



ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

**Fakultät für
Ingenieurwissenschaften
und Informatik**
Institut für Datenbanken und In-
formationssysteme

Zeitliche Darstellung und Modellierung von Prozessen mithilfe von Gantt-Diagrammen

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Martin Sommer
martin.sommer@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Andreas Lanz

2012

Zusammenfassung

Die Dauern von Aktivitäten, die dazugehörigen Abstände zwischen Aktivitäten und die daraus resultierenden Anfangs- und Endzeitpunkte der Aktivitäten spielen in Prozessen eine wichtige Rolle. Deswegen wird eine einfache und übersichtliche Darstellungsmöglichkeit gesucht, mit der Zeiten in blockbasierten Prozesse einfach dargestellt werden können. Diese Darstellung sollte dabei so viele Zeitattribute eines Prozesses wie möglich abbilden (minimale/maximale Dauer, maximaler/minimaler Abstand und Anfangs- und Endzeitpunkte). Dabei darf jedoch keinesfalls die Übersichtlichkeit und Ablesbarkeit verloren gehen. In dieser Arbeit hat sich gezeigt, dass Gantt-Diagramme dafür sehr gut geeignet sind. Jedoch sind die klassischen Gantt-Diagramme nur beschränkt einsetzbar, da sie nicht alle Eigenschaften abdecken können, die in Prozessen verwendet werden. Deswegen ist es notwendig die klassischen Gantt-Diagramme um weitere Elemente zu erweitern, damit man alle Elemente blockbasierter Prozesse auf Gantt-Diagramme gebildet werden können.

Danksagung: Ich danke Andreas Lanz für die Betreuung und Prof. Dr. Manfred Reichert für die Begutachtung dieser Bachelorarbeit.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung und Zielsetzung	2
1.2	Aufbau der Arbeit	2
2	Grundlagen	5
2.1	ADEPT-Metamodell	6
2.2	Zeitaspekte in Prozessen	8
2.2.1	Dauer und Zeitabstände zwischen von Aktivitäten	9
2.2.2	Aktivitäten mit festen Terminen	11
2.2.3	Wiederholende Aktivitäten - Schleifen	11
2.3	Anforderungen an die Zeit-Darstellung	12
3	Repräsentation von Zeit	15
3.1	Gantt-Diagramme	15
3.2	AsbruView	16
3.3	Netzplantechnik	17
3.3.1	CPM (Critical Path Method)	18
3.3.2	MPM (Metra Potential Method)	18
3.3.3	PERT (Program Evaluation and Review Technique)	20
3.4	Projekt Management Werkzeuge	21
3.4.1	Visio	21
3.4.2	A-Plan	22
3.4.3	Acos Plus	23
3.5	Fazit	23

4	Abbildung blockstrukturierter Prozessgraphen auf Gantt-Diagramme	25
4.1	Abbildung von Prozessgraphen auf klassische Gantt-Diagramme	26
4.2	Vorüberlegungen	27
4.3	Knotenattribute	28
4.4	Sequenzen und Parallelität	30
4.4.1	Abbildung von Sequenzen	30
4.4.2	Abbildungen von Zeitkanten	31
4.4.3	Parallele und zusammengefasste Zeitkanten	36
4.4.4	Parallelität	37
4.5	XOR-Entscheidungen und Schleifen	40
4.5.1	XOR-Entscheidungen	40
4.5.2	Schleifen	44
4.6	Beispiel	47
5	Verhalten von eGantt auf Ausführungsebene	49
5.1	Knotenzustände in ADEPT und eGantt	50
5.2	eGantt-Zeitlinie	52
5.3	Verhalten der eGantt-Diagramme auf Ausführungsebene	53
6	Prototyp und Implementierung	63
6.1	Technische Ausführung	63
6.2	Prototyp	66
7	Fazit und Ausblick	69
7.1	Fazit	69
7.2	Ausblick	70
	Abbildungsverzeichnis	73
	Literaturverzeichnis	77

1

Einleitung

Durch den Konkurrenzdruck von Ländern wie China oder Indien ist es für Unternehmen immer wichtiger sich im Wettbewerb gegen diese Länder zu behaupten. Daher gewinnt die Optimierung und Darstellung von Prozessen in den einzelnen Unternehmen immer mehr an Bedeutung [BHM⁺06].

Dabei beinhalten diese Prozesse häufig Zeitaspekte die es zu beachten gilt. Diese zeitlichen Eigenschaften müssen dabei von beliebigen Personen überwacht werden können. Daher wird die Unterstützung der Zeitaspekte und Zeitabhängigkeiten durch Prozess-Management-Systeme (PMS) in der Zukunft eine immer größere Rolle spielen. Momentan existieren aber kaum Möglichkeiten um Zeitaspekte in einem aktivitätenbasierten Prozess darzustellen und so eine einfache Ablesbarkeit der Anfangs- und Endzeitpunkte der einzelnen Aktivitäten zu ermöglichen. Zudem werden in den meisten existierenden Zeit-Darstellungen nicht alle Eigenschaften eines PMS unterstützt. Daher ist das Ziel dieser Arbeit die Entwicklung

einer Darstellungsmöglichkeit für blockbasierte Prozesse, welche Zeitaspekte beinhalten zu entwickeln.

Diese Darstellungsmöglichkeit sollte einen Prozess auf zwei verschiedenen Ebenen darstellen können. Zum einem die Modellierung eines Prozess und zum anderem die Ausführung. Bei der Modellierung stellt die Darstellungsmöglichkeit den allgemeinen Fall eines Prozesses dar. Durch diesen allgemeinen Fall, kann bei der Ausführung eines Prozesses separate Fälle betrachten und ein einmaliger modellierter Prozess kann öfters wiederverwendet werden.

1.1 Aufgabenstellung und Zielsetzung

Ziel dieser Bachelorarbeit ist es, eine Möglichkeit zur Darstellung der zeitlichen Eigenschaften eines Prozesses zu entwickeln. Dabei hat sich herausgestellt, dass Gantt-Diagramme als Grundlage für eine übersichtliche Darstellung am besten geeignet sind. Deswegen werden in dieser Arbeit einzelne Prozesse bzw. Prozess-Instanzen auf Gantt-Diagramme abgebildet. Für die Modellierung der blockbasierten Prozesse wird das ADEPT-Metamodell [Rei00] als Grundlage herangezogen. Die Abbildung muss dabei neben den Prozesskonstrukten Sequenz, Parallelität, XOR-Entscheidung und Schleife auch die Zeitaspekte der minimale-/maximale Dauer einer Aktivität, minimalen/maximalen Abstand zwischen Aktivitäten und Anfangs- und Endpunkte einer Aktivität unterstützen.

Nachdem die Abbildungen eines Prozess auf die Gantt-Diagramme entwickelt wurde, soll anschließend auch das Verhalten dieser Diagramme bei der Prozessausführung näher untersucht werden.

Des Weiteren soll am Ende ein Prototyp entwickelt werden, um die Realisierbarkeit und Machbarkeit der entwickelnden Konzepte zu demonstrieren.

1.2 Aufbau der Arbeit

Diese Bachelorarbeit ist in 7 Kapiteln unterteilt. Nach der Einleitung, wird in Kapitel 2 auf die Grundlagen eines Prozess-Metamodells am Beispiel des ADEPT-Metamodell eingegangen und einzelne Zeitaspekte von Prozessen und die daraus resultierenden Probleme diskutiert.

In Kapitel 3 werden existierende Repräsentationen von Zeit sowie Projekt Management Werkzeuge vorgestellt und auf ihre Eignung untersucht. Schließlich wird ein Fazit über die Repräsentationen gezogen. In den Hauptkapiteln dieser Arbeit 4 & 5 wird zuerst die Abbildungen von blockstrukturierte Prozesse auf Gantt-Diagramme vorgestellt und im Kapitel 5 das Verhalten dieser Abbildungen bei der Prozessausführung untersucht. In den letzten beiden Kapiteln wird der Prototyp diskutiert und ein Fazit mit Ausblick über die gesamte Arbeit gezogen.

2

Grundlagen

Ziel dieser Arbeit ist es, blockstrukturierte Prozesse auf Gantt-Diagramme abzubilden. Diese blockstrukturierte Prozesse müssen dabei in einem Prozess-Metamodell modelliert sein. Als Prozess-Metamodell wird in dieser Arbeit das ADEPT-Metamodell der Universität Ulm genommen [Rei00]. In Abschnitt 2.1 werden die Grundlagen des ADEPT-Metamodell besprochen. Es wird auf die einzelnen Elemente des Metamodells eingegangen und einzelne Konstrukte vorgestellt.

Im Abschnitt 2.2 werden anschließend einzelne Zeitaspekte von Prozesse behandelt, welche in den Gantt-Diagramme später dargestellt werden sollen. Zudem werden in diesem Abschnitt blockstrukturierte Prozess-Metamodelle um Zeitaspekte erweitert.

Im letzten Abschnitt 2.3 werden die Anforderungen definiert, welche die Abbildung der Prozesse auf Gantt-Diagramme später erfüllen muss.

2.1 ADEPT-Metamodell

Das ADEPT-Modell [Rei00] ist ein blockstrukturiertes Prozess-Metamodell, welches an der Universität Ulm entwickelt wurde. Dieses Metamodell definiert Konstrukte die bei fast allen Basis-Modellen zu finden sind. Zu diesen Konstrukten gehören beispielsweise Aktivitäten, Kontrollkanten, Sequenzen, Parallelität und Entscheidungen. Im weiteren werden nun die einzelnen Konstrukte und Begriffe die in dieser Arbeit von Interesse sind diskutiert.

Prozess Ein Prozess auch Teils *Workflow* genannt, ist eine formale Sicht auf eine Folge von Schritten um ein Ziel zu erreichen. Dieses Ziel wird meist in durch einen Ablauf von Prozessaktivitäten die Verbunden sind erreicht. Eine Darstellung eines Prozesses und dessen Ablauf ist in Abbildung 2.1 zu sehen.

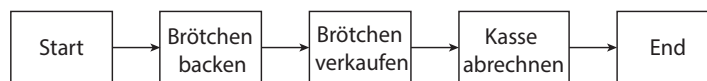


Abbildung 2.1: Einfacher Prozess in ADEPT

Prozessschema Ein Prozessschema ist die Darstellung eines Prozesses, welche in einem Prozess-Management-System modelliert beziehungsweise ausgeführt wird. Dieses Prozessschema besteht meist aus mehreren Aktivitäten und deren Verbindungen. Zudem werden einzelne Informationen über die Aktivitäten bereitgestellt.

Prozessinstanz Eine Prozessinstanz auch *Instanz* genannt, repräsentiert eine Ausführung eines Prozesses. Jede Instanz ist einmalig und hat ihre eigene Zustände. Eine Prozessinstanz ist Vergleichbar mit einer Objektinstanz in der objektorientierten Programmierung.

Aktivität Eine Aktivität, auch Knoten genannt beschreibt einen Arbeitsschritt. Dieser Arbeitsschritt ist Teil innerhalb eines Prozesses. Aktivitäten können Manuell, Automatisch oder durch einen Benutzer ausgeführt werden. In ADEPT werden Aktivitäten als Rechtecke dargestellt (siehe Abbildung 2.2 Nummer 1).

Kontrollfluss Ein Kontrollfluss beschreibt die Abfolge einzelner Aktivitäten in einem Prozess. Diese Abfolge beschreibt bei der Ausführung die Reihenfolge der ausgeführten Aktivitäten.

Der Kontrollfluss wird in ADEPT durch Kontrollkanten zwischen Aktivitäten (siehe Abbildung 2.2 Nummer 2), welche als durchgezogener Pfeil dargestellt sind definiert.

Sequenz Eine Sequenz ist das einfachste Konstrukt in einem Prozess. Es stellt dabei zwei oder mehrere nacheinander ausgeführte Aktivitäten dar (siehe Abbildung 2.2 Nummer 3). Somit besitzt mindestens eine Aktivität einen Nachfolger.

Parallelität Eine Parallelität besteht aus zwei oder mehreren Aktivitäten die parallel ablaufen. In ADEPT wird die Parallelität durch einen AND-Block realisiert. Dieser AND-Block besteht aus einem AND-Split und einem dazugehörigen AND-Join (siehe Abbildung 2.2 Nummer 4). Das AND-Split teilt den eingehenden Kontrollfluss mehrere ausgehende Pfade (auch Zweige genannt) auf. Das AND-Join verbindet alle eingehenden Pfade zu einem ausgehenden Pfad. Die ausgehenden Pfade des AND-Split und die eingehenden Pfade des dazugehörigen AND-Join müssen die selbe Anzahl haben.

Entscheidung Eine Entscheidung, auch XOR-Entscheidung oder Auswahl genannt ist analog wie die Parallelität aus einem XOR-Split und einem dazugehörigen XOR-Join aufgebaut. Dieser XOR-Split und das XOR-Join wird als XOR-Block bezeichnet (siehe Abbildung 2.2 Nummer 5). In einer XOR-Entscheidung wird beim XOR-Split genau ein ausgehender Pfad ausgewählt bzw. alle anderen ausgehenden Pfade werden abgewählt. Wichtig ist es wieder, dass bei dem dazugehörigen XOR-Join die gleiche Anzahl von Pfade eingehen, wie bei dem XOR-Split ausgehen.

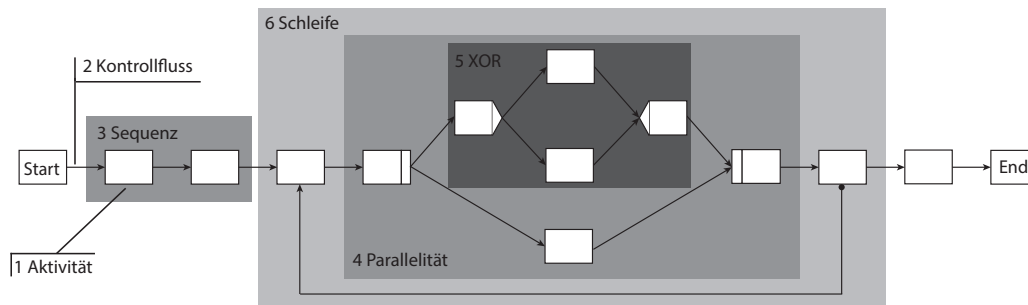


Abbildung 2.2: Basiselemente des ADEPT-Metamodell

Schleife Eine Schleife ist eine Wiederholung von Sequenzen, Parallelitäten, Entscheidungen und Kombinationen daraus. Ein Schleifen-Block (siehe Abbildung 2.2 Nummer 6) wird durch

ein *Start-Loop* gestartet und durch ein *End-Loop* beendet. In ADEPT, wird von dem End-Loopknoten eine Kontrollkante zu dem Start-Loopknoten gezogen und so der Schleifen-Block dargestellt. Bei der Modellierung ist zu Beginn noch nicht bekannt, wie oft die einzelne Schleife wiederholt wird. Bei Erreichen des End-Loopknoten wird jeweils entschieden, ob die Schleife verlassen wird oder nochmals eine Iteration durchgeführt wird.

2.2 Zeitaspekte in Prozessen

Neben der bisher beschriebenen Kontrollfluss-Sicht, existieren auch weitere Sichten auf Prozesse (z.B. *Daten-Sicht*, *Ressourcen-Sicht* und *Operative-Sicht* [vv02][JB96]). In [LWR10] wurden darüber hinaus die wichtigen Elemente der *Zeitbasierten-Sicht* erfasst und beschrieben. Dabei wurden 10 verschiedene *Zeit-Patterns* (TP) identifiziert, welche in 4 übergeordneten Kategorien einsortiert sind. Diese in Tabelle 2.3 dargestellt.

Pattern Catalogue
Category I: Durations and Time Lags
TP1: Time Lags between two Activities
TP2: Durations
TP3: Time Lags between Arbitrary Events
Category II: Restricting Execution Times
TP4: Fixed Date Elements
TP5: Schedule Restricted Elements
TP6: Time-based Restrictions
TP7: Validity Period
Category III: Variability
TP8: Time-dependent Variability
Category IV: Recurrent Process Elements
TP9: Cyclic Elements
TP10: Periodicity

Abbildung 2.3: Pattern Kategorie [LWR10]

In dieser Arbeit sind hauptsächlich die *Zeit-Patterns* TP1, TP2, TP4 und TP9 von Bedeutung. Diese vier Patterns sind für die Umsetzung der Abbildungen auf Gantt-Diagramme essentiell. Diese einzelne Zeitaspekte werden im nächsten Abschnitt näher gezeigt.

2.2.1 Dauer und Zeitabstände zwischen von Aktivitäten

In der der *Zeitbasierten-Sicht* besitzt eine Aktivität eine minimale und eine maximale Dauer. Diese Dauer wird in Zeiteinheiten wie z.B. Stunden (h), Minuten (min), usw. angegeben. Die maximale Dauer einer Aktivität muss dabei gleich oder größer wie die minimale Dauer sein. Wie in Abbildung 2.4 zu sehen ist, wird eine Aktivität mit einer minimaleren Dauer (minD) und maximalen Dauer (maxD) versehen.

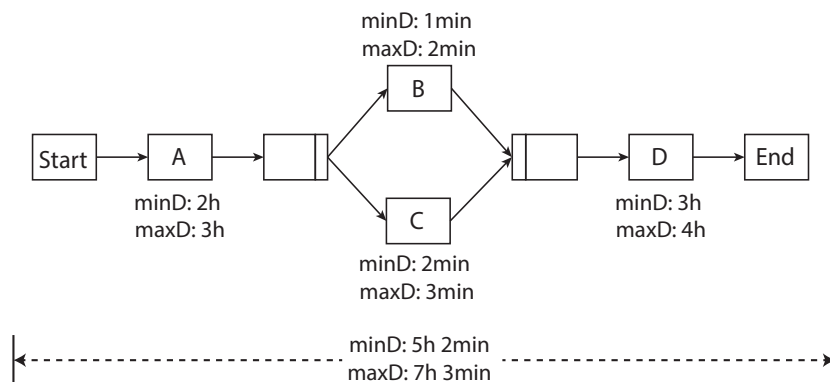


Abbildung 2.4: Aktivitäten- und Prozessdauer

Neben minimaler und maximaler Dauer existieren Zeitkanten, welche zeitliche Abstände zwischen zwei Aktivitäten repräsentieren (siehe Abbildung 2.5). Diese Zeitkanten besitzen jeweils drei Attribute: einen minimalen Abstand, einen maximalen Abstand und die Art der Abhängigkeit der beiden Aktivitäten. Zudem ist die Art der Abhängigkeiten zwischen den Aktivitäten eine der folgenden:

- *Start-Start*: Der zeitliche Abstand ist vom Startzeitpunkt der Startaktivität zum Startzeitpunkt der Zielaktivität definiert. In ADEPT startet die Zeitkante an der linken Kante der Startaktivität und endet an der linken Kante der Zielaktivität.
- *Start-End*: Der zeitliche Abstand ist vom Startzeitpunkt der Startaktivität zum Endzeitpunkt der Zielaktivität definiert. In ADEPT startet die Zeitkante an der linken Kante der Startaktivität und endet an der rechten Kante der Zielaktivität.

- *End-Start*: Der zeitliche Abstand ist vom Endzeitpunkt der Startaktivität und dem Startzeitpunkt der Zielaktivität definiert. In ADEPT startet die Zeitkante an der rechten Kante der Startaktivität und endet an der linken Kante der Zielaktivität.
- *End-End*: Der zeitliche Abstand ist vom Endzeitpunkt der Startaktivität und dem Endzeitpunkt der Zielaktivität definiert. In ADEPT startet die Zeitkante an der rechten Kante der Startaktivität und endet an der rechten Kante der Zielaktivität.

Je nach Abhängigkeit der beiden Aktivitäten ist der minimale/maximale Abstand verschieden definiert. Der minimale Abstand ist die Verzögerung zwischen zwei Aktivitäten und deren Start- bzw. Endpunkten, welcher eintreten muss (z.B. bei *Start-Start* wird der Abstand zwischen den Startzeitpunkten der beiden Aktivitäten definiert). Der maximale Abstand, ist die Verzögerung die eintreten kann. Bei Erreichen des Maximums des maximalen Abstandes muss eine Aktivität ausgeführt werden. Analog wie bei der Dauer, muss der maximale Abstand größer oder gleich dem minimalen Abstand sein.

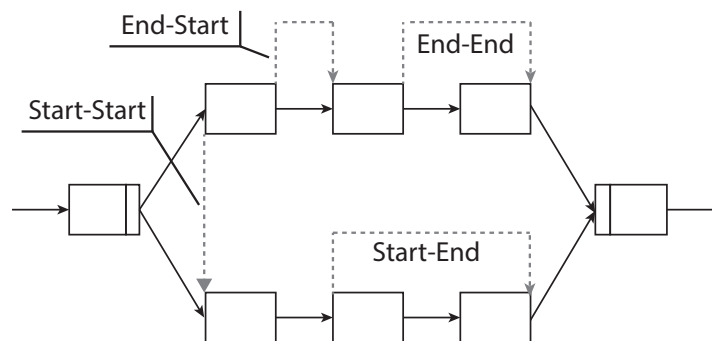


Abbildung 2.5: Prozess mit Zeitkanten in ADEPT

Prozesse besitzen ebenfalls eine minimale und eine maximale Dauer. Diese Dauer setzt sich aus den einzelnen Dauern der Aktivitäten und den zeitlichen Abständen zwischen diesen zusammen. Die maximale Dauer des Prozesses ist der zeitlich schlechteste Weg und die minimale Dauer des Prozesses der zeitlich beste Weg durch einen Prozess. Wie in Abbildung 2.4 zu sehen ist, ist die minimale und maximale Dauer des Prozess von der Startaktivität bis zur Endaktivität gemessen.

Durch die minimale/maximale Dauer einer Aktivität und den minimalen/maximalen Abstand einer Aktivität resultieren für eine Aktivität verschiedene Anfangs- und Endzeitpunkte. Diese allgemeine Anfangs- und Endzeitpunkte sind in vier Zeitpunkte unterteilt. Den *Frühesten Anfangszeitpunkt (FAZ)*, der den frühesten Moment an dem die Aktivität starten kann beschreibt. Den *Spätesten Anfangszeitpunkt (SAZ)*, der den spätesten Moment an dem eine Aktivität starten muss beschreibt. Den *Frühesten Endzeitpunkt (FEZ)*, der den frühesten Moment an dem eine Aktivität beendet sein kann beschreibt und den *Spätesten Endzeitpunkt (SEZ)*, der den spätesten Moment an dem die Aktivität beendet ist beschreibt. Diese vier Zeitpunkte sind in der *Zeitbasierten-Sicht* von großer Bedeutung.

