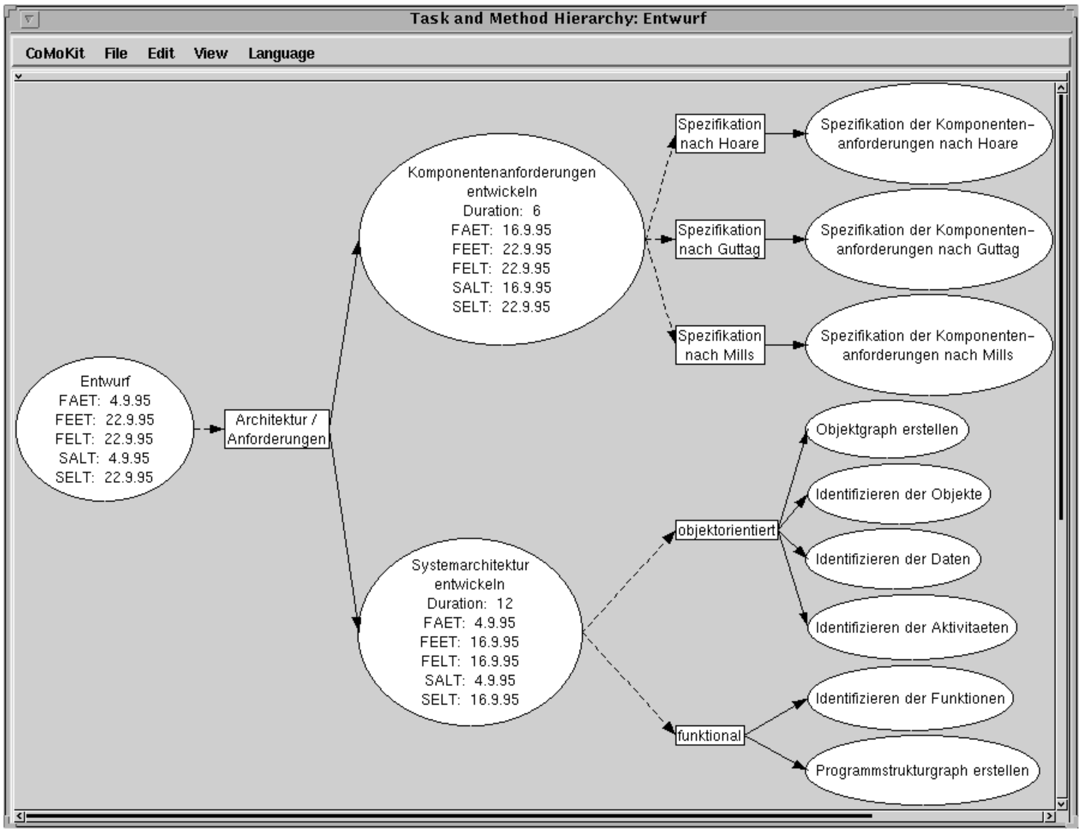

Abbildung 7.3: Konzeptuelles Modell mit Zeitinformationen

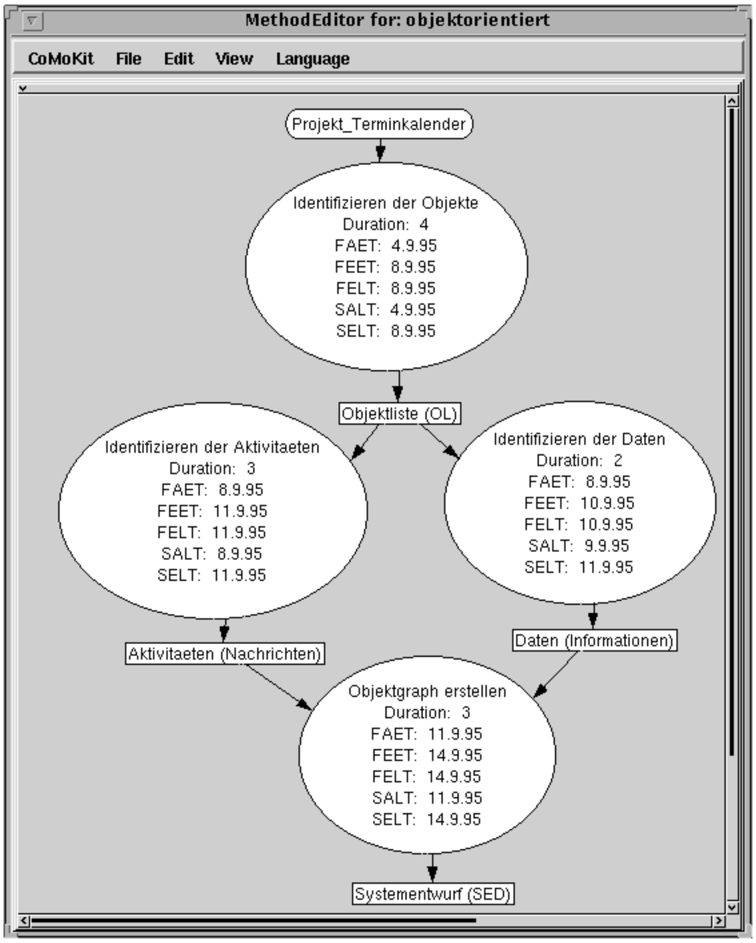

Abbildung 7.4: Informationsfluß mit Zeitinformationen

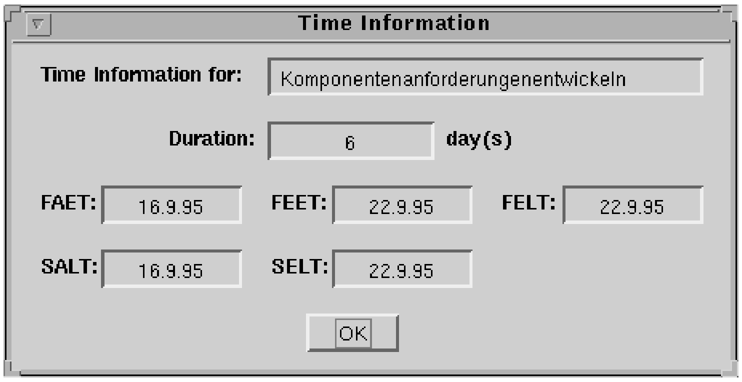

Abbildung 7.5: Anzeige der Zeitinformationen

Kapitel 8

Zusammenfassung und Ausblick

Dieses Kapitel faßt die Inhalte der Arbeit kurz zusammen und gibt einen Ausblick auf mögliche Erweiterungen der Zeitplanungskomponente.

8.1 Zusammenfassung

Durch diese Arbeit wurde das Modellierungswerkzeug CoMo-Kit um eine Zeitplanungkomponente erweitert. Um die Dauer der einzelnen Aufgaben abzuschätzen, werden ihnen Vorgangsdauern zugeordnet. Die nach dem Prinzip der Netzplantechnik durchgeführte Zeitanalyse legt einen Zeitrahmen für die Ausführung der Aufgaben fest: Jeder Vorgang erhält einen frühesten und spätesten Anfangs- und Endzeitpunkt. Diese Zeitinformationen werden zu den abstrakten Aufgaben hin verallgemeinert, um schließlich einen Zeitplan des gesamten Projektes festzulegen.

Dem Projektmanager wird dadurch ein wichtiges Hilfsmittel zur Planung an die Hand gegeben. Er kann zeitliche Abhängigkeiten zwischen den Aufgaben erkennen und Ursachen für Projektverzögerungen ermitteln. Um eine Beschleunigung zu erreichen, können Betriebsmittel gezielt eingesetzt werden.