2.2.2 Aktivitäten mit festen Terminen

Neben Aktivitäten die einen zeitlichen Abstand zu ihren Vorgängern haben, existieren Aktivitäten mit einem festen Start- oder Endzeitpunkt. Jedoch sind während der Modellierung die zeitlichen Abstände zum Prozessstart nicht bekannt. Diese können erst zur Laufzeit bestimmt werden. In Abbildung 2.6 ist die Markierung einer Aktivität mit festem Termin und festlegen ihres Zeitpunktes zu sehen.

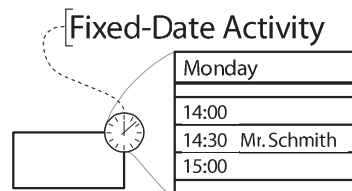


Abbildung 2.6: Aktivität mit festem Termin [LWR10]

2.2.3 Wiederholende Aktivitäten - Schleifen

Bei einer Schleife wird am Ende Entschieden, ob die Schleife wiederholt oder ob sie verlassen wird. Findet eine weitere Iteration der Schleife statt, können zeitliche Abstände zwischen Aktivitäten in den verschiedenen Iterationen entstehen. Die Modellierung eines

zeitlichen Abstandes zwischen zwei Iterationen einer Aktivität ist in Abbildung 2.7 zu sehen. Wie bei einem zeitlichen Abstand üblich, existiert ein minimaler und maximaler Abstand, sowie die Art der Abhängigkeit (*Start-Start*, *Start-End*, *End-Start*, *End-End*)

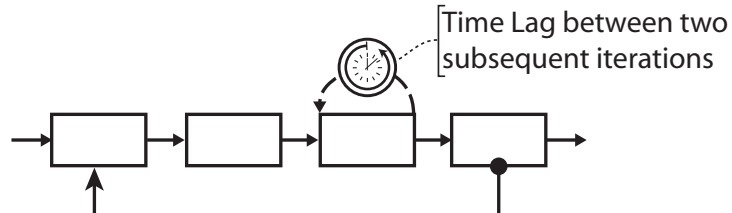


Abbildung 2.7: Weiderholungen bzw. Cycles [LWR10]

Neben den einzelnen Zeitaspekten die in Prozessen dargestellt werden können, werden im nächsten Abschnitt nun die sich ergebenden Anforderungen an die Zeit-Darstellung definiert.

2.3 Anforderungen an die Zeit-Darstellung

Ziel dieser Arbeit, ist eine Darstellung von blockstrukturierter Prozesse auf Gantt-Diagramme zu entwickeln. Diese Darstellung sollten so viele Informationen wie möglich darstellen können, jedoch trotzdem Übersichtlich bleiben. Folgende Zeitaspekte sind für die Analyse der zeitlichen Eigenschaften eines Prozesses besonders wichtig und sollten in dieser Darstellung dargestellt werden:

Dauer Die Darstellung muss es ermöglichen auf einfache Art und Weise die minimale und maximale Dauer einer Aktivität zu bestimmen.

Zeitlicher Abstand In der Darstellung muss der Abstand zwischen Aktivitäten eindeutig bestimmt werden können. Neben dem minimalen Abstand, muss auch der maximale Abstand dargestellt sein.

Neben der minimalen/maximalen Dauer einer Aktivität, dem minimalen/maximalen Abstand zwischen Aktivitäten, sollen in der Darstellung weitere wichtige Zeiteigenschaften

wie *Frühesten Anfangszeitpunkt*, *Spätesten Anfangszeitpunkt*, *Frühesten Endzeitpunkt* und *Spätesten Endzeitpunkt* übersichtlich dargestellt sein.

Nach den Grundlagen des ADEPT-Metamodell, den einzelnen Zeitaspekten und den Anforderungen werden im nächsten Kapitel Repräsentationen von Zeiten betrachtet. Diese stellen zum Teil Algorithmen zur Berechnung der Anfangszeitpunkte bzw. Endzeitpunkte der Aktivitäten zur Verfügung.

3

Repräsentation von Zeit

In diesem Kapitel werden verschiedene Arten der Repräsentationen von Zeiten besprochen. Allgemein gibt es verschiedene Repräsentationsformen: für den Benutzer gut lesbare Darstellungen und graphenbasierte Darstellungen für Programme und Algorithmen. Jede Darstellungsart hat dabei ihre Vor- und Nachteile.

3.1 Gantt-Diagramme

Das klassische Gantt-Diagramme gehört zu den für den Benutzer einfach ablesbaren Darstellungsarten. Sie stellen einen Zeitablauf in Balkenform dar. In Abbildung 3.1 ist ein klassisches Gantt-Diagramm gezeigt. Links in der Abbildung sieht man die typische Tabelle eines Gantt-Diagramm, in welcher Anfang- und Endzeitpunkte und der Name der einzelnen

Aktivität bzw. in Gantt-Diagramme auch Vorgang genannt eingetragen sind. Rechts in der Abbildung, sieht man das Gantt-Diagramm. Dieses Diagramm ist mit einer Zeitleiste versehen, welches das Jahr, den Monat und einzelne Tage anzeigt. Anhand dieser Zeitleiste sind Rechtecke angeordnet, die die einzelne Vorgänge repräsentieren. Die Breite eines Rechtecks stellt dabei die Dauer des Vorgangs dar. Der Ablauf der einzelnen Vorgänge wird durch Pfeile definiert. Die Pfeile verbinden die einzelnen Vorgänge und repräsentieren dazu noch den zeitlichen Abstand zwischen Vorgänge. In klassischen Gantt-Diagrammen können eine feste Dauer eines Vorgangs, ein fester Abstand zwischen Vorgängen und ein fester Startzeitpunkt dargestellt werden.

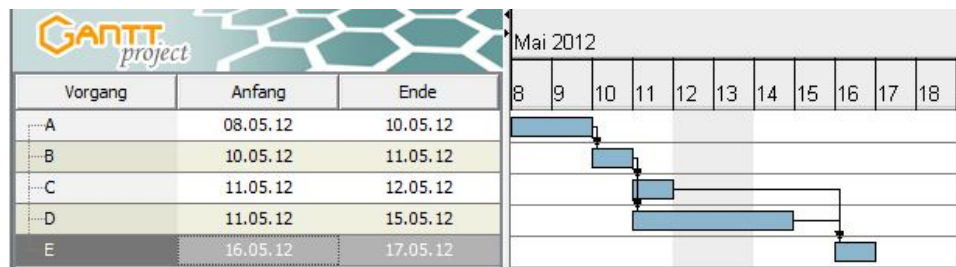


Abbildung 3.1: Gantt Diagramm aus GanttProject

3.2 AsbruView

AsbruView ist eine Erweiterung der Gantt-Diagrammen. Das einzelne Rechteck, welches einen Vorgang im Gantt-Diagramm darstellt, wurde durch ein zweites Rechteck, welches über Vorgang steht ergänzt. Dieses Rechteck stellt die minimale Dauer dar. Das bestehende untere Rechteck wird als maximale Dauer definiert. Diese beide Rechtecke bzw. minimale/-maximale Dauer werden durch zwei Rauten verbunden, die den *Spätesten Startzeitpunkt* und *Frühesten Endzeitpunkt* repräsentieren. Die beiden fehlenden Zeitpunkte, *Frühesten Startzeitpunkt* und *Spätesten Endzeitpunkt* werden durch zwei senkrechte Linien dargestellt. Diese Darstellungsart wird häufig in medizinischen Prozessen angewandt. Sie hat jedoch den Nachteil, dass bei vielen Vorgängen das AsbruView-Diagramm durch die Schichtung der Rechtecke und der Raute in der Vertikalen sehr groß wird und die Übersichtlichkeit ab-

nimmt. Zudem addieren sich zum Teil die *Spätesten Endzeitpunkte* der einzelnen Vorgänge immer weiter auf und entfernen sich immer weiter von den Rechtecken die den Vorgang repräsentieren.

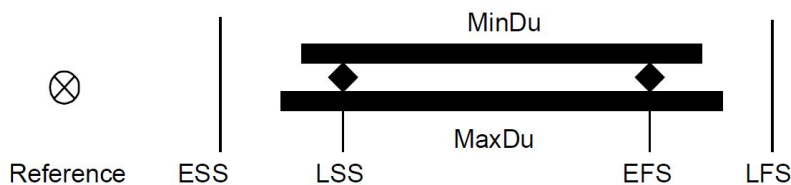


Abbildung 3.2: AsbruView Aktivität [MKSJ98]

3.3 Netzplantechnik

Netzplantechnik bezeichnet den Ablauf, Analyse, Beschreibung, Überwachung und Planung von Abläufen auf der Basis der Graphentheorie. Die Netzplantechnik wird in zwei Arten von Methoden unterteilt (siehe Abbildung 3.3), zum einen die deterministischen und zum anderen die stochastischen Methoden. Zu den deterministischen Methoden gehören die *Critical Path Method (CPM)* und die *Metra Potential Method (MPM)*. Zu den stochastischen Methoden gehören *Program Evaluation an Review Technique (PERT)* und *Graphical Evaluation and Review Technique (GERT)*. Schließlich ist in der Netzplantechnik der kritische Pfad von großer Bedeutung. Er stellt den Pfad mit der längsten Dauer vom Start zum Ende des Netzplanes dar.

Das Ziel der Netzplantechnik ist es die frühesten und spätesten Anfangs- und Endzeitpunkte der einzelnen Vorgänge zu berechnen. Im folgenden werden einige Methoden der Netzplantechnik genauer betrachtet. Die Methode GERT wird dabei nicht näher betrachtet, da sie durch ihre stochastische Ausrichtung für die vorliegende Arbeit nicht von Bedeutung ist. Der interessierte Leser sei hier beispielsweise auf [Rin98] oder [NM02] verwiesen.

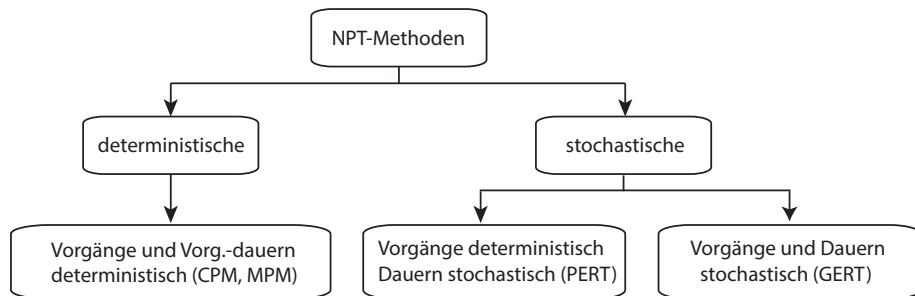


Abbildung 3.3: Methoden der Netzplantechnik [DD07]

3.3.1 CPM (Critical Path Method)

Die *Critical Path Method* ist eine graphenbasierte Zeitrepräsentation, welche als zyklenfreies *Vorgangs-Pfeil-Netz* dargestellt wird (siehe Abbildung 3.4). In CPM werden die einzelnen Aktivitäten (Vorgänge) als Pfeile zwischen einzelnen Knoten (Ereignisse) dargestellt. Diese Pfeile werden mit einem Abstand bewertet, welcher als *End-Start* definiert ist. Zudem sind die Pfeile für die Verkettung verantwortlich. Um nicht sequenziell ablaufende Vorgänge zu synchronisieren (siehe Abbildung 3.4) müssen in CPM Scheinvorgänge eingeführt werden. Diese Scheinvorgänge sind fiktive Vorgänge und werden mit der Dauer Null versehen. In CPM ist es nicht möglich Entscheidungen und Schleifen zu darzustellen. Zudem können nur minimale Dauern und minimale Abstände zwischen Aktivitäten dargestellt werden. Feste Termine sind nicht unterstützt. Die Berechnung der vorhandenen Anfangs- und Endzeitpunkte erfolgt durch einen Shortest-Path-Partitioning-Algorithmus [EM92] in der Laufzeitkomplexität $O(n^2)$.

3.3.2 MPM (Metra Potential Method)

Die *Metra Potential Method* ist analog zu CPM eine graphenbasierte Form einer Zeitrepräsentation. Jedoch beruht MPM auf einem *Vorgangs-Knoten-Netz*. Dabei werden Aktivitäten als Knoten dargestellt. Der zeitliche Abstand zwischen den Knoten wird durch eine Kante repräsentiert. Dies ist in der Abbildung 3.5 zu sehen.

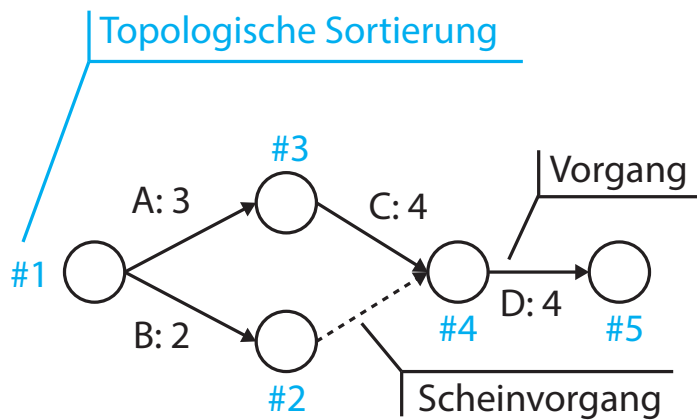


Abbildung 3.4: CPM Graph

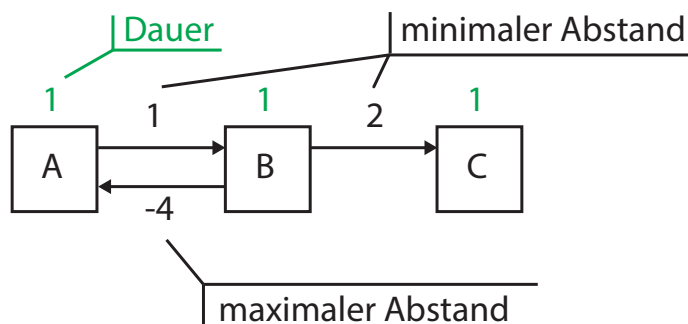


Abbildung 3.5: MPM Graph

Der zeitliche Abstand zwischen den einzelnen Knoten (Aktivitäten) ist in einer *Start-Start* Beziehung modelliert. Der Abstand ist vom Beginn des Startknoten der Kante zum Beginn des Zielknoten der Kante definiert. Diese Kante stellt den minimalen Abstand dar. Neben diesem minimalen Abstand, ist es in MPM möglich einen maximalen Abstand zwischen Knoten zu definieren. Dieser maximale Abstand, wird durch eine rückwärtsgerichtete Kante dargestellt. Der maximale Abstand wird zusätzlich Negativ bewertet (siehe Abbildung 3.5). Folglich wird

der minimale Abstand mit einer positiven Zeiteinheit bewertet und der maximale Abstand mit einem negativen Zeiteinheit. Durch diesen zusätzlichen maximalen Abstand kann MPM im Vergleich zu CPM mehr Zeitpunkte berechnen. Diese Zeitpunkte bestehen aus frühester, spätester Anfangszeitpunkt und frühester und spätester Endzeitpunkt. Die Berechnung in MPM erfolgt durch das *Label-Correcting-Verfahren* [NM02]. Die Laufzeitkomplexität ist $O(nm)$, mit n Anzahl der Knoten und m Anzahl der Kanten. Schleifen und Entscheidungen, lassen sich in MPM analog zu CPM nicht darstellen.

3.3.3 PERT (Program Evaluation and Review Technique)

PERT sind ereignisorientierte Netzpläne, welche auf einem *Ereignis-Knoten-Netz* basieren. In PERT stehen dabei Ereignisse im Mittelpunkt, welche als Knoten modelliert sind. Dabei ist der große Unterschied zu CPM und MPM, dass die Zeiten bei PERT in Form einer Wahrscheinlichkeitsverteilung angegeben werden. Dabei werden drei Werte angegeben. Der beste Fall a , der häufigste Fall m und der schlechteste Fall b . Diese drei Fälle werden durch die Berechnungsvorschrift, die in Abbildung 3.6 zu sehen ist, zu einem Mittelwert zusammengefasst. Die Varianz der drei Fälle wird für die Dauer zusätzlich berechnet. Durch diese Art der Angabe der Dauern ist in PERT keine genaue Aussage über früheste und späteste Zeitpunkte möglich. Die Modellierung von Schleifen und Entscheidungen ist in PERT nicht möglich. Die Berechnung der einzelnen geschätzten Zeitpunkte in PERT ist analog wie in CPM über den Shortest-Path-Partitioning-Algorithmus [EM92] in der Laufzeitkomplexität $O(n^2)$ möglich. Als zeitlicher Abstand bzw. Dauer, welche im Shortest-Path-Partitioning-Algorithmus benutzt wird, wird hierbei der berechnete Mittelwert gesetzt.

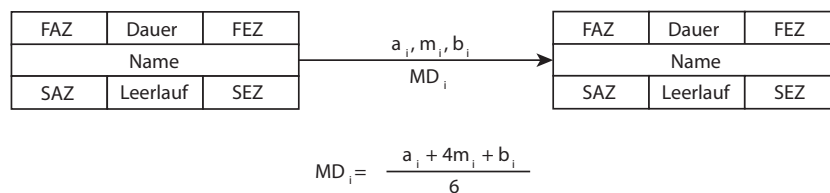


Abbildung 3.6: Allgemeiner PERT-Graph

Nach den einzelnen Betrachtungen der Repräsentationsformen der Zeit, werden im nächsten Abschnitt verbreitete Projekt Management Werkzeuge und ihre interne Darstellung der Zeit vorgestellt.

3.4 Projekt Management Werkzeuge

In diesen Abschnitt werden Projekt Management Werkzeuge vorgestellt, die meist Gantt-Diagrammen oder leicht abgeänderte Gantt-Diagrammen als Zeit-Repräsentation verwenden. Neben Visio von Microsoft, werden Programme vorgestellt, welche in Planung von Projekten verschiedener Größe häufig eingesetzt werden.

3.4.1 Visio

Das Programm Visio wurde von Microsoft entwickelt. Es wird dazu verwendet Gantt-Diagramme zu modellieren und zu erstellen. Das Gantt-Diagramm wird dabei in einer Tabelle definiert (siehe Abbildung 3.7). In dieser Tabelle werden Anfang, Abschluss, die Dauer (resultiert aus Anfang und Abschluss) und die einzelnen Vorgänger eines Vorgangs bzw. die zugehörige Phase (mehrere Vorgänge zusammen) definiert. Das Gantt-Diagramm wird durch eine Zeitleiste unterstützt, darauf werden einzelne Vorgänge, die als Rechteck dargestellt sind platziert. Für jeden einzelnen Vorgang bzw. Rechteck wird eine neue Zeile erstellt, d.h. in jeder Zeile steht ein Vorgang. Die Breite des einzelnen Rechtecks stellt die Dauer des Vorgangs dar und wird anhand der Zeitlinie ausgerichtet. Zeitliche Abstände und die Abläufe, werden durch Pfeile realisiert. Der zeitliche Abstand wird durch die Länge des Pfeils dargestellt. Mehrere Vorgänge können in Visio zu einer Phase gebündelt werden. Sie stellt die Gesamtdauer der Vorgänge und deren zeitlichen Abstände dar. Die Dauer ist wieder die Breite des Rechtecks, jedoch sind an der linken und rechten Kante des Rechtecks Dreiecke angebracht. Ein festes Startdatum eines Vorgangs ist in Visio durch eine schwarze Raute modelliert. In Visio sind Sequenzen und Parallelitäten bei der Modellierung möglich. Zudem ist nur eine feste Dauer und einen festen Abstand im Gantt-Diagramm darstellbar. Schleifen, Entscheidungen oder eine maximale Dauer bzw. maximalen Abstand ist in Visio nicht darstellbar.



Abbildung 3.7: Gantt Diagramm in Microsoft Visio

3.4.2 A-Plan

In A-Plan ist die Zeitanzeige ebenfalls als Gantt-Diagramm realisiert. Die Darstellung von Dauer und den zeitlichen Abstand ist wie in Visio gehandhabt. Das Rechteck repräsentiert die Dauer, die Pfeile den einzelnen Abstand (siehe Abbildung 3.8). Die Eingabe für das Gantt-Diagramm läuft ebenfalls über eine Tabelle, bei der Beginn des Vorgangs und eine feste Dauer eingegeben werden müssen. Zudem werden Vorgänge wieder in einer übergeordneten Phase zusammengefasst. Es beschreibt ein Projekt und unterstützt den Ablauf des Projektes. Dieses System wird meist bei mittleren Software Projekten eingesetzt. Die Darstellung von Entscheidungen und maximaler Dauer bzw. maximalen Abstand ist in A-Plan nicht möglich. Schleifen können in A-Plan über wiederholende Vorgänge realisiert werden.

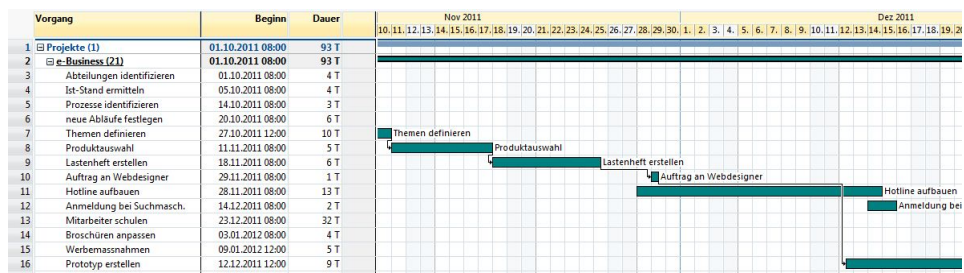


Abbildung 3.8: Gantt Diagramm in A-Plan

3.4.3 Acos Plus

Acos plus ist wie A-Plan ein Programm zur Planung von Projekten. Die zeitliche Repräsentation in A-Plan wird ebenfalls in Gantt-Diagrammen realisiert. Jedoch wurden diese Gantt-Diagramme durch mehrere Dinge und Symbole erweitert. In Acos Plus ist es möglich einen maximalen Abstand und eine maximale Dauer zu definieren. Das Gantt-Diagramm ist dadurch um Pufferzeiten (hellblaue Linie) und ein Stoppschild (siehe Abbildung 3.9) erweitert. Letztens zeigt den spätesten Anfang eines Vorgangs. Jedoch wird dieser späteste Anfang nicht auf weitere Vorgänge übertragen und bleibt nur für den einzelnen Vorgang darstellbar. Das Gantt-Diagramm in Acos Plus zeigt immer den besten Fall und Zusatzinformationen an. Zusätzlich werden Pufferzeiten und der späteste Anfang für einen einzelnen Vorgang dargestellt. Schleifen werden wie A-Plan durch wiederholende Vorgänge modelliert. Die Möglichkeit zur Modellierung von Entscheidungen und die Darstellung des SEZ und SAZ über alle Vorgänge ist nicht gegeben.

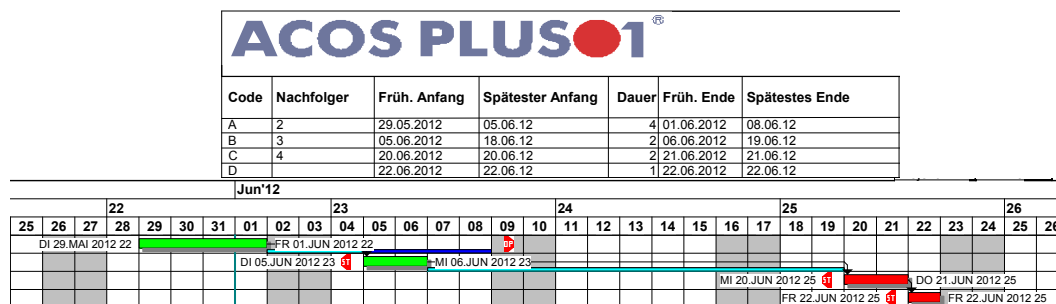


Abbildung 3.9: Gantt Diagramm in Acos Plus

3.5 Fazit

Nach der Betrachtung der einzelnen Repräsentationen der Zeit kann nun jede Repräsentation nach ihren Eigenschaften die sie erfüllt bewertet werden. Einen festen Abstand und eine feste Dauer hat indirekt jede Darstellungsart, jedoch sind Eigenschaften wie XOR (Entscheidung), Schleife und maximaler Abstand bzw. Dauer nicht von jeder Darstellungsart

unterstützt. Zudem ist eine einfache Verständlichkeit für einen Benutzer bei manchen Darstellungsarten nur zum Teil, bei manchen gar nicht gegeben. Die einzelnen Bewertungen sind in Tabelle 3.1 zu sehen. Ein + steht für eine positive Bewertung, ein +- für eine neutrale Bewertung und ein - für eine negative Bewertung.

Art	XOR	Schleife	Max Abstand/Dauer	Verständlichkeit
Gantt	-	-	-	+
Asbru	-	-	+	+-
CPM	-	-	-	-
MPM	-	-	+	-
PERT	-	-	-	-

Tabelle 3.1: Bewertung der Zeit Repräsentationen

Durch diese Bewertung, erkennt man, dass Gantt-Diagramme für den Benutzer eine hohe Verständlichkeit aufweisen. Sie bilden den Abstand und die Dauer einfach ab. Jedoch unterstützen die klassischen Gantt-Diagramme nicht alle Eigenschaften, die bei einem Prozess notwendig sind. In der Tabelle 3.2 sieht man, dass in Acos plus durch die Erweiterung der Gantt-Diagrammen mehr Eigenschaften unterstützt. Jedoch nimmt die Verständlichkeit und die Übersichtlichkeit ab. Um die Abbildungen für Prozesse auf Gantt-Diagramme zu erreichen, muss man diese wie bei Acos plus geschehen erweitern. Jedoch werden selbst in Acos Plus und den anderen Repräsentationen nicht alle Eigenschaften unterstützt, die in Prozesse existieren. Zudem ist eine Ansicht für den schlechtesten Fall im Prozess bzw. Diagramm notwendig, welche aus der Unterstützung von maximaler Dauer und maximaler Abstand resultiert.

Art	XOR	Schleife	Max Abstand/Dauer	Verständlichkeit
Visio	-	-	-	+
A-Plan	-	+	-	+
Acos plus	-	+	+-	+-

Tabelle 3.2: Bewertung der einzelnen Programme

Im nächsten Kapitel, werden die Abbildungen von Prozessen auf die Gantt-Diagramme gezeigt. Zudem werden die Gantt-Diagramme im nächsten Kapitel um die fehlenden Eigenschaften erweitert.

4

Abbildung blockstrukturierter Prozessgraphen auf Gantt-Diagramme

Im folgenden Kapitel wird die Abbildung blockstrukturierter Prozessgraphen auf Gantt-Diagramme am Beispiel des ADEPT-Metamodells betrachtet. Hierzu werden die klassischen Gantt-Diagramme um zusätzliche Konstrukte erweitert, um eine vollständige Abbildung der Prozessgraphen zu ermöglichen.

Neben minimaler und maximaler Dauer von Aktivitäten, minimalen und maximalen Abständen zwischen Aktivitäten, Sequenzen und Parallelität werden hierbei auch Entscheidungen (XOR) und Schleifen betrachtet. Zudem werden zwei unterschiedliche Betrachtungsweisen vorgestellt, die es erlauben die *Frühesten Anfangszeitpunkte* (FAZ) bzw. *Frühesten Endzeitpunkte* (FEZ) oder *Spätesten Anfangszeitpunkte* (SAZ) bzw. *Spätesten Endzeitpunkte* (SEZ) von Aktivitäten zu bestimmen.

4.1 Abbildung von Prozessgraphen auf klassische Gantt-Diagramme

In den klassischen Gantt-Diagramme besitzen Aktivitäten (Vorgang) eine feste Dauer. Darüber hinaus ist es möglich einen festen Abstand zwischen Aktivitäten (Vorgängen) anzugeben. Ein Vorgang wird hierbei als Rechteck dargestellt (siehe Abbildung 4.1 Markierung 1). Die Breite des Rechtecks zeigt die Dauer des Vorgangs an. Der Abstand zwischen Vorgängen wird durch einen Pfeil, der die Vorgänge verbindet (siehe Abbildung 4.1 Markierung 2) dargestellt. Dabei bestimmt die Länge des Pfeils auf der waagrechten Ebene den zeitlichen Abstand zwischen den beiden Vorgängen. Für jeden einzelnen Vorgang wird in einem Gantt-Diagramm eine neue Zeile erstellt, damit diese übersichtlich mit Pfeilen verbunden werden können. Neben Sequenzen wird auch Parallelität unterstützt. Dabei gehen aus einem Vorgang mehrere Pfeile aus. Bei der Zusammenführung der Pfade die aus der Parallelität resultieren, werden mehrere Pfeile bzw. Abstände auf einem Vorgang zusammengeführt (siehe Abbildung 4.1 Markierung 3). Damit sind die Möglichkeiten klassischer Gantt-Diagramme erschöpft.

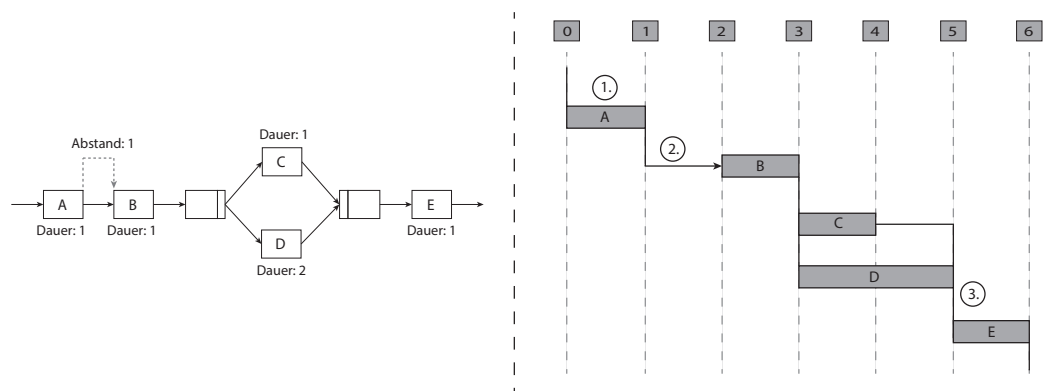


Abbildung 4.1: Prozess Abbildung auf klassische Gantt-Diagramme

In klassischen Gantt-Diagrammen sind somit nicht alle zuvor vorgestellten zeitlichen Attribute von Prozessen darstellbar. Es existiert nur eine feste Dauer für eine Vorgang sowie ein fester

Abstand zwischen Vorgängen. Zeitliche Attribute wie minimale und maximale Dauer eines Vorgangs, minimale und maximale Abstände zwischen Vorgängen, Schleifen und sowie XOR-Entscheidungen werden nicht unterstützt. Um Prozesse sinnvoll auf Gantt-Diagramme abbilden zu können müssen klassische Gantt-Diagramme daher um neue Konzepte erweitert werden. Die nötigen Erweiterungen der klassischen Gantt-Diagramme werden im diesem Kapitel vorgestellt. Diese Erweiterung wird im folgenden als *extended-Gantt* (kurz *eGantt*) bezeichnet .

4.2 Vorüberlegungen

Ziel der Erweiterung *eGantt* ist es möglichst viele Zeiteigenschaften eines Prozesses in einem Diagramm darstellen zu können. Jedoch sollte hierbei die Übersichtlichkeit und die einfache Ablesbarkeit der Zeiteigenschaften weiterhin im Vordergrund stehen. Für die einfache Bestimmung und Ablesbarkeit der Zeiteigenschaften eines Prozesses sind folgende Eigenschaften besonders wichtig: *Frühester Anfangszeitpunkt*, *Frühester Endzeitpunkt*, *Spätester Anfangszeitpunkt*, *Spätester Endzeitpunkt*, minimale und maximale Dauer eines Vorgangs, sowie die minimalen und maximalen Abstände zwischen den einzelnen Vorgängen. Dazu kommen noch XOR-Entscheidungen und Schleifen.

Da die Übersichtlichkeit schnell abnimmt wenn FAZ, FEZ, SAZ und SEZ gleichzeitig in einem Diagramm dargestellt werden (siehe Kapitel 3.2), werden die einzelnen Zeitpunkte bei *eGantt* in zwei Darstellungsarten (siehe Abbildung 4.2) aufgesplittet. Der FAZ und der FEZ ist in der Min-Darstellung zusammengefasst. Die Min-Darstellung zeigt damit immer den besten Fall, es werden also hauptsächlich immer die minimalen Zeiten und Abstände betrachtet. Die Max-Darstellung betrachten den SAZ und den SEZ, welche immer den schlechtesten Fall darstellen. In *eGantt* liegt bei der Darstellung der Modellierungsphase von Prozessen das Augenmerk auf der Darstellung des allgemeinen Fall und nicht auf einer einzelnen Prozessinstanz. Der bei der Ausführung des Prozesses zeitlich eintretende Fall, wird in dieser Darstellung daher nicht dargestellt.

Im weiteren Teil werden die verschiedene Konzepte von Prozessen und deren Abbildungen auf *eGantt* betrachtet. Anfangs werden die einzelnen Knotenattribute und dann die Abbildung größerer Prozessfragmente vorgestellt.

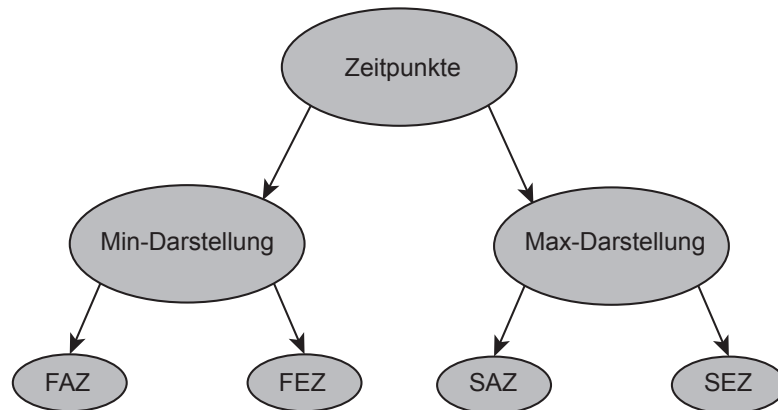


Abbildung 4.2: Min- und Max-Darstellung und ihre Beziehung zu FAZ/FEZ bzw. SAZ/SEZ

4.3 Knotenattribute

Wie im Kapitel 2 besprochen, besitzt jede Aktivität eine minimale und maximale Dauer. In klassischen Gantt-Diagrammen ist dies so nicht darstellbar da ein Vorgang nur eine feste Dauer besitzt. In eGantt wird die Darstellung von Vorgängen daher erweitert um sowohl die minimale als auch die maximale Dauer von Aktivitäten darstellen zu können. Die Dauer wird analog zu den klassischen Gantt-Diagrammen als Rechteck dargestellt. Die minimale Dauer wird dabei als Breite eines Rechtecks dargestellt. Die maximale Dauer wird, wie in Abbildung 4.3 dargestellt, zusätzlich in Form eines gestrichelten Rechtecks an das Rechteck der minimalen Dauer angefügt. Dieser Vorgang mit minimaler und maximaler Dauer wird als *eGantt-Vorgang* bezeichnet. In Abbildung 4.3 ist auf der linken Seite ein Prozess mit einer Aktivität A zu sehen. Diese hat eine minimale Dauer von 3 und eine maximale Dauer von 5 Zeiteinheiten. Auf der rechten Seite der Abbildung ist die Darstellung dieses Prozesses als eGantt-Diagramm zu sehen, welches den Vorgang A enthält. Wie in klassischen Gantt-Diagrammen üblich, wird auch in eGantt eine Zeitleiste eingefügt (siehe Abbildung 4.3), um die Dauer der einzelnen Vorgänge einfach bestimmen zu können.

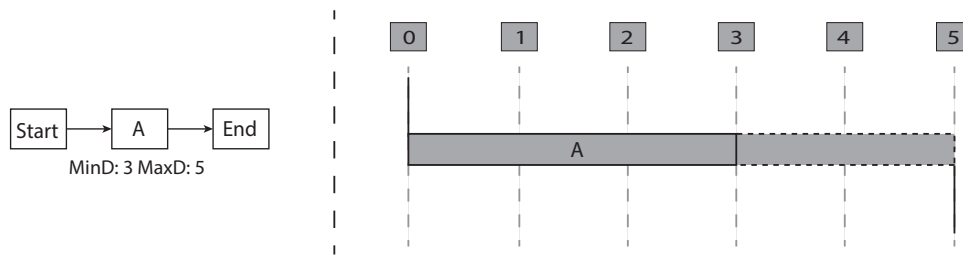


Abbildung 4.3: Vorgang mit minimaler und maximaler Dauer

Neben „einfachen“ Aktivitäten, existieren noch Aktivitäten mit festem Start- bzw. festen Endzeitpunkt. Bei diesen Aktivitäten kann zum Modellierungszeitpunkt der Beginn der Aktivität nicht eindeutig bestimmt werden, da der Beginn der Aktivität erst zum Ausführungszeitpunkt des Prozesses festgelegt wird. Um ein Vorgang mit festem Start- bzw. Endzeitpunkt von „einfachen“ Vorgängen unterscheiden zu können wird dieser daher durch eine spezielle Markierung gekennzeichnet (siehe Abbildung 4.4). Für den festen Startzeitpunkt ist diese Markierung an der linken Seite des eGantt-Vorgangs angebracht, für einen festen Endzeitpunkt an der rechten Seite. Zur Laufzeit wird dieser Vorgang dann an die entsprechende Position verschoben, sobald der entsprechende Zeitpunkt bekannt ist (siehe Kapitel 5).

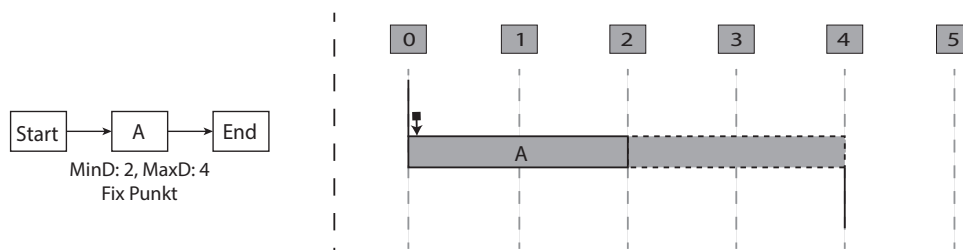


Abbildung 4.4: Vorgang mit festem Startpunkt und minimaler/maximaler Dauer

4.4 Sequenzen und Parallelität

Da Prozesse nicht nur aus einzelnen Aktivität bzw. Vorgängen bestehen, werden in diesem Abschnitt Sequenzen und Parallelitäten betrachtet. Zudem werden die beiden Darstellungsarten Min- und Max-Darstellung für die einzelnen Konstrukte vorgestellt.

4.4.1 Abbildung von Sequenzen

Wie in Abschnitt 4.2 erläutert, gibt es zwei verschiedene Darstellungsarten. Bei der Min-Darstellung, die den FAZ und FEZ betrachtet, wird für die Ausrichtung des eGantt-Vorgang immer der beste Fall angenommen. Somit ist der Beginn eines einzelnen eGantt-Vorgang an seinem FAZ ausgerichtet. Weiter erfolgt bei der Min-Darstellung, wie in Abbildung 4.5 zu sehen ist, die Verbindung zum Vorgängervorgang konsequenterweise über deren FEZ. Es wird somit eine Linie vom FEZ des Vorgängervorgang zum FAZ des Nachfolgvorgangs eingefügt.

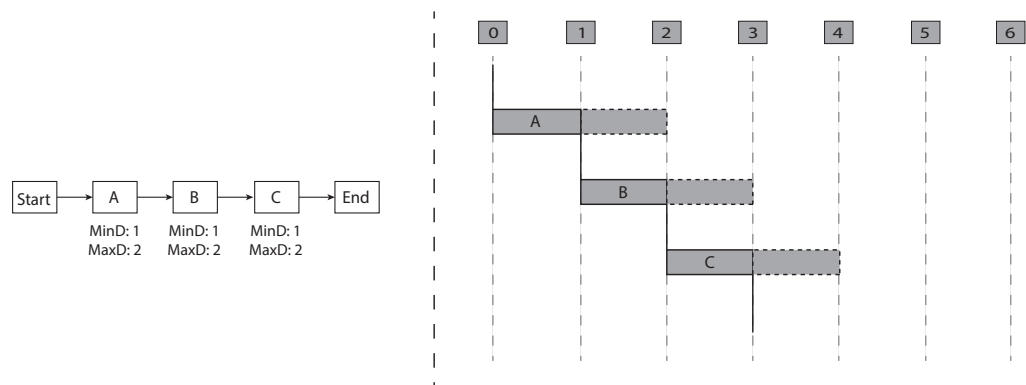


Abbildung 4.5: Sequentielle Ausführung von Vorgängen in Min-Darstellung

Neben der Min-Darstellung existiert wie in Abschnitt 4.2 besprochen noch die Max-Darstellung. Allgemein betrachtet die Max-Darstellung die schlechtesten Fälle der Zeiten. Dementsprechend stellt sie den SAZ und SEZ des Aktivitäten dar. Daher wird bei der Max-Darstellung der eGantt-Vorgang an seinem SAZ ausgerichtet und die Verbindung zum Folgevorgang

findet über den SEZ der Vorgängeraktivität statt. Es wird somit eine Linie vom SEZ des Vorgängervorgang zum SAZ des Nachfolgevorgang gezogen (siehe Abbildung 4.6).

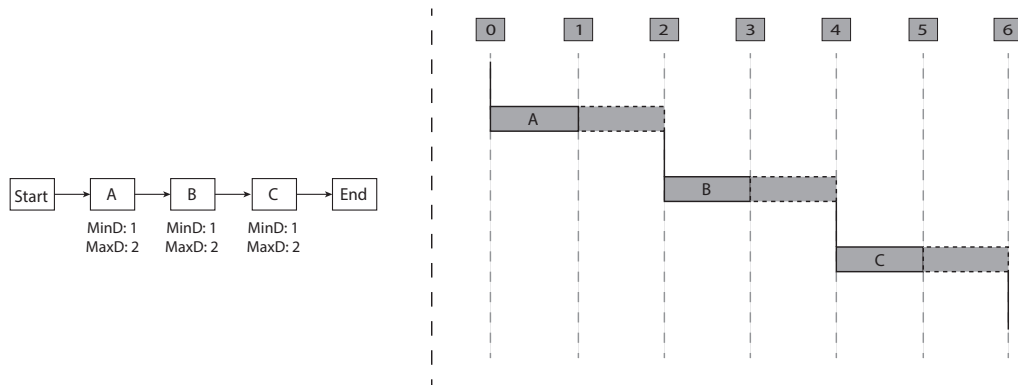


Abbildung 4.6: Sequentielle Ausführung von Vorgängen in Max-Darstellung

4.4.2 Abbildungen von Zeitkanten

Neben maximaler und minimaler Dauer von Prozess-Aktivitäten, existieren auch Zeitkanten zwischen Prozess-Aktivitäten. Diese Zeitkanten werden in einem eGantt-Diagramm wie im klassischen Gantt-Diagramm als Abstände zwischen den jeweiligen Vorgängen dargestellt. Die Zeitkanten besitzen wie in Kapitel 2.2.3 besprochen, einen minimalen und maximalen Abstand und zusätzlich eine Abhängigkeitsbeziehung zwischen den einzelnen Aktivitäten (*Start-Start*, *Start-End*, *End-Start*, *End-End*).

Für die Min-Darstellung wird, wie in Abschnitt 4.2 beschrieben, der beste Fall für die Zeiten angenommen. Es wird somit der minimale Abstand herangezogen. Hierzu wird ein durchgezogener waagrechter Pfeil an die senkrechte Linie des Vorgängervorgangs angehängt und der eGantt-Vorgang wird im Diagramm um den minimalen Abstand der beiden Vorgängen nach rechts verschoben (siehe Abbildung 4.7). Also ist der FAZ des eGantt-Vorgangs an dem FEZ plus dem minimalen Abstand der beiden Vorgängen ausgerichtet (siehe Abbildung 4.7).