Die in Abschnitt 1.2 genannten Anforderungen wurden durch das im Rahmen dieser Diplomarbeit entwickelten Verfahren der Allgemeinen Hierarchischen Netzplantechnik erfüllt. CoMo-Kit eignet sich dadurch besser sowohl zur Planung **vor** dem Projektablauf, als auch zur Umplanung **während** der Abarbeitung.

Die Ergebnisse der Zeitplanung mit Hilfe der Netzplantechnik sind wie bei jeder zukunftsorientierten Planung mit Vorbehalten zu sehen. Da die verwendeten Vorgangsdauern lediglich Schätzwerte sind, kann in vielen Fällen nicht die exakte Einhaltung der Planwerte garantiert werden. Die detaillierten Resultate dieser Verfahren dürfen nicht über die Unsicherheit der Ausgangswerte und damit auch die der Ergebnisse hinwegtäuschen.

Die zusätzliche Funktionalität von CoMo-Kit bringt einen erhöhten Berechnungsaufwand mit sich. Durch die große Ausdrucksstärke der zugelassenen konzeptuellen Modelle (vg. Abschnitt 4.1 über offene Subnetzpläne) und die Möglichkeit von umfangreichen zeitlichen Beziehungen ist der Analyseaufwand höher als bei den traditionellen Verfahren der Netzplantechnik.

8.2 Ausblick

Dieser Abschnitt gibt einen Ausblick auf die möglichen Erweiterungen der Zeitplanungskomponente in CoMo-Kit.

8.2.1 Berücksichtigung von Kosten

Diese Arbeit berücksichtigt bei der Zeitanalyse keine Kosten. Durch Einbindung von **zeitabhängigen** Kosten können folgende Ziele erreicht werden [25]:

- Kostenkontrolle durch einen Soll-Ist-Vergleich
- Bereitstellung von Informationen für die Kostenrechnung
- Ermittlung von Produktpreisen

8.2.2 Einführung von Constraints

Sind bei der Abarbeitung eines konzeptuellen Modells mehrere Entscheidungen zu treffen, so können diese Entscheidungen voneinander abhängig sein. Das Beispiel in Abbildung 8.1 veranschaulicht diese Abhängigkeiten.

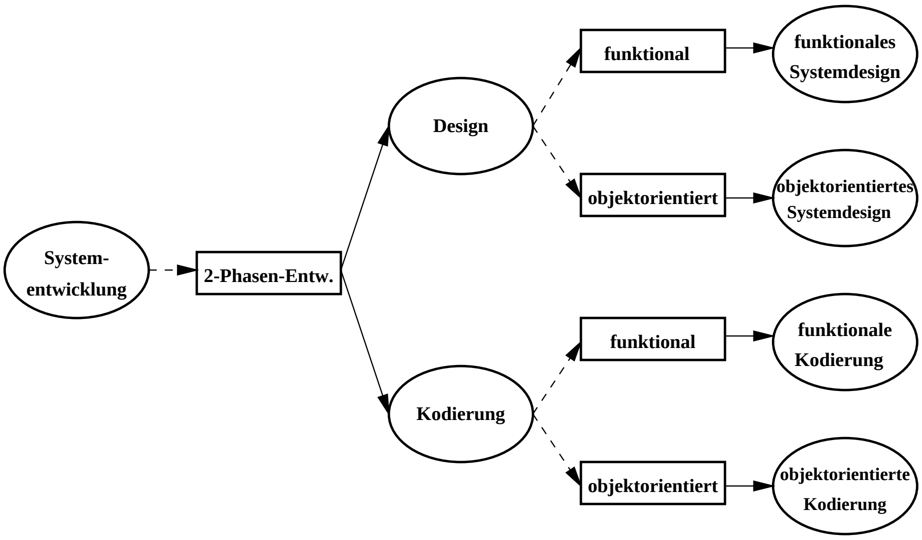

Abbildung 8.1: Konzeptuelles Modell: Systementwicklung

Nach einem objektorientierten Design muß auch eine objektorientierte Kodierung erfolgen. Entscheidungen können sich also gegenseitig beeinflussen. Dieses Problem ließe sich im Beispiel durch eine andere Modellierung beseitigen. Aber insbesondere wenn man an einer möglichst natürlichen Wissensmodellierung interessiert ist, lassen sich diese Abhängigkeiten nicht vermeiden.

Diese Beziehungen können in der aktuellen Version von CoMo-Kit nicht dargestellt werden. Es ist geplant diese Abhängigkeiten als Constraints einzuführen. Das Verfahren der Allgemeinen Hierarchischen Netzplanung kann diese Constraints verarbeiten, wenn einzelne Lösungen (z.B. objektorientiertes Design in Verbindung mit funktionaler Kodierung) ausgeschlossen werden. Diese Auswahlmöglichkeiten stehen dann für die Zeitanalyse nicht mehr zur Verfügung.

8.2.3 Verkürzung der Projektdauer

Es wurde bereits beschrieben, daß zur Kürzung der Projektdauer diejenigen Vorgänge beschleunigt werden müssen, die auf dem kritischen Pfad liegen. Sie zeichnen sich dadurch aus, daß ihre frühesten und spätesten Zeitpunkte identisch sind. Der Algorithmus von Ford-Fulkerson kann verwendet werden, um die Netzplanverkürzung kostenoptimal durchzuführen [7, 8]. In [26] ist ein Algorithmus beschrieben, der diejenigen Vorgänge kürzt, deren Beschleunigung sich auf möglichst viele Pfade auswirkt. Eine mögliche Erweiterung wäre eine deutlichere Hervorhebung der kritischen Vorgänge in der Aufgaben-Methoden-Hierarchie und Hilfsmittel, um deren Verkürzung zu unterstützen. Verkürzungen können erreicht werden durch [25]:

- den Abbau von Sicherheitsreserven in der Ausführungszeit,
- den erhöhten Einsatz von Betriebsmitteln,
- eine zeitliche Überlappung von Vorgängen und
- der Änderung der Ablaufstruktur.

Ein zusätzliche Schwierigkeit ergibt sich, wenn durch die Verkürzung der Vorgangsdauer andere Pfade kritisch werden.

Literaturverzeichnis

[1] G. Altrogge. *Netzplantechnik.* R. Oldenbourg Verlag, München, 1994.

[2] W. B. Crowston. Decision CPM: Network Reduction and Solution. *Operational Research Quaterly*, 21:435–452, 1970.