Um die zeitlichen Eigenschaften des Prozesses beurteilen zu können ist jedoch auch der

maximale Abstand von Bedeutung. Dieser wird daher hinter dem eGantt-Vorgang mit einem gestrichelten Rechteck und integriertem gestrichelten Pfeil angehängt. Damit ist leicht zu Erkennen, um wie viel sich der Vorgang möglicherweise nach hinten verschieben kann.

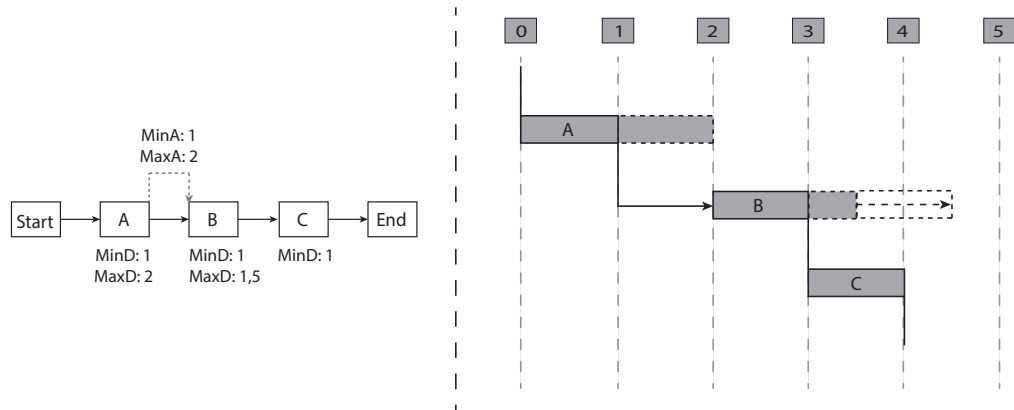


Abbildung 4.7: Sequenz mit zeitlichem Abstand in Min-Darstellung

Bei der Max-Darstellung, die den SAZ und SEZ abbildet, wird analog an die senkrechte Linie der minimale zeitliche Abstand der beiden Vorgängen als Pfeil angehängt. Jedoch wird der maximale Abstand im Vergleich zur Min-Darstellung vor den eGantt-Vorgang gezogen, um den Vorgang entsprechend am SAZ auszurichten (siehe Abbildung 4.8). Die Ausrichtung des eGantt-Vorgangs ist also am SEZ der Vorgängervorgangs plus dem maximalen Abstand der beiden Vorgängen.

Aus den vorgestellten Abbildungsvorschriften lassen sich nun die Berechnungsvorschriften für die einzelnen Zeitpunkte eines eGantt-Vorgangs ableiten. Bei der Min-Darstellung (siehe Abbildung 4.9) wird dabei der beste Fall der Zeiten betrachtet. Somit steht bei der Min-Darstellung die Berechnung für FAZ und FEZ im Vordergrund. Darüber hinaus können weitere, für eine Beurteilung der Zeiteigenschaften, wichtige Zeitpunkte berechnet bzw. aus dem eGantt-Diagramm abgelesen werden.

In Abbildung 4.9 sind einzelne Zeitpunkte zu sehen. Diese Zeitpunkte werden durch den FEZ des Vorgängervorgangs (FEZ_X), den minimalen Abstand zwischen den Vorgängen (min_A), minimale Dauer des Folgevorgangs (min_D), maximale Dauer der Folgevorgangs (max_D) und maximalen Abstand zwischen den Vorgängen (max_D) bestimmt. Daraus resultieren der

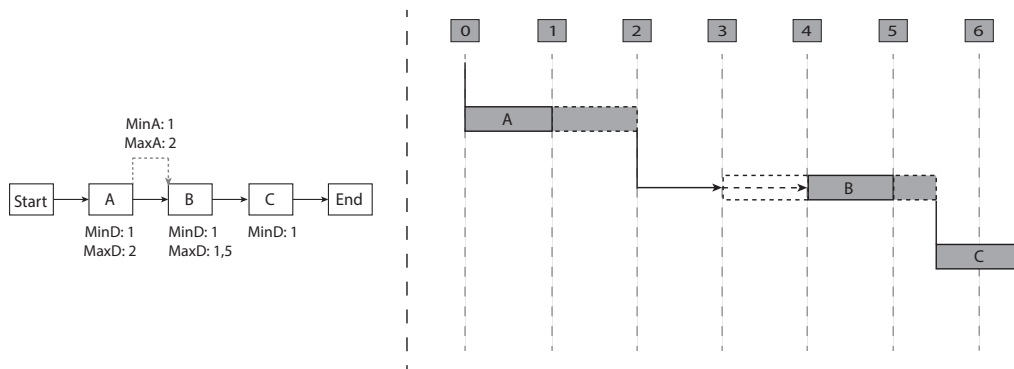


Abbildung 4.8: Sequenz mit zeitlichem Abstand in Max-Darstellung

früheste Anfangszeitpunkt des Folgevorgang (FAZ_Y) und der früheste Endzeitpunkt der Folgevorgang (FEZ_Y).

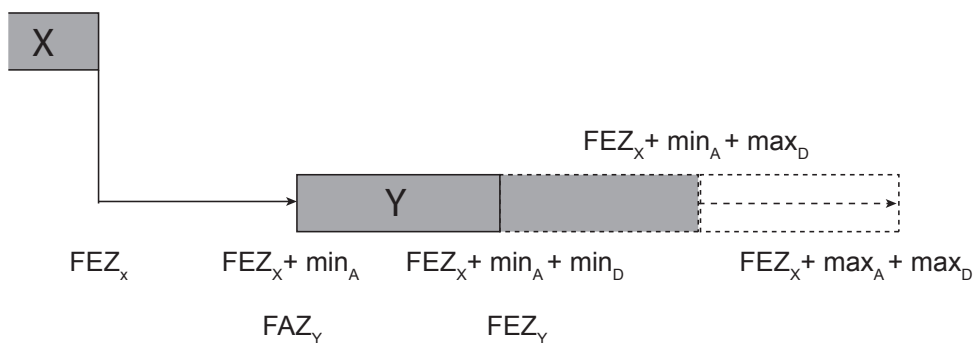


Abbildung 4.9: Min-Darstellung eines eGantt-Vorgang

Analog zur Min-Darstellung ist die Berechnungsvorschrift für die Max-Darstellung in Abbildung 4.10 zu sehen. Bei der Max-Darstellung steht dabei die Berechnung von SAZ und SEZ im Vordergrund. Analog zur Min-Darstellung existieren in der Max-Darstellung neben SAZ und SEZ noch weitere Zeitpunkte, welche für eine Beurteilung der Zeiteigenschaften berechnet bzw. abgelesen werden können.

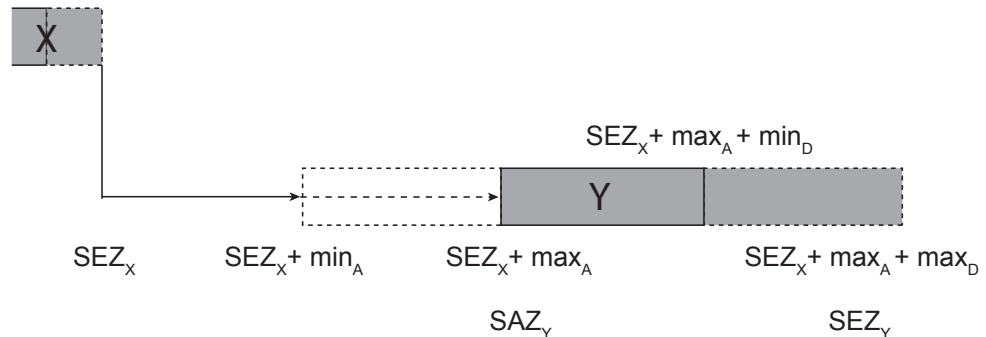


Abbildung 4.10: Max-Darstellung eines eGantt-Vorgang

Wie in Kapitel 2 gezeigt, können Zeitkanten auch über mehrere Prozess-Aktivitäten hinweg modelliert werden (siehe Prozessansicht in Abbildung 4.11). Durch diese Modellierung existieren verschiedene Zeiten die in die Berechnung für die Anfangs- und Endzeitpunkte einfließen. In Abbildung 4.11 führt beispielsweise ein Weg über die Aktivität B nach C und ein zweiter Weg über die Zeitkante zwischen Aktivität A und C. Durch diese verschiedenen Wege und deren zeitlichen Attribute existierenden verschiedenen Ergebnisse der Berechnung. Dadurch muss eine eindeutige Auswahl für den Anfangs- und Endzeitpunkt getroffen werden. Diese Auswahl wird bei der Min-Darstellung über das Maximum der resultierenden Ergebnisse der Anfangs- und Endzeitpunkte getroffen und bei der Max-Darstellung über das Minimum der Zeitpunkte. Der entsprechende Zeitpunkt, wird für den einzelnen Vorgang als Anfangszeitpunkt gesetzt (in Abbildung 4.11 Vorgang C). Bei der Min-Darstellung ist der ausgewählte Anfangszeitpunkt somit der FAZ und der Endzeitpunkt der FEZ. Analog ist bei der Max-Darstellung der Anfangszeitpunkt gleich dem SAZ und der Endzeitpunkt gleich dem SEZ.

In Abbildung 4.11 ist die Min-Darstellung mit einer aktivitäten-übergreifender Zeitkante dargestellt. Das eGantt-Diagramm zeigt dabei wieder den FAZ und FEZ des Vorgangs an. Analog ist in Abbildung 4.12 die Max-Darstellung mit SAZ und SEZ dargestellt. Die einzelnen Arten der Abhängigkeitsbeziehung zwischen den Prozess-Aktivitäten, wird wegen

der Übersichtlichkeit im eGantt-Diagramm nicht angezeigt. Bei der Berechnung werden sie aber beachtet.

Optional kann der Modellierer die aktivitäten-übergreifende Zeitkanten im eGantt-Diagramm auch anzeigen lassen. Sie werden durch eine senkrechte Linie mit angehängten Pfeil ausgegraut dargestellt (siehe Abbildung 4.11 & 4.12). Standardmäßig werden sie jedoch der besseren Übersichtlichkeit wegen nicht angezeigt.

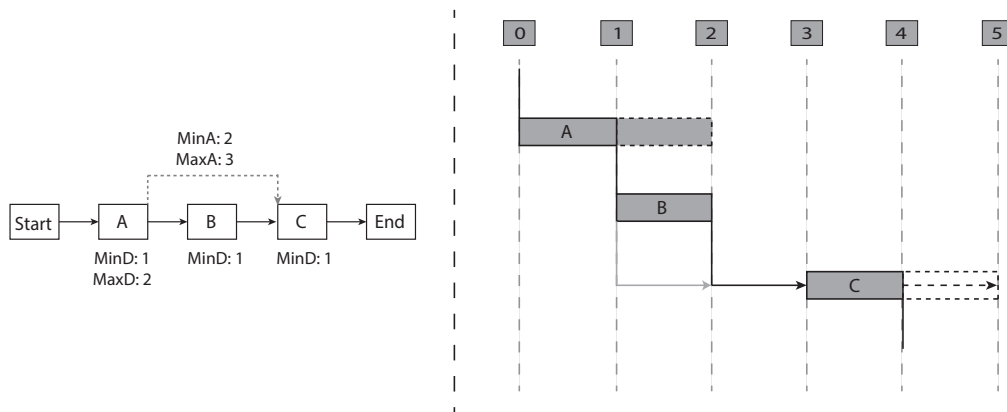


Abbildung 4.11: Sequenz mit zeitlichem Abstand in Min-Darstellung

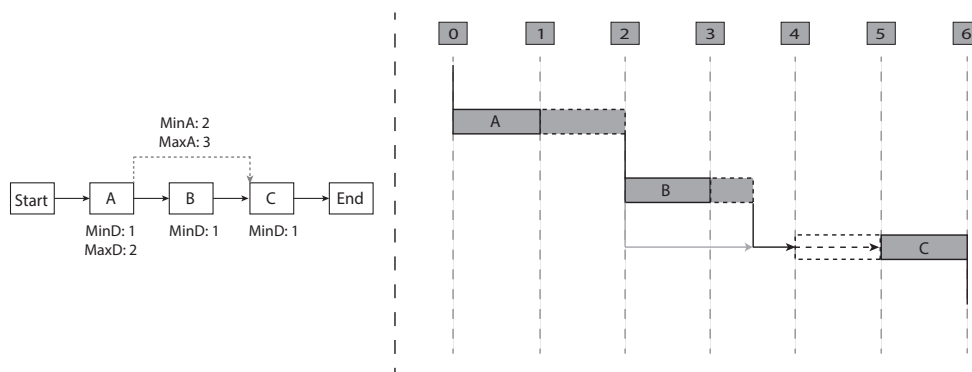


Abbildung 4.12: Sequenz mit zeitlichem Abstand in Max-Darstellung

4.4.3 Parallele und zusammengefasste Zeitkanten

Wie im vorigen Abschnitt erklärt, können Zeitkanten aktivitäten-übergreifend modelliert werden (siehe Abbildung 4.13). Deswegen entstehen wie im vorigen Abschnitt erwähnt verschiedene Wege um Anfangs und Endzeitpunkte zu bestimmen, dabei wird der maximale bzw. minimale Wert je nach Darstellungsart für die Anfangs- und Endzeitpunkte der Zielaktivität (Zielvorgang) gesetzt.

Durch die verschiedenen Wege der Bestimmung der Anfang und Endzeitpunkte beeinflusst nicht jede Zeitkante die Zielaktivität. Dadurch ist nicht jede Zeitkante gleich restriktiv. Es wichtig zu wissen, welche Zeitkanten die Zielaktivität beeinflusst und welche Zeitkanten weniger restriktiv sind. Um dieses Problem zu lösen kann man sich durch eine Selektierung einer Aktivität in der Prozessansicht die einzelnen Abstände im eGantt-Diagramm anzeigen lassen (siehe Abbildung 4.14 & 4.15).

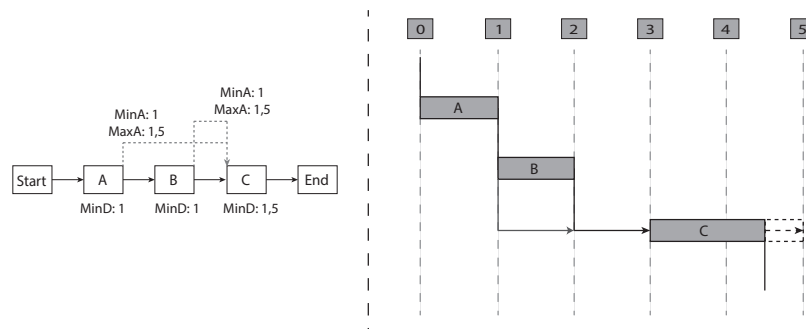


Abbildung 4.13: Zeitliche Abstände mit zusammengefassten Zeitkanten

In Abbildung 4.14 sieht man die Selektierung der Aktivität A und die daraus resultierende Anzeige der Abstände die aus dem Vorgang A im eGantt-Diagramm entspringen. Dabei bedeuten grüne Pfeile (bzw. ein gestricheltes Rechteck mit gestricheltem Pfeil), dass dieser Abstand den Zielvorgang beeinflusst.

In Abbildung 4.15 sieht man eine analoge Selektierung der Aktivität A in der Prozessansicht. Im eGantt-Diagramm werden die Abstände des selektierten Vorgang zu seinen Zielvorgängen farblich hervorgehoben. Jedoch handelt es sich dieses mal um eine weniger restriktive

Zeitkante (Abstand) die von Vorgang A zu Vorgang C führt. Daher werden die fehlerhaften Zeiten (Pfeile) Orange hervorgehoben.

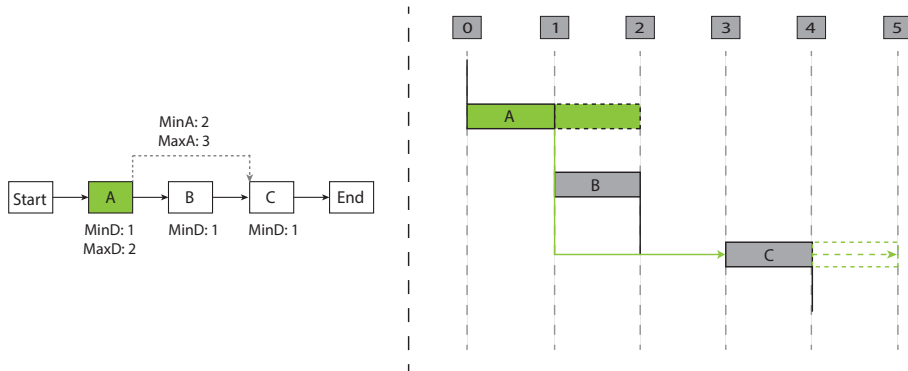


Abbildung 4.14: Darstellung von Zeitkanten mit selektierter Aktivität

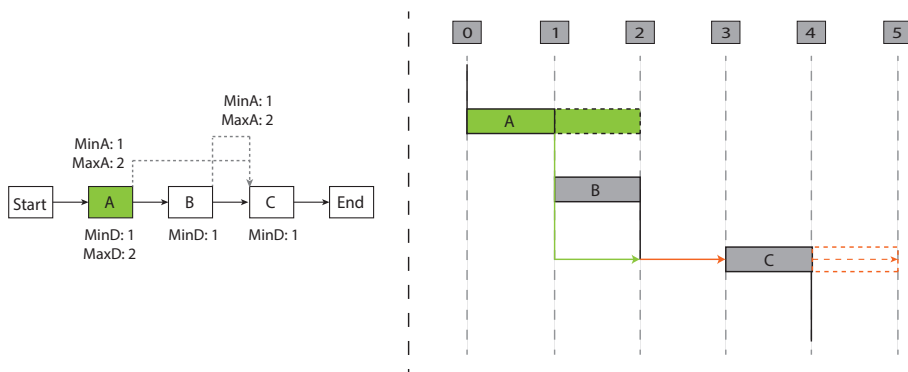


Abbildung 4.15: Darstellung von Zeitkanten mit selektierter Aktivität

4.4.4 Parallelität

Neben der Sequenz existiert standardmäßig noch die Parallelität. Bei einer Parallelität gibt es mehrere Zweige, in denen Aktivitäten zeitgleich ablaufen können. Durch diese verschiedenen Zweige ergeben sich bei der Vereinigung wieder verschiedene Anfangs-

und Endzeitpunkte für die Join-Aktivität (auch AND-Join genannt). Wie bei den aktivitätenübergreifend Zeitkanten wird für die Join-Aktivität bei der Min-Darstellung das Maximum der Anfangs- und Endzeitpunkte genommen, welche für die Join-Aktivität als Anfangs- bzw. Endzeitpunkt gesetzt wird und analog bei der Max-Darstellung das Minimum der Zeitpunkte. Dadurch existiert für die Zweige welche die Join-Aktivität nicht beeinflussen eine Pufferzeit (die Gesamtdauer eines Zweiges ist kleiner als die eines anderen, also existiert ein Abstand bis zur Join-Aktivität. Dieser Abstand wird als Pufferzeit bezeichnet).

Diese Pufferzeit wird im eGantt-Diagramm als gepunkteter Pfeil bei der Min-Darstellung hinter dem maximalen Abstand bei (wenn nicht vorhanden direkt hinter dem eGantt-Vorgang) oder bei der Max-Darstellung hinter dem eGantt-Vorgang angehängt. Wie die Pufferzeit genutzt wird, muss von Projekt zu Projekt beschlossen werden. Ob die Vorgänge später starten dürfen oder ob sie ihre Zeiten fix einhalten müssen, wird zu Beginn eines Projektes definiert.

In Abbildung 4.16 sieht man eine Parallelität. Die senkrechte Linie des Split-Vorgangs A (ein Vorgang mit mehreren Nachfolger) wird verlängert und die einzelnen Zweige an diesen Vorgang angehängt. Die Berechnungsvorschriften des einzelnen Vorgang sind in den Zweigen analog wie im Abschnitt 4.4.1. Dabei beeinflusst der Vorgang C den Join-Vorgang D und der Vorgang B besitzt eine Pufferzeit.

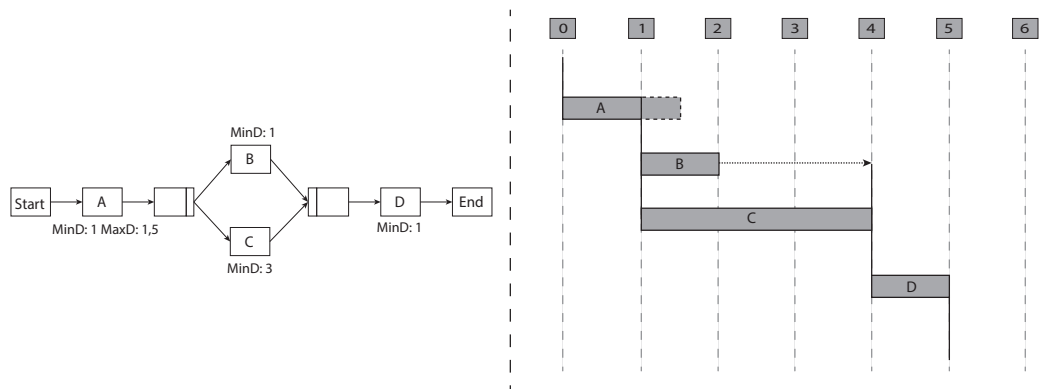


Abbildung 4.16: Parallelität in Min-Darstellung

In Abbildung 4.17 sieht man analog zu Abbildung 4.16 eine Parallelität. Jedoch existiert zwischen Aktivität A und Aktivität B eine Zeitkante.

Der eGantt-Vorgang wird wie in der Berechnungsvorschrift in Abschnitt 4.4.1 ausgerichtet. Hierdurch wird der Vorgang B im eGantt-Diagramm nach hinten verschoben. Die Pufferzeit wird in der Min-Darstellung an das gestrichelte Rechteck des maximalen Abstandes angehängt (wenn nicht vorhanden direkt hinter dem eGantt-Vorgang).

In Abbildung 4.18 wird in der Prozessansicht zwischen Aktivitäten in den verschiedenen Zweigen eine Zeitkante modelliert. Die Berechnung des Zielvorgangs verhält sich wie die Berechnungsvorschrift aus Abschnitt 4.4.1. Auch hier wird die Zielvorgang aufgrund des Zeitabstandes nach hinten verschoben.

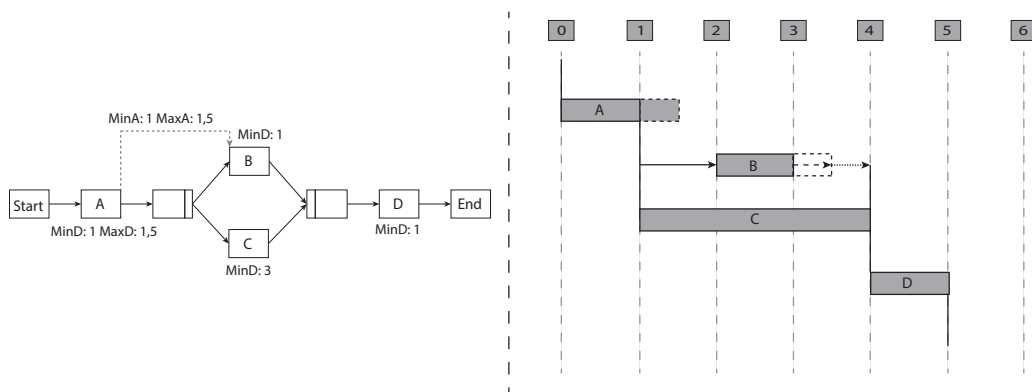


Abbildung 4.17: Parallelität mit Zeitkanten in Min-Darstellung

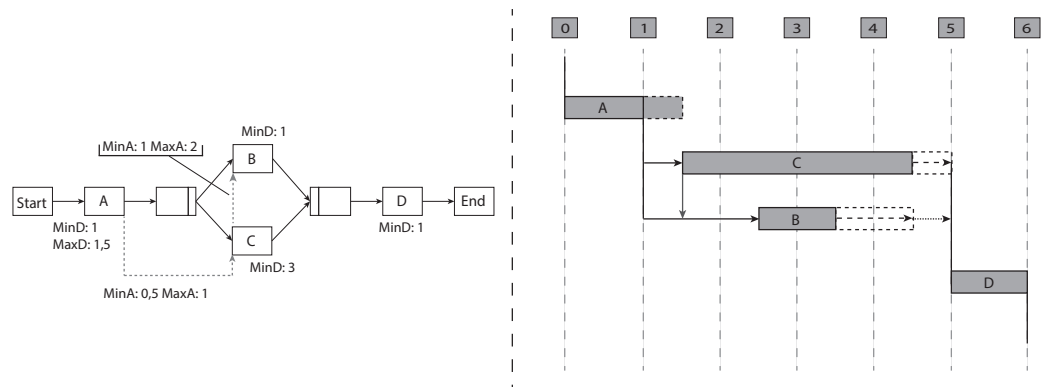


Abbildung 4.18: Zeitabstände zwischen Parallel-Verzweigungen in Min-Darstellung

4.5 XOR-Entscheidungen und Schleifen

In klassischen Gantt-Diagrammen existieren keine XOR-Entscheidungen und Schleifen. Diese beide Elemente sind in Prozesse jedoch essentiell. Aus diesem Grund erweitert eGantt das klassische Gantt-Diagramm um Möglichkeiten zur Darstellung dieser beiden Konzepte und komplettiert dadurch die meisten Eigenschaften der Prozesse. Zuerst wird in diesem Abschnitt die Abbildung der XOR-Entscheidungen behandelt und anschließend dann die von Schleifen.

4.5.1 XOR-Entscheidungen

Bei einer XOR-Entscheidungen wird zwischen mehreren Zweigen eine Auswahl getroffen. Durch die Auswahl verschiedener Zweige resultieren verschiedene Möglichkeiten der Berechnungen von Anfang- und Endzeitpunkte der Folgeaktivität des XOR-Block, welche als XOR-Join-Aktivität (analog zu AND-Join) fungiert. Bei der Min-Darstellung (Berechnung durch die minimale Dauer und den minimalen Abstand) und der Max-Darstellung (Berechnung durch die maximale Dauer und den maximalen Abstand) existieren dadurch für die Anfangs- und Endzeitpunkte jeweils ein bester Fall (*Best-Case*), der dem „kürzesten“ Zweig in der XOR-Entscheidung entspricht, und ein schlechtester Fall (*Worst-Case*) der dem „längsten“ Zweig in der XOR-Entscheidung entspricht. Der jeweilige Fälle (*Best-Case*

& *Worst-Case*) der Anfangs- und Endzeitpunkte werden mit einem Exponent markiert. Ein „+“ für den *Best-Case* und ein „-“ für den *Worst-Case*. Die sich daraus ergebende Aufteilung der Anfangs- und Endzeitpunkte ist in Abbildung 4.19 zu sehen. Falls mehr als zwei Zweige in einer XOR-Entscheidung existieren, entspricht ein Zweig dem *Best-Case*, ein Zweig dem *Worst-Case* und die restlichen Zweige haben keine eindeutige Bedeutung.

Die Berechnungsvorschriften für die einzelnen Vorgängen in den Zweigen der XOR-Entscheidungen sind die selben wie im Abschnitt 4.4.1. Die einzelnen Zweige der XOR-Entscheidung im eGantt-Diagramm sind der besseren Unterscheidbarkeit wegen mit einem grauen Rechteck umschlossen. Zur Verdeutlichung, dass es sich um eine XOR-Entscheidung handelt wird an der rechten-oberen Ecke eine XOR-Markierung angebracht wie sie in Abbildung 4.20 zu sehen ist.

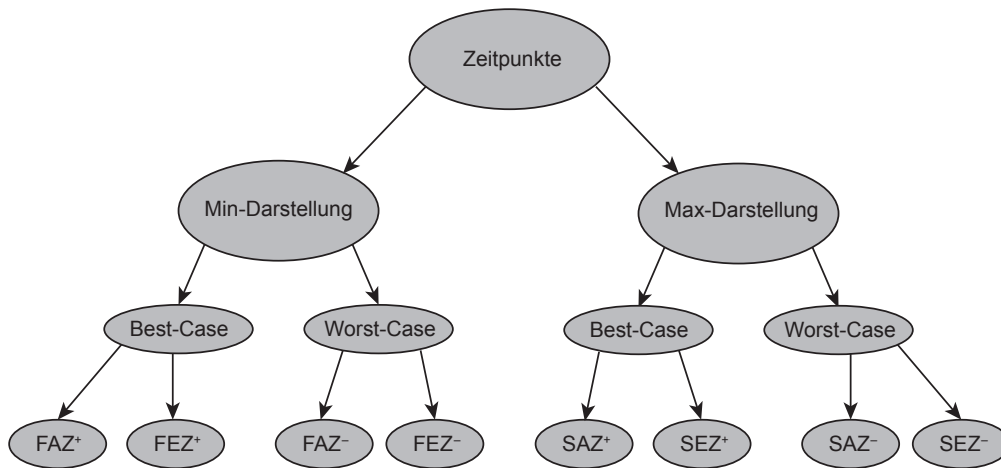


Abbildung 4.19: Erweiterte Anfangs und Enzeitpunkte

Für den Folgevorgang des XOR-Blockes gilt jedoch aufgrund der in Abbildung 4.20 dargestellten Problematik der unterschiedlichen Zweig-Dauern (*Best-Case*, *Worst-Case*) nicht die normale Berechnungsvorschrift aus Abschnitt 4.4.1. Für das Fragezeichen in Abbildung 4.20 ist deswegen eine neue Berechnungsvorschrift zu finden. Diese ist für die Min-Darstellung in Abbildung 4.21 zu sehen. Zusätzlich zu FAZ und FEZ des Folgevorgang, sind weitere Zeiteigenschaften die keine genaue Bedeutung haben zu sehen. Hierbei wird am besten

Falls des Vorgängervorgang (FEZ_X^+) die einzelnen Zeitpunkte der Nachfolgervorgang ausgerichtet. Die minimale/maximale Dauer und des minimale/maximale Abstand sind analog zu der vorigen Berechnungsvorschrift benannt. In der Berechnungsvorschrift werden der Best-Case des FAZ (FAZ_Y^+) und FEZ (FEZ_Y^+) des Vorgangs berechnet.

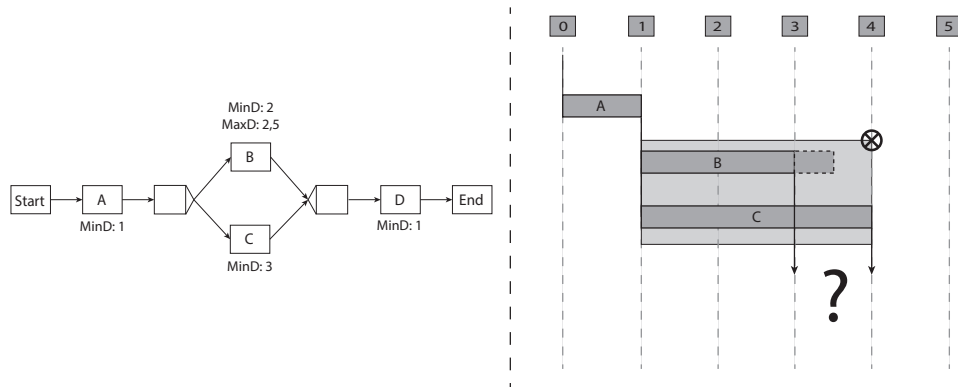


Abbildung 4.20: Ausgangsproblem einer XOR-Entscheidung in Min-Darstellung

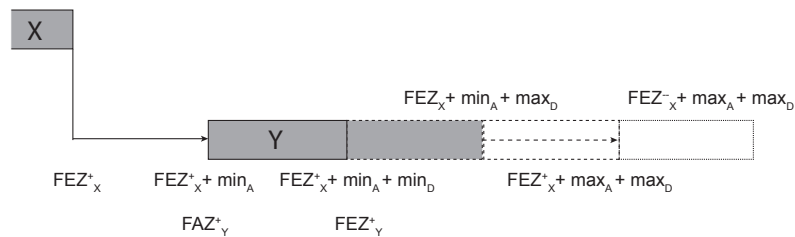


Abbildung 4.21: Berechnung des Folgevorgangs in Min-Darstellung

In Abbildung 4.22 ist die Berechnungsvorschrift aus der Abbildung 4.21 angewandt. Der eGantt-Vorgang ist dabei am besten Fall des FAZ ausgerichtet, also am FEZ des „kürzesten“ Zweiges plus dem minimalen Abstand zum Vorgängervorgang. Hinter dem maximalen Abstand des eGantt-Vorgangs wurde ein gepunktetes Rechteck angehängt. Das Ende des gepunkteten Rechtecks ist dabei der FEZ des „längsten“ Zweiges plus dem maximalen Abstand zum Vorgängervorgang und dessen maximalen Dauer. Das gepunktete Rechteck

wird nur für den Folgevorgang eines XOR-Blockes gezeigt. Für weitere eGantt-Vorgänge wird dieses gepunktete Rechteck nicht betrachtet, da es für weitere Vorgänge nur bedingt relevant ist. In Abbildung 4.22 ist die Lösung für das „Fragezeichen“ dargestellt.

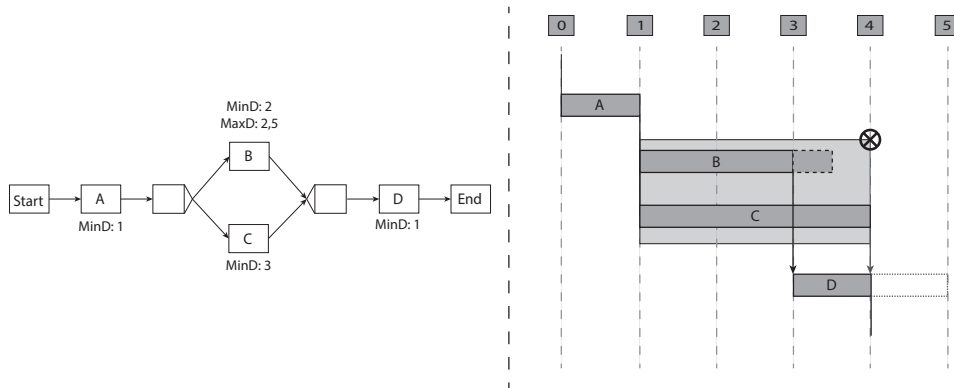


Abbildung 4.22: XOR-Entscheidung in Min-Darstellung

Analog zur Min-Darstellung ergibt sich bei der Max-Darstellung (Abbildung 4.24) das selbe Problem und wird durch die Erweiterung der Berechnungsvorschrift (welche in Abbildung 4.23 gezeigt wird) behoben. Dabei wird der Folgevorgang auf den schlechtesten SAZ gesetzt, also der SEZ des „längsten“ Zweiges plus die maximale Dauer des Vorgängervorgang plus den maximalen Abstand zum Vorgängervorgang. Der Anfang des gepunkteten Rechtecks stellt dabei den besten SAZ („kürzester“ Weg) dar. Analog wie bei der Min-Darstellung wird dieses gepunktete Rechteck für weitere eGantt-Vorgänge im eGantt-Diagramm nicht weiter übernommen. In Abbildung 4.24 ist die Anwendung der Berechnungsvorschrift aus Abbildung 4.23 zu sehen.

Zeitkanten zwischen Aktivitäten Außerhalb des XOR-Blockes und einer einzelnen Aktivitäten innerhalb des XOR-Blockes sind problematisch, da durch Abwahl bzw. Auswahl eines Zweiges der Prozess und dessen Zeiten stark verändert werden. Durch die gepunkteten Rechtecke der eGantt-Aktivität nach dem XOR-Block lassen sich allgemein die besten und schlechtesten Fälle ablesen. Doch Einzelfallbewertungen können nicht angezeigt werden. Im weiteren folgen nun zwei Beispiele (Abbildung 4.25/4.26 und Abbildung 4.27/4.28). Diese stellen das Problem mit Zeitkanten zwischen Aktivitäten außerhalb und innerhalb eines XOR-

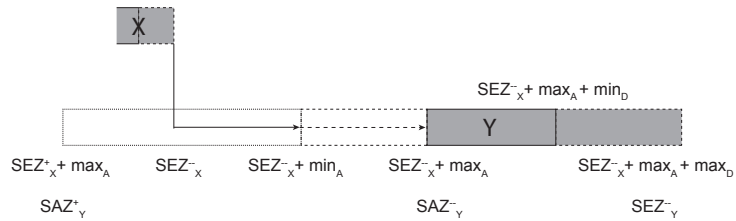


Abbildung 4.23: Berechnung des Folgevorgang in Max-Darstellung

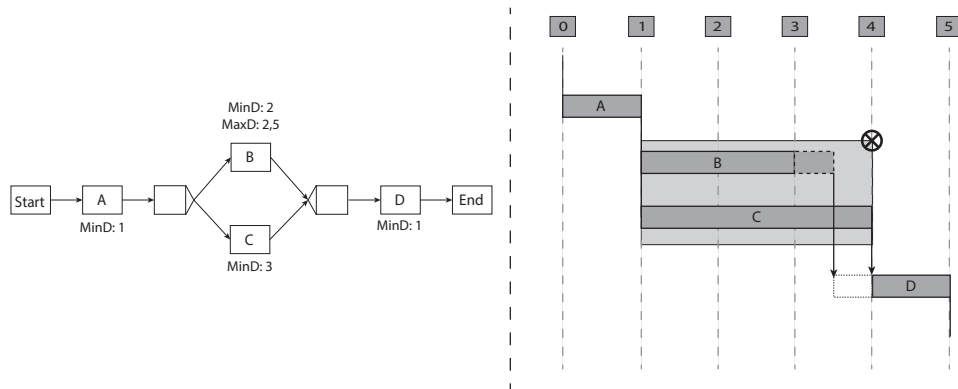


Abbildung 4.24: XOR-Block in Max-Darstellung

Blockes dar. Dieses Problem wird durch Kombination der einzelnen Berechnungsvorschriften aus Abschnitt 4.4.1 und den obigen Berechnungsvorschriften aus Abbildung 4.21 & 4.23 gelöst.

4.5.2 Schleifen

Um die Konstrukte der eGantt-Diagramme zu komplettieren fehlt noch ein Schleifen-Konstrukt. Schleifen werden dabei im klassischen Gantt-Diagramm nicht unterstützt. Schleifen sind Wiederholungen und haben am Ende der Schleife die Möglichkeit die Schleife zu verlassen oder nochmals eine Iteration durchzuführen. Dadurch kann ein Vorgang zwischen verschie-

4.5 XOR-Entscheidungen und Schleifen

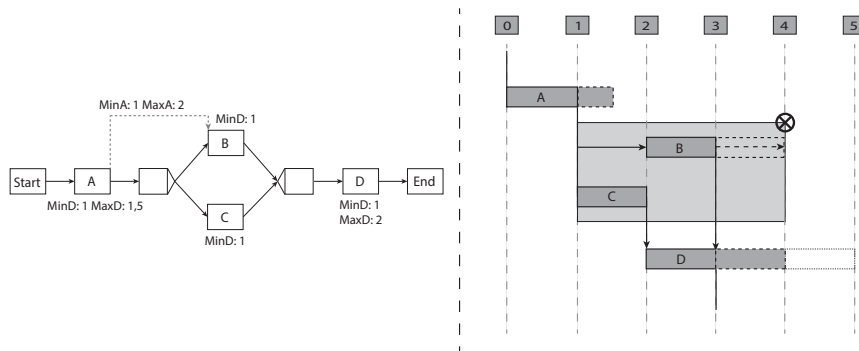


Abbildung 4.25: XOR-Block mit Zeitkanten in Min-Darstellung

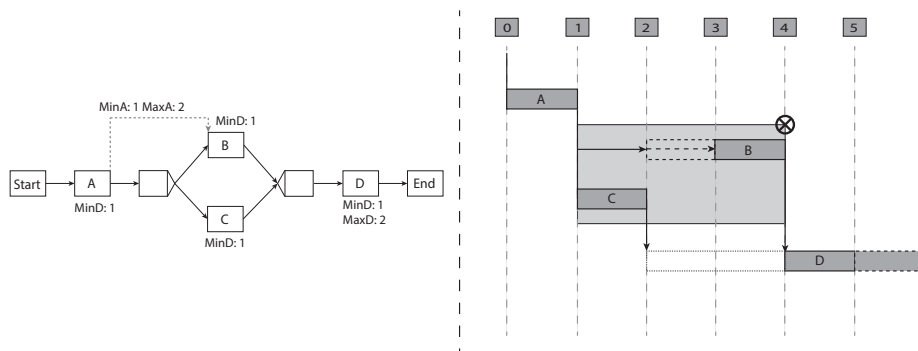


Abbildung 4.26: XOR-Block mit Zeitkanten in Max-Darstellung

denen Iterationen zu sich selbst einen zeitlichen Abstand besitzen und zu dem Folgevorgang der Schleife. Um dies darstellen zu können muss man im eGantt-Diagramm eine Lösung finden, um diese zwei zeitlichen Abstände gleichzeitig darstellen zu können. Dies wird durch einsetzen eines Dummyvorgang gelöst. Dieser Dummyvorgang stellt den Vorgang (bei mehr als einen Vorgang im Schleifenblock, ist es eine Vereinigung aller Vorgängen in dem Schleifenblock, ähnlich wie eine Phase), welcher innerhalb des Schleifenblocks ist, für die nächste Iteration nochmals dar. Dieser Dummyvorgang wird ausgegraut unterhalb des letzten Vorgangs der vorigen Iteration im Schleifenblock angeordnet, ähnlich wie bei einem AND-Join (siehe Abbildung 4.29). Jedoch wird die Ausgangsverbindung des Dummyvorgangs (senkrechte Linie mit waagrechten Pfeil) so verlängert, dass der nächsten FAZ des

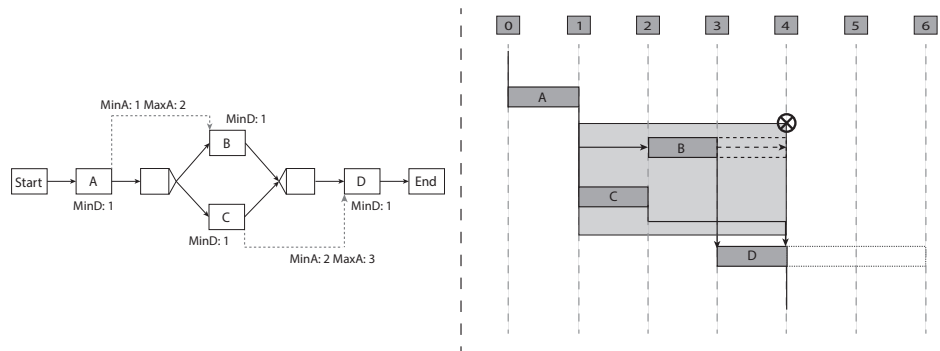


Abbildung 4.27: XOR-Block mit ausgehender Zeitkanten in Min-Darstellung

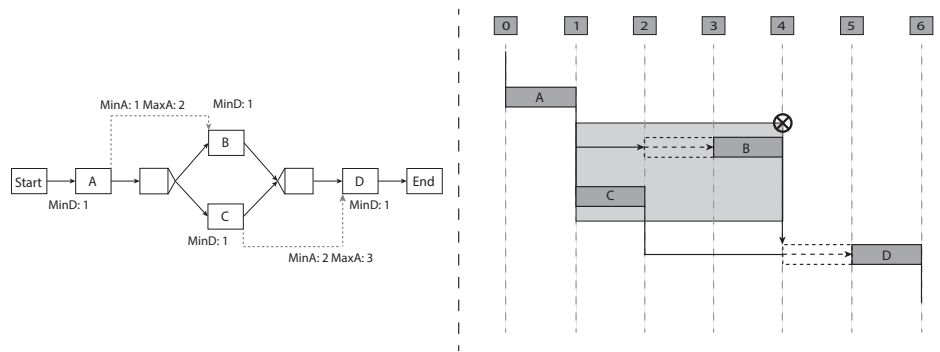


Abbildung 4.28: XOR-Block mit ausgehender Zeitkanten in Max-Darstellung

Vorgang außerhalb des Schleifenblocks darstellt wird. Also wird der Folgevorgang, welcher nach der Schleife folgt, für die nächste Iteration angedeutet. Zudem wird die Schleife, um sie zu verdeutlichen und zu bündeln mit einem gestrichelten Rechteck umrahmt und mit einem Schleifensymbol rechts-oben markiert. Zudem gibt es wie bei der XOR-Entscheidung eine Verlängerung des eGantt-Folgevorgangs durch ein rechts-offenes gepunktetes Rechteck. Dieses Rechteck verdeutlicht die Möglichkeit weiterer Iterationen.