[3] B. Dellen. Verwaltung von Abhängigkeiten bei der Operationalisierung konzeptueller Modelle. *Diplomarbeit Universität Kaiserslautern*, 1994.

[4] H. Eisner. A Generalized Network Approach to the Planning and Scheduling of a Research Project. *Operation Research*, 10:115–125, 1961.

[5] S. E. Elmaghraby. An Algebra For The Analysis of Generalized Activity Networks. *Management Science*, 10:494–514, 1964.

[6] S. E. Elmaghraby. Some Network Models. *Managemant Science*, 29, 1970.

[7] Ford and D. R. Fulkerson. Flow in Networks. *Princeton, New Jersey*, 1962.

[8] D. R. Fulkerson. A network flow computation for project cost curves. *Management Science*, 7, 1961.

[9] L. Hennicke. *Wissensbasierte Erweiterung der Netzplantechnik.* Physika-Verlag, Heidelberg, 1991.

[10] G. Holzschuh. *Was ist Netzplantechnik ?* Elitera-Verlag, 1973.

[11] IBM. 1440 Project Control System 1140-MX-02X Users Manual, Application Program. *IBM-Bibliothek H 20-0103-0.*

[12] R. Kaerkes, R. H. Möhring, W. Oberschelp, F. J. Radermacher, and M. M. Richter. Mathematische Untersuchungen zur stochastischen Kapazitätsoptimierung. Technical report, Abschlußbericht des Forschungsvorhabens, Lehrstuhl für Mathematik, Lehrstuhl für Angewandte Mathematik, RWTH Aachen, December 1981.

[13] J. E. Kelley. Critical-path planning and scheduling: Mathematical basis. *Operations Research*, 9:296–320, 1961.

[14] H. Lüttgen. Reduktion und Dekomposition von Planungsnetzen. *Ablauf und Planungsforschung*, 7:92–104, 1966.

[15] H. Lüttgen. Reduktion und Dekomposition von PERT-Netzen. *Ablauf und Planungsforschung*, 8:370–379, 1967.

[16] F. Maurer. Hypermediabasiertes Knowledge Engineering für verteilte wissensbasierte Systeme. *Dissertation Universität Kaiserslautern*, 1993.

[17] F. Maurer and G Pews. Ein Knowledge-Engineering-Ansatz für kooperativen Design am Beispiel Bebauungsplanung. *Künstliche Intelligenz, Themenheft Knowledge Engineering*, 1:28–34, January 1995.

[18] S. C. Parikh and W. S. Jewell. Decomposition of Project Networks. *Journal of the Institute of Management Science*, 11:444–459, 1965.

[19] C. R. Phillips. Fifteen Key Features of Computer Programs for CPM and PERT. *The Journal of Industrial Engineering*, XV(1), 1964.

[20] R. S. Pressman. *Software Engineering, A Practitioner's Approach.* Mc Graw Hill, 1982.

[21] A. Pritsker and G. E. Whitehouse. GERT, Graphical Evaluation and REview Technique, Part II. *Probabilistic and Industrial Engineering Applications, Santa Monica*, 1966.

[22] A. Schmitz, H. J. Stommel, and W. Theis. Auswählen und Verdichten von Projektinformationen in Netzplanform mit dem Netzplanprogrammsystem MANDAS. *Zeitschrift für wirtschaftliche Fertigung*, 12:662–665, 1973.

[23] W.-G. Schmitz. Multiple Aufgabezerlegung von konzeptuellen Modellen. *Diplomarbeit Universität Kaiserslautern*, 1994.

[24] J. Schwarze. Simulation von Netzplänen mit stochastischen und unabhängigen Vorgangsdauern. In *Operation Research Proceedings*, pages 190–197, Berlin, Heidelberg, New York, 1980.

[25] J. Schwarze. *Netzplantechnik.* Verlag Neue Wirtschafts-Briefe, Herne/Berlin, 1986.

[26] R. E. Thompson. Adjusting Network Plans with 'PERT Slack Bonus'. *The Journal of Industrial Engineering*, 17:145–149, January 1966.

[27] N. Thumb. *Grundlagen und Praxis der Netzplantechnik.* Verlag Moderne Industrie, München, 2 edition, 1969.

[28] B. Wielinga, G. Schreiber, and J. Breuker. KADS a modelling approach to knowledge engineering. In G. Schreiber, editor, *Special Issue: The KADS approach to Knowledge Engineering, Knowledge Acquisition*, volume 4. Academic Press, 1992.

[29] S. Zelewski. Ansätze der Künstlichen Intelligenz-Forschung zur Unterstützung der Netzplantechnik. *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung*, 12:1112–1129, 1988.

[30] H.-J. Zimmermann. *Netzplantechnik*. Walter de Gruyter Co., Berlin, 1971.