In Abbildung 4.29 ist die Min-Darstellung einer Schleife und in Abbildung 4.30 ist die Max-Darstellung einer Schleife zu sehen. Analog zu der Min-Darstellung wird die Schleife mit einem gepunkteten Rechteck umrahmt und der Dummyvorgang ist dabei wie in der Min-Darstellung ausgegraut. Der nächste FAZ in der Min-Darstellung bzw. SAZ in der Max-

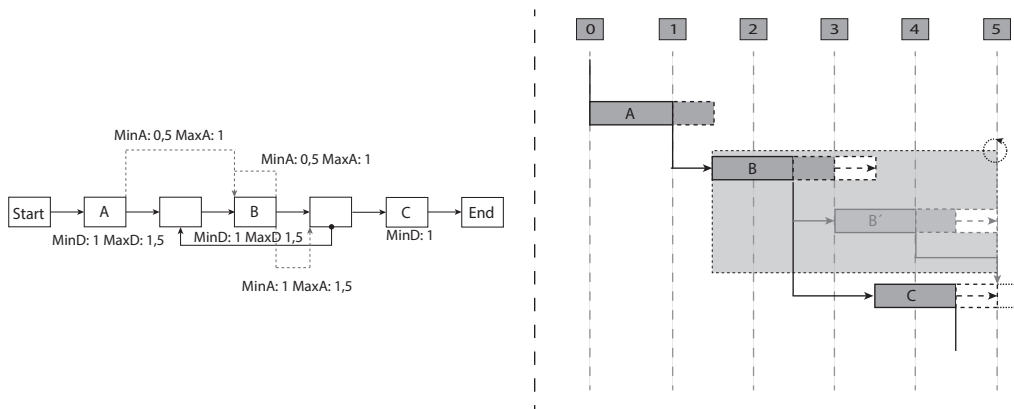


Abbildung 4.29: Schleife in Min-Darstellung

Darstellung des Folgevorgang nach dem Schleifenblock ist durch die Verbindung, welche den Dummyvorgang verlässt angedeutet.

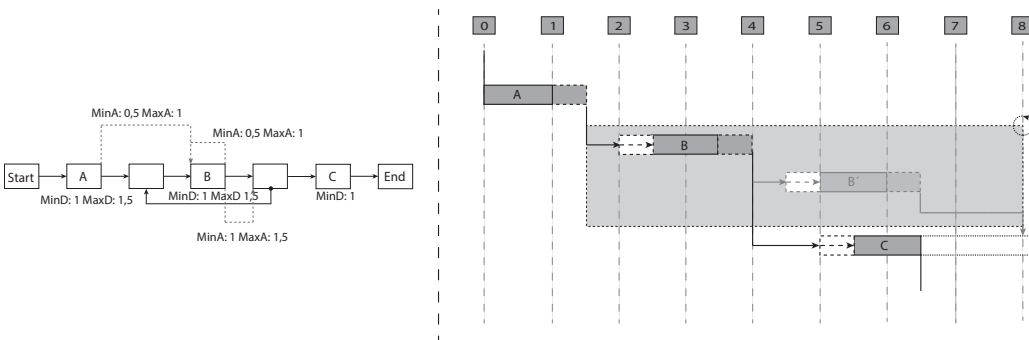


Abbildung 4.30: Schleife in Max-Darstellung

4.6 Beispiel

Nach dem Abarbeiten der einzelnen Abbildungsvorschriften, zeigt 4.31 ein größeres Beispiel eines eGantt-Diagramm in Min-Darstellung. Dieses Beispiel soll verdeutlichen, dass auch komplexere Prozesse in eGantt leicht darstellbar sind und durch die Unterstützung der

Kapitel 4 Abbildung blockstrukturierter Prozessgraphen auf Gantt-Diagramme

Prozessansicht gut lesbar sind. Der Prozess in 4.31 dient nur der Veranschaulichung und ist auf der Prozessebene nicht optimal modelliert, einzelne XOR-Blöcke könnten beispielsweise zusammengefasst werden.

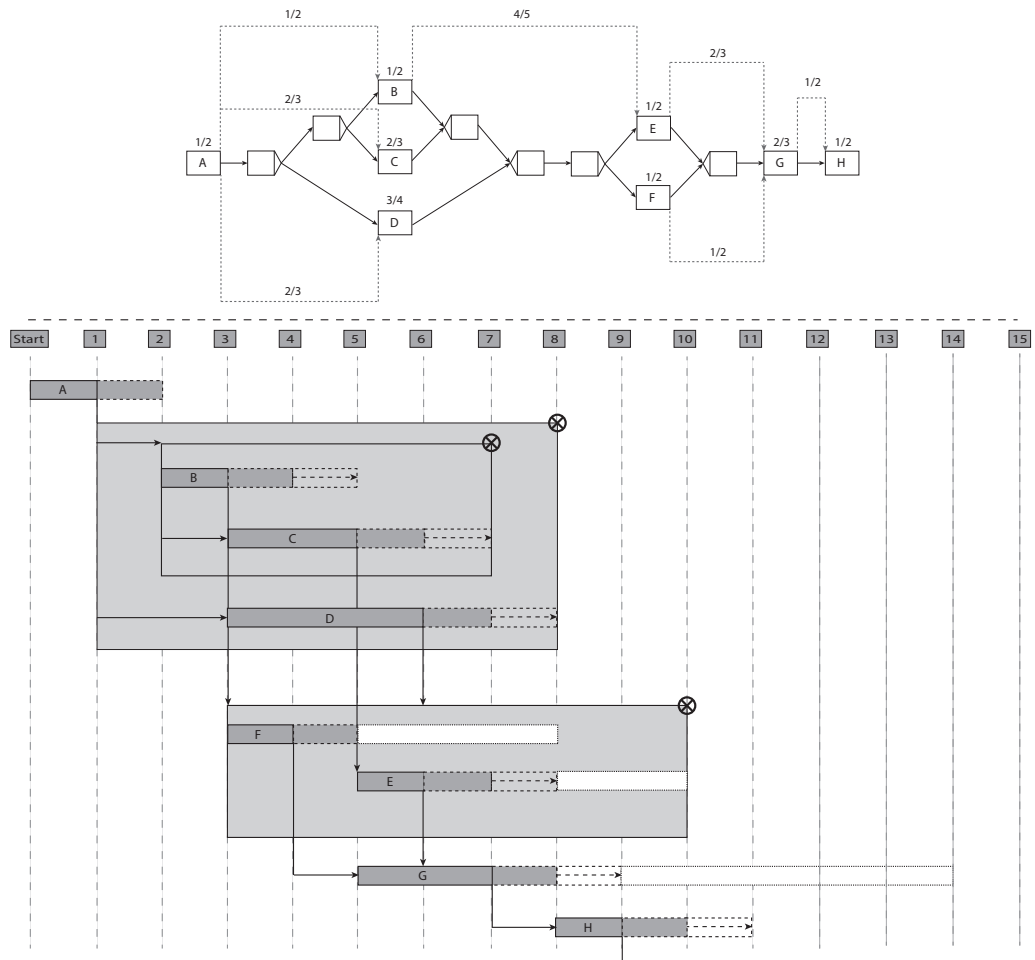


Abbildung 4.31: Komplexeres Prozessbeispiel mit mehreren Abhängigkeiten

Nach der Besprechung der einzelnen Abbildungen von blockbasierten Prozessen auf die eGantt-Diagramme, wird im nächsten Kapitel das Verhalten der einzelnen Abbildungen auf der Ausführungsebene besprochen.

5

Verhalten von eGantt auf Ausführungsebene

Nach den einzelnen Abbildungen von blockbasierten Prozessen auf eGantt im vorigen Kapitel, behandelt diese Kapitel das Verhalten dieser Abbildungen bei der Ausführung eines Prozesses. Analog wie bei den Abbildungen im vorigen Kapitel existieren auch auf der Ausführungsebene zwei verschiedene Darstellungen, die Min-Darstellung und die Max-Darstellung. Beide Darstellungen werden dabei in diesem Kapitel erklärt und betrachtet, da sie sich zum Teil unterschiedlich verhalten.

5.1 Knotenzustände in ADEPT und eGantt

Bei der Ausführung werden die einzelnen Aktivitäten im ADEPT-Metamodell mit einem Zustand bewertet. Dieser Zustand kann sich allgemein bei Aktivierung, Deaktivierung, usw. einer Aktivität ändern. Die einzelnen Zustände und ihre Übergänge sind in Abbildung 5.1 gezeigt. Dabei wurde das Zustandsdiagramm des ADEPT-Metamodell um einen weiteren Zustand *Blocked* erweitert. Dieser Zustand ist für die Zeiterweiterung des ADEPT-Metamodell nötig, um das warten auf die Erfüllung eines minimalen Abstand zu realisieren.

Um die Ausführung von eGantt-Diagrammen zu realisieren, müssen die Zustände die im ADEPT-Metamodell existieren auf die eGantt übertragen werden. Dabei sind nicht alle Zustände aus dem ADEPT-Metamodell in eGantt von Bedeutung bzw. umgesetzt. Folgende Zustände werden in eGantt-Diagrammen umgesetzt und abgebildet: *NotActivated*, *Activated*, *Started*, *Blocked*, *Skipped* und *Completed*.

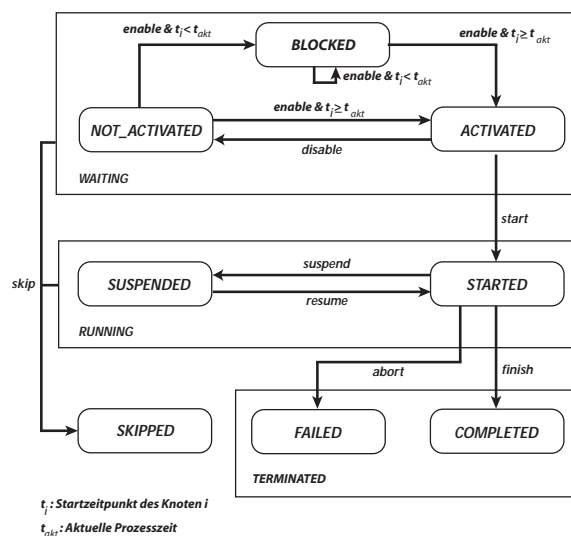


Abbildung 5.1: ADEPT Knotenzustände mit Zeiterweiterung (in Anlehnung an [Rei00])

Jeder dieser einzelnen Zustände ist in eGantt umgesetzt und ist in Abbildung 5.2 dargestellt. Bei dem Zustand *Activated* wird der momentane Vorgang mit der Farbe Blau gefärbt. Ist ein

Vorgang *Started* wird er Grün eingefärbt. Beim abwählen eines Vorgang (*Skipped*) wird der Vorgang ausgeblendet. Ist der Vorgang abgeschlossen (*Complete*) wird er ausgegraut. Bei einer Blockierung (*Blocked*) des Vorgangs durch einen minimalen Abstand wird der Vorgang orange eingefärbt. Ist der Vorgang noch nicht Aktiviert worden (*NotActivated*) bleibt er im Standard Grau.

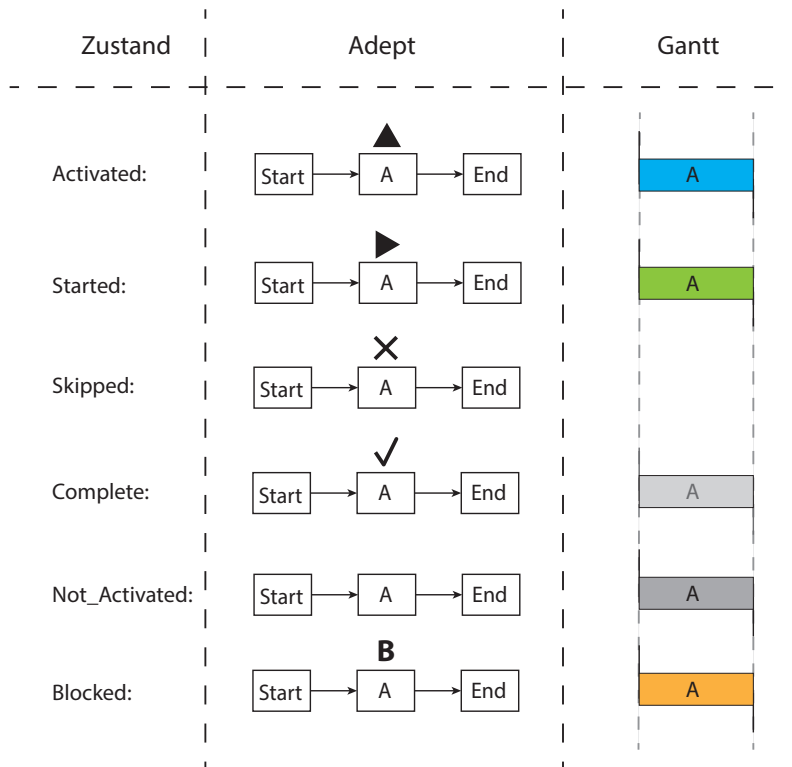


Abbildung 5.2: Abbildung der Knotenzustände in eGantt

Durch die Einführung der Zustände werden die einzelnen Aktivitäten bzw. Vorgänge hinsichtlich ihrer zeitlichen Eigenschaften bewertet. Doch braucht man zur zeitlichen Orientierung im Prozess zusätzlich eine Zeitlinie. Diese Zeitlinie beschreibt die momentane Prozesszeit. Durch diese Prozesszeit können die einzelnen Aktivitäten bzw. Vorgänge und ihre Zustände erst bewertet werden.

5.2 eGantt-Zeitlinie

Die eGantt-Zeitlinie (siehe Abbildung 5.3) bewegt sich im eGantt-Diagramm anhand der Zeitskala von links nach rechts. Wobei links im eGantt-Diagramm der Startzeitpunkt des Prozesses ist. Dieser Startzeitpunkt ist dabei das Datum und die Uhrzeit bei Start des Prozesses. Nach Start des Prozesses verschiebt sich die Zeitlinie anhand des momentanen Datum und Uhrzeit immer weiter der Zeitskala entlang in Richtung Endzeitpunkt des Prozesses. Die eGantt-Zeitlinie bewertet dabei auch einzelne Zustände der Vorgänge. Ob ein Vorgang blockiert ist oder gestartet werden kann, kann durch die Zeitlinie und den momentan beschrittene Abstände bzw. Vorgänge bestimmt werden.

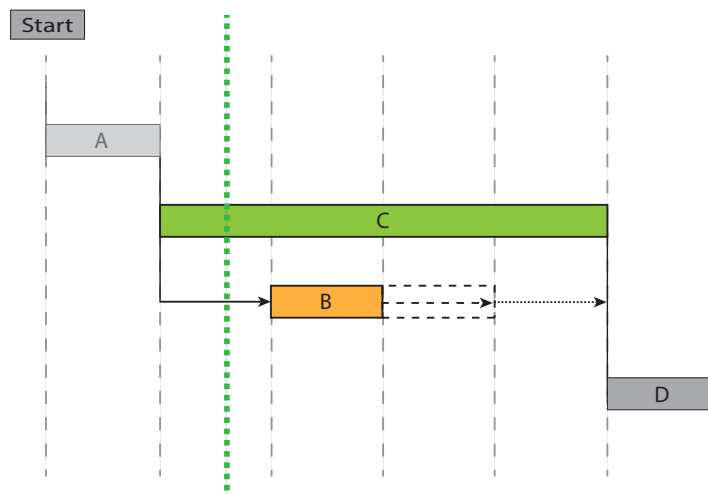


Abbildung 5.3: Timeline mit bewerteten Knoten

Nach der Behandlung der Zustände und der eGantt-Zeitlinie, werden nun Beispiele und einzelne Darstellungsarten mit ihrem Verhalten auf der Ausführungsebene besprochen.

5.3 Verhalten der eGantt-Diagramme auf Ausführungsebene

Auf der Ausführungsebene existieren analog zur Modellierungszeit zwei verschiedene Darstellungsarten. Einmal die Min-Darstellung die den FAZ und den FEZ der einzelnen Vorgänge im eGantt-Diagramm anzeigt und die Max-Darstellung die den SAZ und SEZ der einzelnen Vorgänge beschreibt.

Beim initialisieren einer Prozessinstanz aus einem Prozess wird die Startaktivität ausgeführt. Nach dem beenden der Startaktivität kann die erste Aktivität im Prozess bzw. der erste Vorgang im eGantt-Diagramm gestartet und zusätzlich die eGantt-Zeitlinie aktiviert werden. Je nach Dauer des Vorgangs bleibt dieser Vorgang im Zustand *Started*, nach Erreichen der minimalen Dauer kann der Vorgang nun beendet werden. Wird die maximale Dauer überschritten wird ein Fehler gemeldet. Die Fehlerbehandlung ist dabei jedoch nicht Teil dieser Arbeit.

Analog werden die Zeitabstände bewertet. Dabei muss der minimale Abstand durch die Zeitlinie komplett abgehandelt werden. Während des beschreitens des minimalen Abstandes ist der nachfolgende Vorgang blockiert. Erst nach überschreiten des minimalen Abstand wird der Zustand des nachfolgenden Vorgang auf aktiviert gesetzt. Liegt der Vorgang direkt nach dem minimalen Abstand und der maximale Abstand ist gleich dem minimalen Abstand muss der Vorgang nach überschreiten des minimalen Abstand sofort gestartet werden. Existiert aber ein maximaler Abstand der größer ist als der minimale Abstand, muss der Vorgang nicht gleich gestartet werden. Dieser Vorgang kann weiter noch aktiviert bleiben. Jedoch beim Überschreiten des maximalen Abstandes muss der Vorgang gestartet werden.

Wie oben schon erwähnt gibt es analog wie in den vorigen Kapitel zwei verschiedene Darstellungsarten. In Abbildung 5.4 ist eine Ausführung eines einfachen Prozesses in Min-Darstellung zu sehen. Links in der Abbildung ist dabei immer die Prozessansicht mit ihren Zuständen (ADEPT) und rechts ist das eGantt-Diagramm mit den bewerteten Vorgängen zu sehen. In Schritt 1 ist in Abbildung 5.4 der Prozessgraph und das eGantt-Diagramm nach der Modellierung zu sehen. In Schritt 2 wurde die Startaktivität beendet und die Folgeaktivität aktiviert. Im selben Moment nach der Aktivierung wurde in Schritt 3 die Aktivität gestartet. Dieser Vorgang musste gleich nach der Aktivierung gestartet werden, da kein maximal

Abstand oder irgendwelcher Puffer zur Verfügung stehen. In Schritt 4 ist der Prozess schon eine Zeit weiter voran gegangen. Die eGantt-Zeitlinie hat dabei die maximale Dauer des Vorgangs A beschränkt. In der Min-Darstellung werden dabei alle Folgevorgänge im eGantt-Diagramm bei beschreiten der maximalen Dauer an der Zeitlinie neu ausgerichtet. Somit verschieben sich alle Folgevorgänge anhand der eGantt-Zeitlinie (siehe Abbildung 5.4 Schritt 4 Vorgang B). In den weiteren Schritten in Abbildung 5.4 wird Vorgang B gestartet und im letzten Schritt wird er Beendet, da die eGantt-Zeitlinie den Vorgang verlässt.

In Abbildung 5.5 sieht man die Min-Darstellung eines Prozesses. Im Vergleich zu der vorigen Abbildung ist dieser Prozess mit minimalen und maximalen Abständen (Zeitkanten) modelliert. Bei diesem Prozess verhalten sich minimale und maximale Dauer gleich wie im vorigen Beispiel. Jedoch sind die Zeitkanten bzw. die Abstände neu. Diese Abstände verhalten sich jedoch eigentlich simultan zu der Dauer. Überschreitet die eGantt-Zeitlinie die minimale Dauer, werden die Folgevorgänge anhand der eGantt-Zeitlinie mit nach Rechts verschoben (siehe Abbildung 5.5 Schritt 6). Der Vorgang in Schritt 6 ist dabei ab dem minimal Abstand Aktivierbar (davor ist der Vorgang blockiert), jedoch muss er erst nach Erreichen des maximalen Vorgangs gestartet werden.

In Abbildung 5.6 ist die Max-Darstellung eines Prozesses zu sehen. Wie in den vorigen Kapitel erwähnt, werden die Vorgänge an den SAZ und SEZ angeordnet. Bei dieser Darstellungsart ist eine Verschiebung anhand der eGantt-Zeitlinie nicht möglich. Beim Beenden eines Vorgangs, werden die Folgevorgänge meist weiter nach Links gerückt (siehe Abbildung 5.6 Schritt 5 & 6), außer jeder Vorgang dauert seine maximale Zeit und die Abstände sind zusätzlich maximal.

Parallelitäten verhalten sich analog zu Sequenzen in den beiden Darstellungsarten. Jeder einzelne Zweig der Parallelität wird dabei wie eine Sequenz betrachtet. Der Unterschied ist dabei nur, dass es zwei verschiedene Zweige gibt und diese jeweils beide abgearbeitet sein müssen, damit der Join-Vorgang aktivierbar ist.

In einem XOR-Block kann man verschiedene Zweige auswählen bzw. abwählen. In der Abbildung 5.7 sieht man die Min-Darstellung eines Prozesses mit einer XOR-Entscheidung. In Schritt 3 wird dabei ausgewählt, welcher Folgevorgang ausgewählt bzw. abgewählt wird (in diesem Fall wird Vorgang B abgewählt und Vorgang C ausgewählt). Der abgewählte Vorgang wird anschließend ausgeblendet (Zustand: *skipped*) und der ausgewählte Vorgang

versucht zu aktivieren (bei einem minimalen Abstand kann er auch blockiert sein). Zur Verdeutlichung, dass in diesem Diagramm eine Entscheidung stattgefunden hat, wird das Rechteck mit der Markierung beibehalten. Bei verlassen des XOR-Blockes, wird dieser auch mit ausgegraut.

Zuletzt fehlt noch die Ausführung einer Schleife. Bei Schleifen können theoretisch unendlich viele Iterationen auftreten. Durch diese Iterationen verschieben sich alle Folgevorgänge um die Gesamtdauer des Schleifeninhalts nach hinten. Auf der Modellierungsebene haben wir den Dummyvorgang eingeführt. Jedoch müssen bei der Ausführung mehrere Iterationen berücksichtigt werden. Deswegen wird der Dummyvorgang für jede Iteration neu gesetzt und berechnet. In Abbildung 5.8 & 5.9 sieht man eine Ausführung einer Schleife mit einer Iteration. In Schritt 5 wird die Entscheidung getroffen, dass nochmals ein Durchlauf der Schleife stattfindet. In Schritt 6 wird der Dummyvorgang der vorigen Iterationen zu einem „normalen“ Vorgang, jedoch wird die Benennung des Vorgangs mit einem Index versehen. Für die nächste Iteration wird wieder ein Dummyvorgang erstellt und an den Dummyvorgang der vorigen Iteration angehängt. Anhand des neuen Vorgang, verschieben sich alle Folgevorgänge der Schleife um eine Iteration nach hinten. In Schritt 7 wird dieses mal Entschieden die Schleife zu verlassen. Der momentane Dummyvorgang der ausgegraut zu sehen ist, wird ausgeblendet und das gestrichelte Rechteck der Schleife wird angepasst und ausgegraut.

Nach der Behandlung der Ausführungsebene, wird im nächsten Kapitel ein Prototyp, welcher die vorgestellte Abbildungen realisiert und Ideen für Implementation für die Modellierungsebene gezeigt und besprochen. Dieser Prototyp zeigt, dass die Abbildungen aus dem vorigen Kapitel in der Realität und Programmier technisch überhaupt umsetzbar sind.

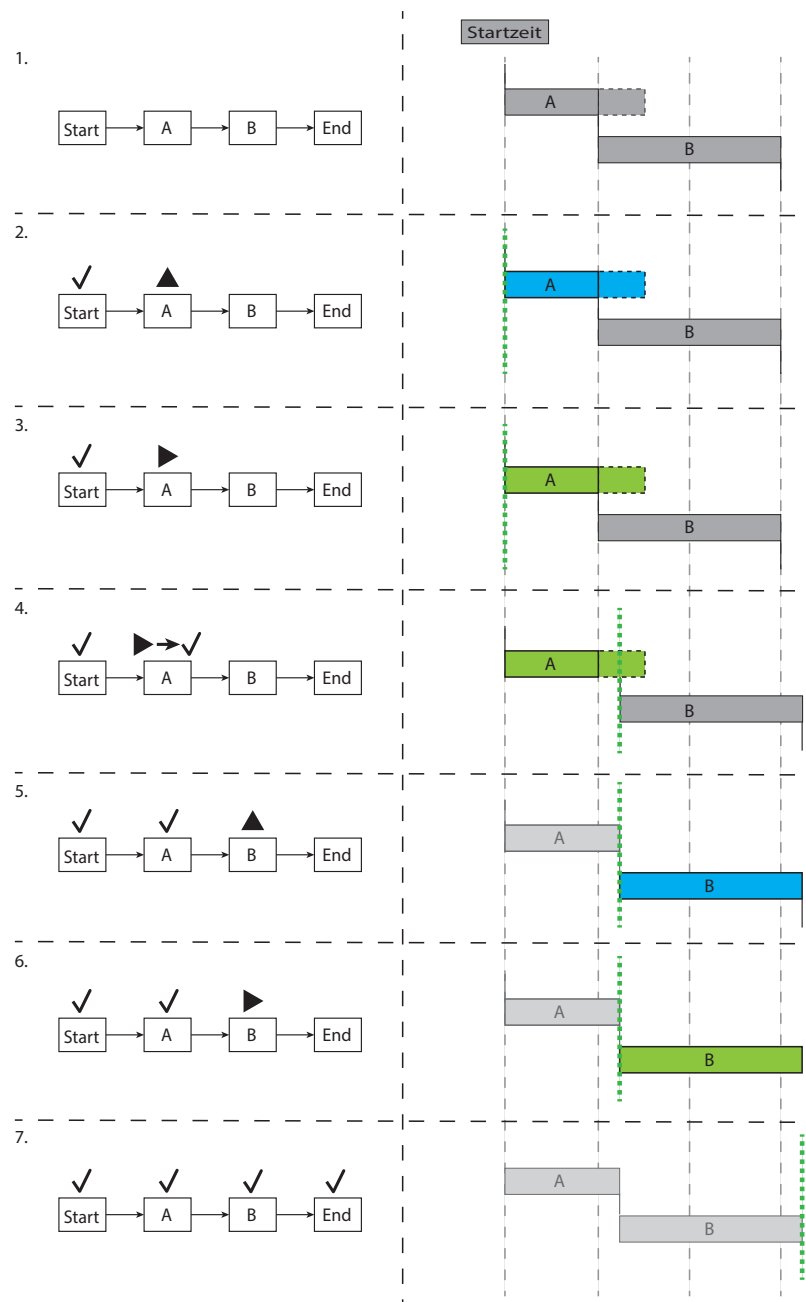


Abbildung 5.4: Ausführung einer Sequenz in Min-Darstellung

5.3 Verhalten der eGantt-Diagramme auf Ausführungsebene

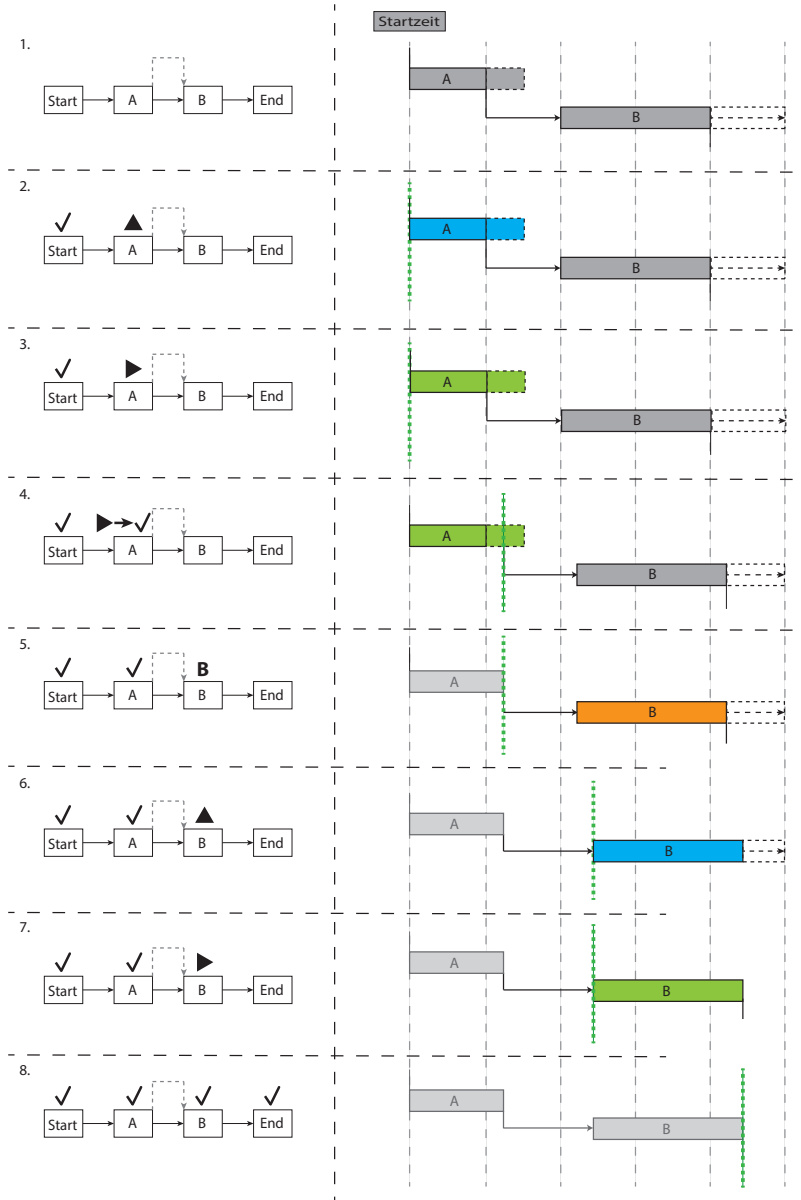


Abbildung 5.5: Sequenz mit zeitlichen Abständen in Min-Darstellung

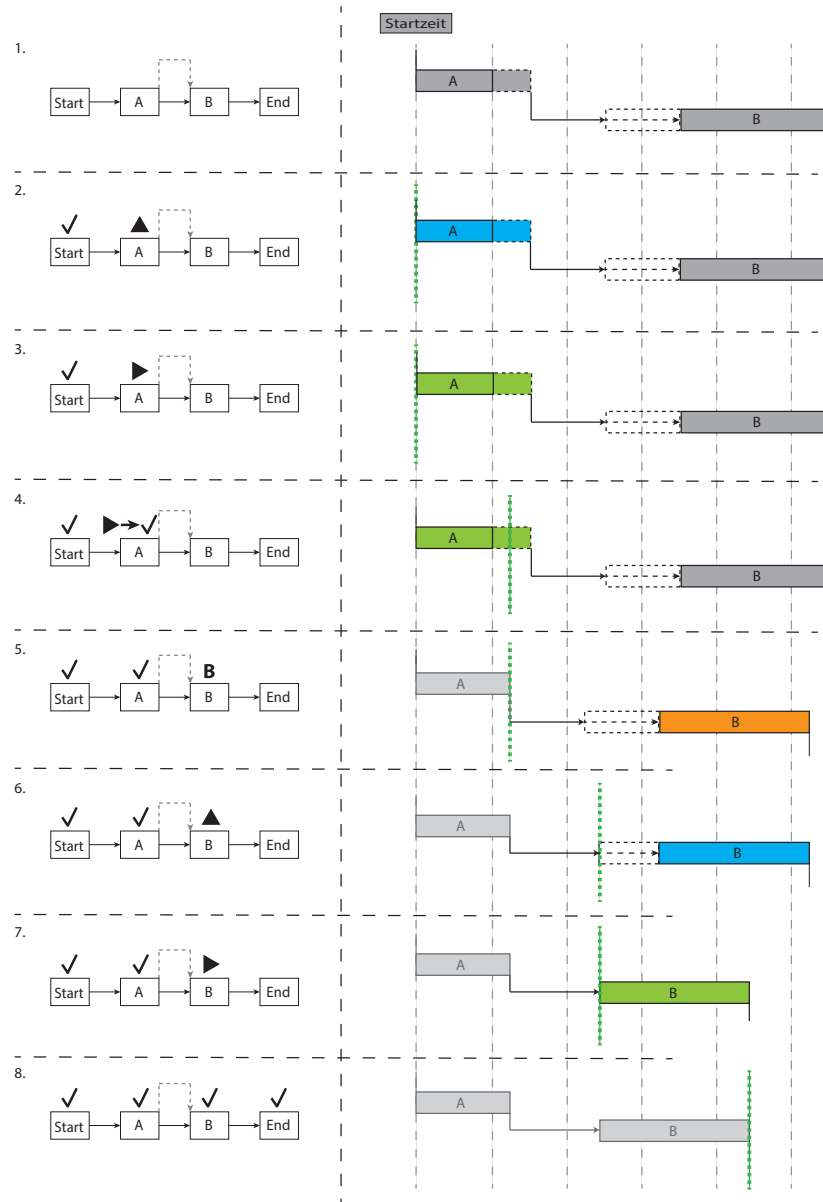


Abbildung 5.6: Sequenz mit zeitlichen Abständen in Max-Darstellung

5.3 Verhalten der eGantt-Diagramme auf Ausführungsebene

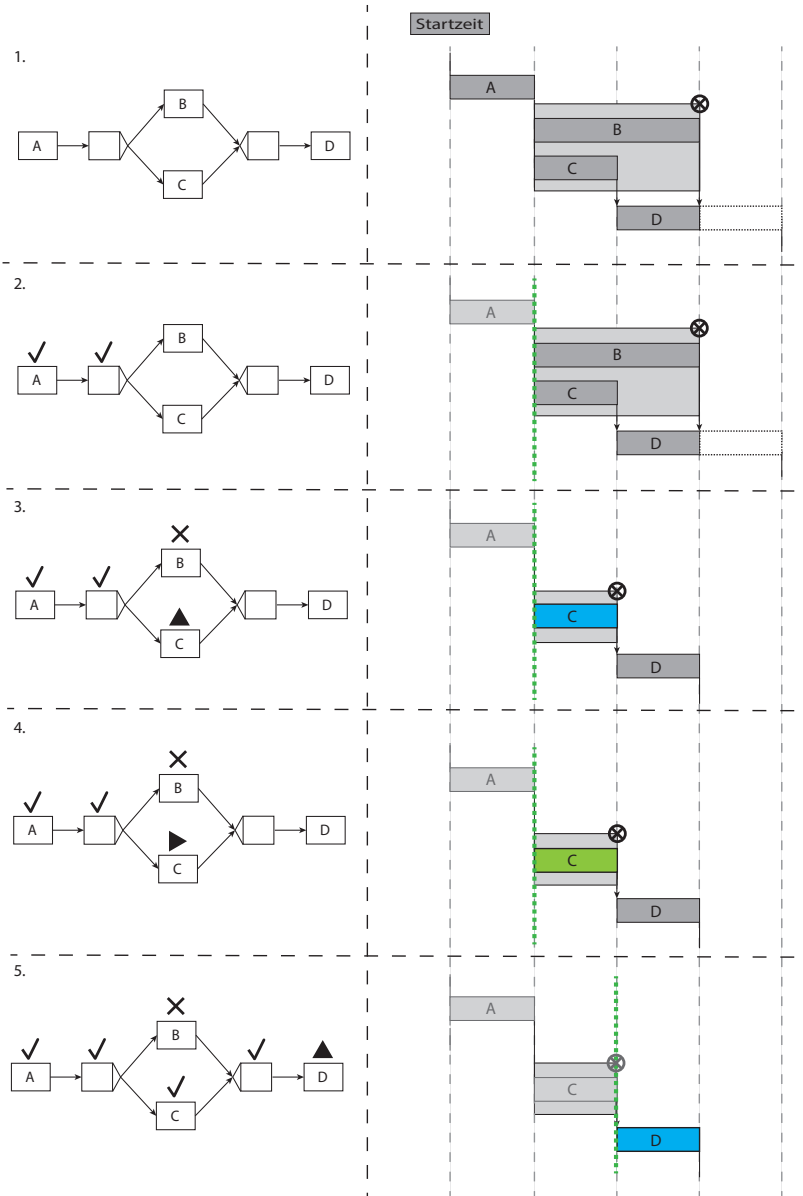


Abbildung 5.7: Ausführung eines XOR-Blockes in Min-Darstellung

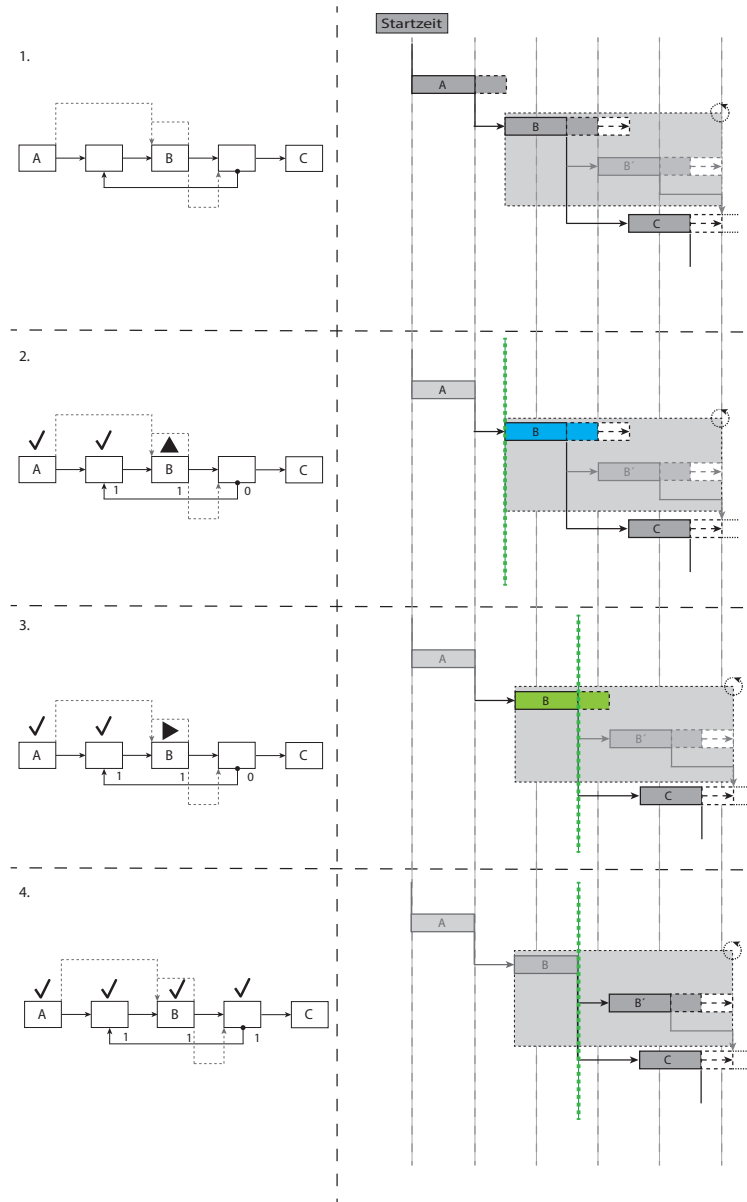


Abbildung 5.8: Ausführung einer Schleife in Min-Darstellung

5.3 Verhalten der eGantt-Diagramme auf Ausführungsebene

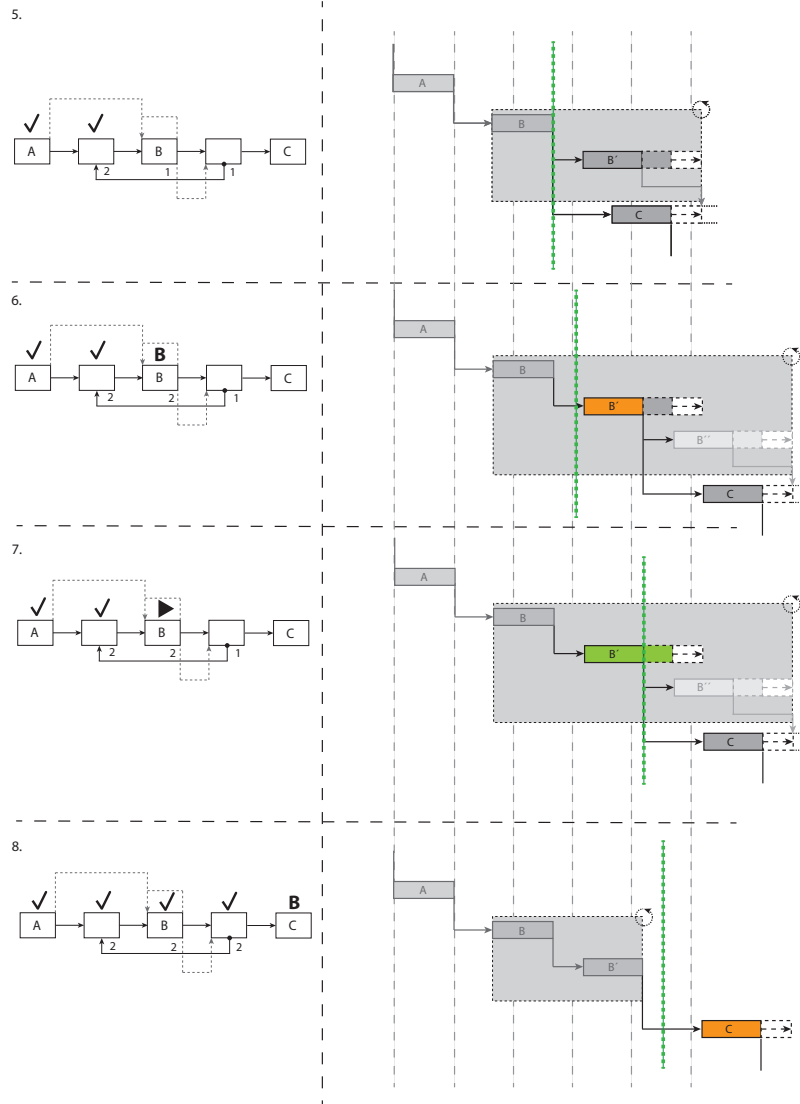


Abbildung 5.9: Ausführung einer Schleife in Min-Darstellung

6

Prototyp und Implementierung

Nach den Abbildungen der einzelnen Prozess-Elemente und der Diskussion des Verhaltens der Abbildungen auf Ausführungsebene, wird in diesem Kapitel eine Machbarkeitsanalyse vorgestellt. Diese Machbarkeitsanalyse stellt einen kleinen Prototyp in AristaFlow dar. AristaFlow ist ein Prozess-Management-System (PMS), welches auf dem ADEPT-Metamodell aufbaut. Teil dieser Arbeit, ist es aus einem Prozess, welcher in AristaFlow modelliert ist ein Gantt-Diagramm darzustellen.

6.1 Technische Ausführung

Wie in Kapitel 3 gezeigt, existieren für die einzelnen Berechnungen der Anfangs- und Endzeitpunkte verschiedene Algorithmen. In diesem Prototyp ist dabei eine Erweiterung des in

CPM verwendeten Shortest-Path-Partitioning-Algorithmus [EM92] angewendet worden. Im Sequenzdiagramm in Abbildung 6.2 sind die einzelne Methodenaufrufe und Berechnungen zu sehen. Zu Beginn werden alle Aktivitäten aus dem Prozess topologisch sortiert und in einem Array abgespeichert. Im nächsten Schritt wird für jede Aktivität ihre minimale bzw. maximale Dauer vom Prozesstemplate, welches AristaFlow anbietet, abgefragt. Zudem werden alle Zeitkanten mit ihren Abständen und Start- und Zielaktivität vom Prozesstemplate geholt. Nach diesen Abfragen auf das Template, werden in der Reihenfolge der topologischen Sortierung mit den Berechnungsvorschriften aus Kapitel 4.4.1 die Start- und Endzeitpunkte der einzelnen Aktivitäten berechnet. Nachdem die einzelnen Start- und Endzeitpunkte berechnet und die minimale und maximale Dauer abgefragt wurde, werden die einzelnen Vorgänge auf das eGantt-Diagramm abgebildet. Zuletzt werden noch Schleifen und XOR-Entscheidungen berechnet und dargestellt.



```
699 private GanttTimeNode getTime(GanttTimeNode a, Template template, GanttTimeNode[] z, StructuredConstraintEdge[] zc)
700 {
701     int[] vor = template.getPredByEdgeType(a.node.getID(), EdgeType.ET_CONTROL);
702     long vortime = 0;
703     for(int i = 0; i < vor.length;i++){
704         if((int)vortime < z[template.getNodeTopologicalID(vor[i])].kDurr + z[template.getNodeTopologicalID(vor[i])].g
705             vortime = z[template.getNodeTopologicalID(vor[i])].kDurr + z[template.getNodeTopologicalID(vor[i])].g
706             //System.out.println("TopoID" + template.getNodeTopologicalID(vor[i]));
707         }
708     }
709     //System.out.println(vortime);
710     a.getTime = vortime;
711     a.kDurr = a.minDur;
712     a.minDis = 0;
713     a.maxDis = 0;
714
715     long zminDis = 0;
716     long zmaxDis = 0;
717     long zkDurr = 0;
718
719     int[] start = zeitKantenZiele(a, zeitkanten, template);
720     if(start.length != 0){
721         for(int i = 0; i < start.length;i++){
722             GanttTimeNode ttt = GanttTimeNode(template, z[start[i]].node, a.node);
723             RelationType reltype = ttt.getRelationType();
724             long lmin = z[start[i]].minDur;
725             long lmax = z[start[i]].maxDur;
726             long lDurr = z[start[i]].kDurr;
727             long lDis = z[start[i]].minDis;
728             long lMaxDis = z[start[i]].maxDis;
729             ttt.setRelationType(reltype);
730             ttt.setMinDur(lmin);
731             ttt.setMaxDur(lmax);
732             ttt.setkDurr(lDurr);
733             ttt.setminDis(lDis);
734             ttt.setMaxDis(lMaxDis);
735             z[start[i]] = ttt;
736         }
737     }
738     return a;
739 }
```

Abbildung 6.1: GanttView.java in Eclipse

Die Umsetzung des Prototyps wurde in Eclipse realisiert. Dabei wurde in AristaFlow sehr einfach ein weiteres Fenster im Process Template Editor (der Programmteil in AristaFlow für die Modellierungsebene) hinzufügen. Dadurch ist die GanttView im Editor darstellbar. Diese GanttView ist dabei in Eclipse in der Klasse GanttView.java (siehe Abbildung 6.1) realisiert. Diese Hauptklasse realisiert die eGantt-Diagramme und die einzelnen Abbildungen.

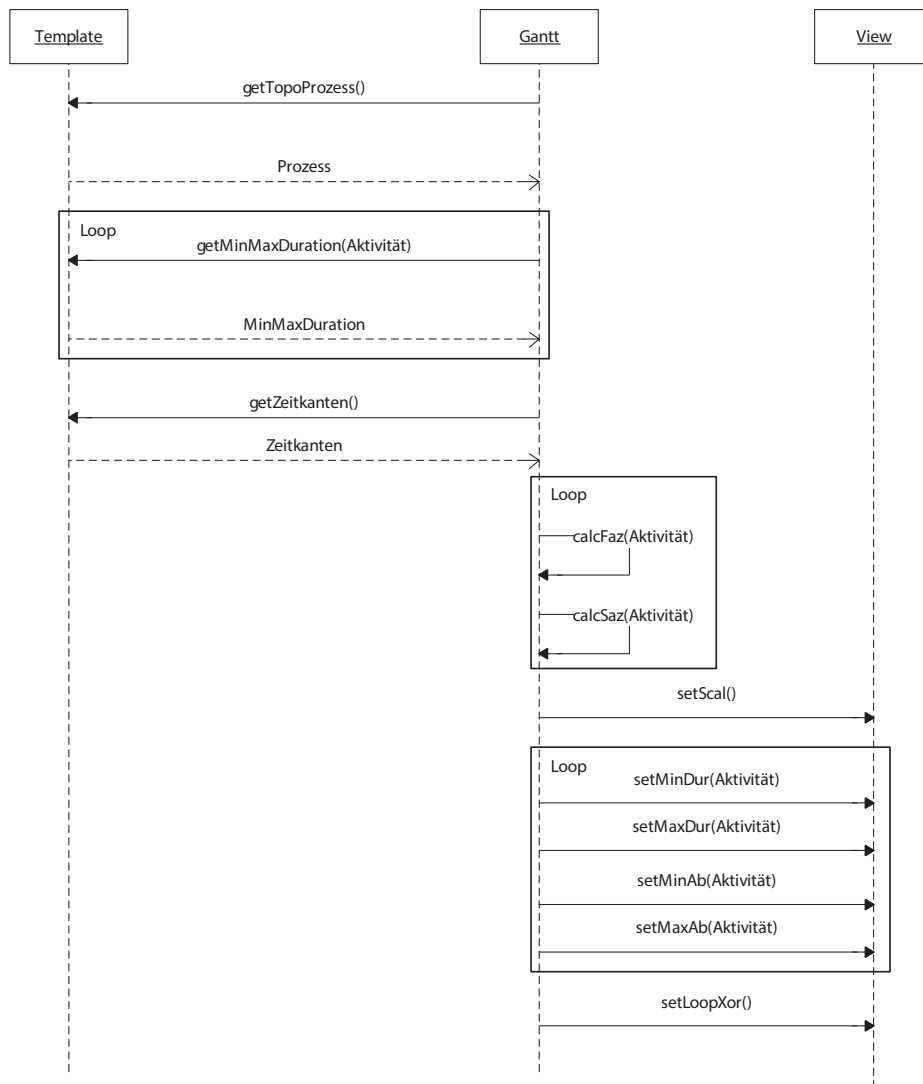


Abbildung 6.2: Vereinfachtes Sequenz-Diagramm für die eGantt-Darstellung

Im nächsten Abschnitt werden einzelne Screenshots des Prototyps vorgestellt. Der Ausgangsprozess wurde in ADEPT modelliert und in AristFlow umgesetzt. Dabei wurde versucht, soviel Zeitaspekte wie möglich zu realisieren.

6.2 Prototyp

Zu Beginn dieser Bachelorarbeit wurde eine Machbarkeitsanalyse für die Abbildung blockstrukturierter Prozesse auf eGantt-Diagramme durchgeführt. Dabei wurde nur die Modellierungsebene betrachtet.

In Screenshot 6.3 wird die komplette Ansicht des AristaFlow Process Template Editor mit der Erweiterung für die Abbildung der eGantt-Diagramme gezeigt.

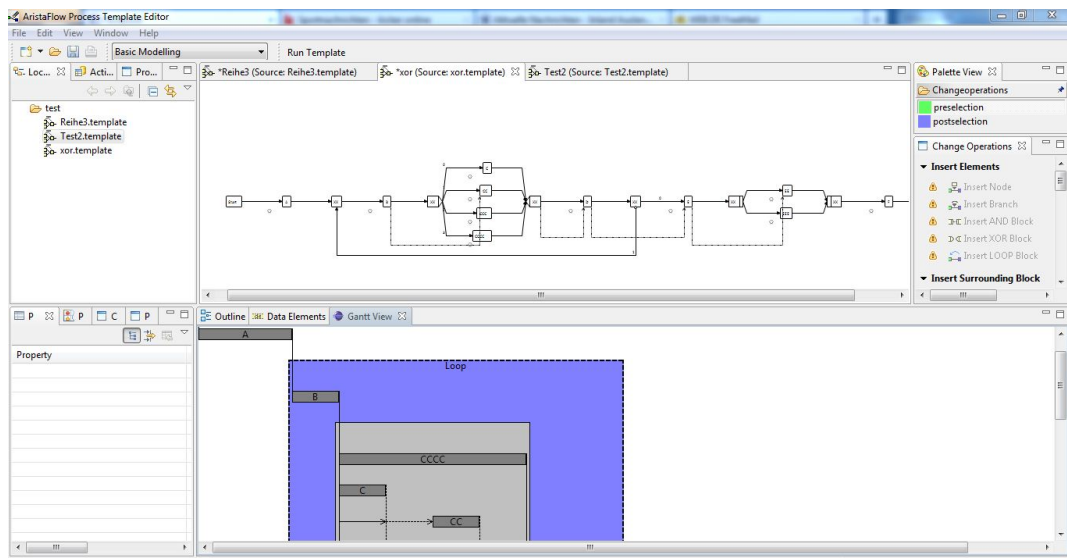


Abbildung 6.3: Ansicht des Prototyps in AristaFlow

Oben-Mittig ist der ADEPT-Prozessgraph angeordnet. Aus diesem Prozessgraph wird die Darstellung des Gantt-Diagramm abgeleitet und die einzelnen Abbildungen berechnet.

Links und Rechts des Prozessgraphen finden sich Funktionen um den Prozessgraphen zu verändern. Links sind die Repositories angeordnet, die für die Realisierung einzelner Aktivitäten verwendet werden können. Diese Repositories geben beispielsweise die Möglichkeit Aktivitäten mit einer Aktivitätenvorlage zu versehen und auf eine Datenbank zugreifen zu können. Rechts sind die einzelne Basisoperationen angebracht um den Prozess zu modellieren.

Rechts-unten ist das Gantt-Diagramm dargestellt. Es resultiert aus dem oben gezeigten

Prozessgraphen. Änderungen am Gantt-Diagramm sind bisher nicht direkt möglich. Sie müssen über den ADEPT-Prozessgraphen durchgeführt werden.

Links des Gantt-Diagramm ist das Eingabefenster für die Zeitaspekte. Bei Markierung einer Aktivität, kann die minimale und maximale Dauer festgelegt werden oder diese einen festen Start- bzw. Endzeitpunkt besitzt. Bei Markierung einer Zeitkante kann der minimale und maximale Abstand, sowie die Art der Abhängigkeit zwischen den beiden Aktivitäten die durch die Zeitkante verbunden sind eingestellt werden.

In nächsten Kapitel wird ein Fazit über die Ergebnisse dieser Arbeit gezogen und ein Ausblick auf mögliche weitere Entwicklungen gegeben.

7

Fazit und Ausblick

7.1 Fazit

In dieser Arbeit ist untersucht worden, wie blockstrukturierte Prozesse auf Gantt-Diagramme abgebildet werden können. Dazu haben wir zu Beginn einzelne Grundlagen der Metasprache ADEPT und die einzelnen Zeitconstraints kennen gelernt. Dabei stellte sich für die Abbildungen der Gantt-Diagramme Anforderungen und Eigenschaften heraus, die sie unterstützen müssen.

Nachdem die Anforderungen und Eigenschaften abgesteckt waren, wurden einzelne Repräsentationen von Zeit vorgestellt und nach ihren Eigenschaften bewertet. Diese Bewertungen und Eigenschaften zeigten, dass Gantt-Diagramme für die Abbildungen und die Übersichtlichkeit am effizientesten sind. Jedoch zeigte sich auch, dass die klassische Gantt-Diagramme nicht alle benötigten Prozess-Eigenschaften unterstützen.

Da die klassischen Gantt-Diagramme nicht alle Eigenschaften unterstützen, wurden diese in Kapitel 4 um entsprechende Konstrukte erweitert. Diese erweiterten Gantt-Diagramme wurden als *eGantt-Diagramme* bezeichnet. Diese eGantt-Diagramme unterstützen dabei alle Anforderungen und Eigenschaften eines Prozesses. Schleifen, XOR-Entscheidungen, früheste Anfangs- und Endzeitpunkte, späteste Anfangs- und Endzeitpunkte können in diesen Diagrammen dargestellt werden.

Nach den einzelnen Abbildungen auf Modellierungsebene, wurde im nächsten Abschnitt das Verhalten der eGantt-Diagramme auf Ausführungsebene betrachtet. Es sind Zustände für die einzelnen Vorgänge eingeführt und ihr Verhalten anhand von Beispielen verdeutlicht worden.

Nach den Abbildungen und deren Verhalten ist am Ende noch eine Machbarkeitsanalyse implementiert worden. Sie zeigt die Realisierbarkeit der Grundidee und die Durchführung einzelner Abbildungen auf Modellierungsebene.

Wie diese Arbeit zeigt, ist die Darstellung von Zeiteigenschaften die in blockstrukturierten Prozesse modelliert sind durch eGantt-Diagramme möglich. eGantt-Diagramme sind eine einfache Darstellung mit vielen Informationen und besitzen trotzdem noch eine hohe Übersichtlichkeit. Jedoch sind noch längst nicht alle Eigenschaften und Elemente, die in Prozessen verwendet werden in eGantt-Diagramme umgesetzt. Das zeigt, dass das Ende des Weges noch nicht erreicht ist.

7.2 Ausblick

Wie im vorigen Abschnitt erwähnt sind nicht alle Eigenschaften blockstrukturierter Prozesse in eGantt-Diagramme umgesetzt. Diese Eigenschaften, wie z.B. ereignisbasierte Aktivitäten, Subprozesse wurden bisher noch nicht behandelt. Dabei sind Subprozesse, die beispielsweise als Phasen in Gantt-Diagrammen abgebildet werden können, von großer Bedeutung. Desweiteren wäre es wünschenswert, weitere Zeitaspekte aus Tabelle 2.3 darstellen zu können.

Nach dieser vollständigen Betrachtung der einzelnen Abbildungen der eGantt-Diagramme kann man die Implementierung der Modellierungsebene und der Ausführungsebene durchgeführt werden. Für diese Implementierung ist jedoch die Komplettierung der Berechnungsvor-

schriften nötig. Durch Phasen und Ereignissen sind diese berechenbar und darstellbar. Das Verhalten der einzelnen eGantt-Diagramme wurde bei der Ausführung betrachtet. Jedoch wurde die Fehlerbehebung und Fehlererkennung vernachlässigt. Diese müsste noch betrachtet und auf die eGantt-Diagramme angewendet werden.

Neben den beiden Darstellungsarten in den eGantt-Diagrammen, ist eine dritte Darstellungsart für den Mittelwert möglich. Dieser Mittelwert müsste mit *Process Mining* und *Time Mining* bestimmt und berechnet werden.

Durch Hinzunahme einer Organisationsstruktur (z.B. alle Teilnehmer eines Hausbau) kann eine Darstellung für eine einzelne Einheit der Organisationsstruktur (z.B. Elektriker) dargestellt werden. Dabei werden einzelne eGantt-Diagramme zusammengefasst und als „Phase“ abgebildet.

Abschließend lässt sich feststellen, dass durch die eGantt-Diagramme die Zeitaspekte in blockbasierten Prozessen einfach und übersichtlich dargestellt werden können. Sie ermöglichen zudem durch ihr Verhalten bei der Ausführung eine einfache Orientierung, an welcher Position ein Prozess sich gerade befindet und helfen kritische Momente in einem Prozess schnell zu erkennen.

Abbildungsverzeichnis

2.1	Einfacher Prozess in ADEPT	6
2.2	Basiselemente des ADEPT-Metamodell	7
2.3	Pattern Kategorie [LWR10]	8
2.4	Aktivitäten- und Prozessdauer	9
2.5	Prozess mit Zeitkanten in ADEPT	10
2.6	Aktivität mit festen Termin [LWR10]	11
2.7	Weiderholungen bzw. Cycles [LWR10]	12
3.1	Gantt Diagramm aus GanttProject	16
3.2	AsbruView Aktivität [MKSJ98]	17
3.3	Methoden der Netzplantechnik [DD07]	18
3.4	CPM Graph	19
3.5	MPM Graph	19
3.6	Allgemeiner PERT-Graph	20
3.7	Gantt Diagramm in Microsoft Visio	22
3.8	Gantt Diagramm in A-Plan	22
3.9	Gantt Diagramm in Acos Plus	23
4.1	Prozess Abbildung auf klassische Gantt-Diagramme	26
4.2	Min- und Max-Darstellung und ihre Beziehung zu FAZ/FEZ bzw. SAZ/SEZ	28
4.3	Vorgang mit minimaler und maximaler Dauer	29
4.4	Vorgang mit festem Startpunkt und minimaler/maximaler Dauer	29
4.5	Sequentielle Ausführung von Vorgängen in Min-Darstellung	30
4.6	Sequentielle Ausführung von Vorgängen in Max-Darstellung	31

4.7	Sequenz mit zeitlichem Abstand in Min-Darstellung	32
4.8	Sequenz mit zeitlichem Abstand in Max-Darstellung	33
4.9	Min-Darstellung eines eGantt-Vorgang	33
4.10	Max-Darstellung eines eGantt-Vorgang	34
4.11	Sequenz mit zeitlichem Abstand in Min-Darstellung	35
4.12	Sequenz mit zeitlichem Abstand in Max-Darstellung	35
4.13	Zeitliche Abstände mit zusammengefassten Zeitkanten	36
4.14	Darstellung von Zeitkanten mit selektierten Aktivität	37
4.15	Darstellung von Zeitkanten mit selektierten Aktivität	37
4.16	Parallelität in Min-Darstellung	38
4.17	Parallelität mit Zeitkanten in Min-Darstellung	39
4.18	Zeitabstände zwischen Parallel-Verzweigungen in Min-Darstellung	40
4.19	Erweiterte Anfangs und Enzeitpunkte	41
4.20	Ausgangsproblem einer XOR-Entscheidung in Min-Darstellung	42
4.21	Berechnung des Folgevorgang in Min-Darstellung	42
4.22	XOR-Entscheidung in Min-Darstellung	43
4.23	Berechnung des Folgevorgang in Max-Darstellung	44
4.24	XOR-Block in Max-Darstellung	44
4.25	XOR-Block mit Zeitkanten in Min-Darstellung	45
4.26	XOR-Block mit Zeitkanten in Max-Darstellung	45
4.27	XOR-Block mit ausgehender Zeitkanten in Min-Darstellung	46
4.28	XOR-Block mit ausgehender Zeitkanten in Max-Darstellung	46
4.29	Schleife in Min-Darstellung	47
4.30	Schleife in Max-Darstellung	47
4.31	Komplexeres Prozessbeispiel mit mehreren Abhängigkeiten	48
5.1	ADEPT Knotenzustände mit Zeiterweiterung (in Anlehnung an [Rei00])	50
5.2	Abbildung der Knotenzustände in eGantt	51
5.3	Timeline mit bewerteten Knoten	52
5.4	Ausführung einer Sequenz in Min-Darstellung	56
5.5	Sequenz mit zeitlichen Abständen in Min-Darstellung	57
5.6	Sequenz mit zeitlichen Abständen in Max-Darstellung	58

5.7	Ausführung eines XOR-Blockes in Min-Darstellung	59
5.8	Ausführung einer Schleife in Min-Darstellung	60
5.9	Ausführung einer Schleife in Min-Darstellung	61
6.1	GanttView.java in Eclipse	64
6.2	Vereinfachtes Sequenz-Diagramm für die eGantt-Darstellung	65
6.3	Ansicht des Prototyps in AristaFlow	66

Literaturverzeichnis

- [BHM⁺06] BHATTACHARYA, Arindam ; HEMERLINGA, Jim ; MICHAEL, David ; NICOL, Ron ; TANG, Tjun ; WADDELL, Kevin ; WONG, John: *Organizing for Global Advantage in China India and Other Rapidly Developing Economies*. The Boston Consulting Group, 2006
- [DD07] DOMSCHKE, Wolfgang ; DREXL, Andreas: *Einführung in Operations Research*. Springer, Berlin, 2007. – ISBN 3540709487
- [EM92] EVANS, James R. ; MINIEKA, Edward: *Optimization algorithms for networks and graphs* (2. ed.), 1992
- [JB96] JABLONSKI, Stefan ; BUSSLER, Christoph: *Workflow Management: Modeling Concepts, Architecture and Implementation*. International Thomson Computer Press, 1996. – ISBN 1850322228
- [LWR10] LANZ, Andreas ; WEBER, Barbara ; REICHERT, Manfred: Workflow Time Patterns for Process-aware Information Systems. In: *Proceedings of the 11th International Workshop, BPMDS 2010, and 15th International Conference, EMMSAD 2010* Bd. 50. Hammamet, Tunesia : Springer Berlin / Heidelberg, June 2010 (Lecture Notes in Business Information Processing), S. 94–107
- [MKSJ98] MIKSCH, Silvia ; KOSARA, Robert ; SHAHAR, Yuval ; JHONSON, Peter: AsbruView: Visualization of Time-Oriented, Skeletal Plans. In: SIMMONS, R (Hrsg.) ; VELOSO, M (Hrsg.) ; SMITH, S (Hrsg.): *The Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98)*, AAAI Press, 1998, S. 11–18
- [NM02] NEUMANN, K. ; MORLOCK, M.: *Operations Research*. Hanser Verlag, 2002. – ISBN 9783446221406

Literaturverzeichnis

- [Rei00] REICHERT, Manfred: Dynamische Ablaufänderungen in Workflow-Management-Systemen, University Ulm, 2000
- [Rin98] RINZA, P.: *Projektmanagement: Planung, Überwachung und Steuerung von technischen und nichttechnischen Vorhaben*. Springer, 1998. – ISBN 9783540640219
- [vv02] VAN DER AALST, Wil ; VAN HEE, Kees: *Workflow Management: Models, Methods, and Systems*. The MIT Press, 2002. – ISBN 0262011891

Name: Martin Sommer

Matrikelnummer: 643177

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Martin Sommer