Vorlesung „Software Architekturmodelle“

Konsistenz III: Quantitative Analyse

15.2.2002

Dr. Harald Störrle

Organisatorisches

- Gute Nachrichten: es gibt einen Schein - allerdings wird er die üblichen Scheine (4+2 SWS) nicht ersetzen können, da diese VL nur 2SWS umfaßt.
- Daher: bitte in Teilnehmerliste eintragen, auch rückwirkend/stellvertretend!
- Außerdem: bitte die Feedbackbögen an mich zurück, so dies nicht geschehen ist.
- Wer weitere Kommentare hat, kann diese
  - direkt bei mir loswerden, oder
  - per Mail (stoerrle@informati.uni-muenchen.de), oder
  - für die Feedback-Runde am 5.2. aufheben.

Aufgaben von vor Weihnachten

Ich weiß es nicht mehr - hab nachgeschaut, und nix gefunden. Falls doch jemand was gemacht hat - nur zu!

Wiederaufnahme des Thema

- Es ging um Konsistenz von Modellen, also Mengen aufeinander bezogener Diagramme (und Formulare).
- Es ging um die Formalisierung, und damit Operationalisierung von intuitiven Konzepten und Vorstellungen.
- Bislang ging es nur um qualitative Fragen, also ist/ist nicht konsistent, erreicht/erreicht nicht einen Endzustand usw.
- Heute geht es um quantitative Aussagen, also „ist die Kapazität größer als die mittlere Füllung“, oder „reicht der Durchsatz für das Verkehrsaufkommen aus“ usw.
- Allerdings ist die Vorlesung heute nur „Proof of Concept“!

Ziel heute

- Das Ziel heute ist die quantitative Analysen des Bibliotheksmodells (in Ausschnitten).
- Quantitativ kann hierbei mindestens zweierlei bedeuten:
  - Lastanalyse: Ereignisdauern, Prozessorleistungen, Benutzungsprofile, ...
  - Verkehrsanalyse: Ereignishäufigkeiten, Verteilungen, Erwartungswerte, ...
- Wir werden beide Aspekte behandeln (so die Zeit reicht), aber nur auf dem Papier: Werkzeuge gibt es leider nur sehr wenige.

Last und Leistung: Was ist zu tun?

Für alle quantitativen Untersuchungen sind jeweils drei Fragen zu klären:

- Wie erhalten wir die Leistungs-Daten für die Teilsysteme?
- Wie bringen wir sie in das Modell?
- Wie können wir das Modell unter dem Last-Gesichtspunkt auswerten?

Diese Fragen werden uns heute immer wieder begegnen!

Last und Leistung: Motivation

Eine realisitische Ausgangsfrage ist die Frage nach der Antwortzeit: Für eine gegebene Anfrage des Benutzers (und definierte Randbedingungen), wie lange dauert es, bis das System die gewünschte Leistung erbracht hat?

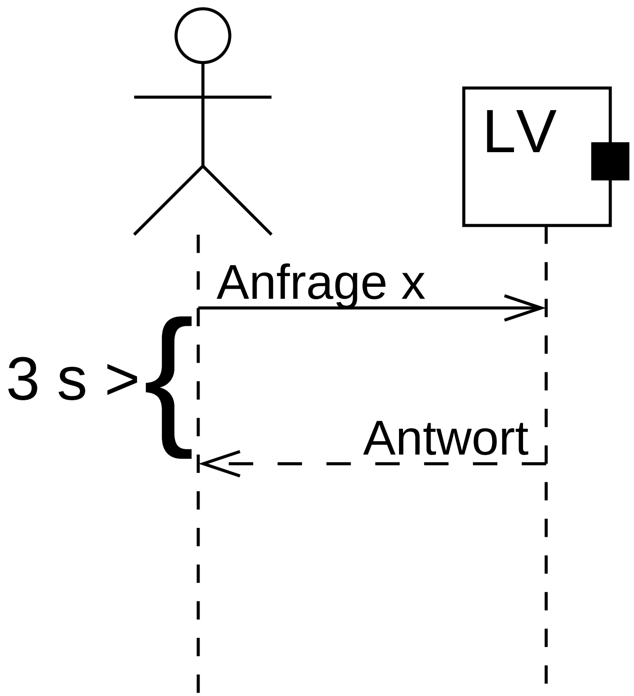
Beispiel E-Business: beim Online-Banking u.ä. bedeutet Antwortzeit die Zeit von der Eingabe bis zum abgeschlossenen Seitenaufbau. Der durchschnittliche Surfer akzeptiert nur Antwortzeiten von wenigen Sekunden.

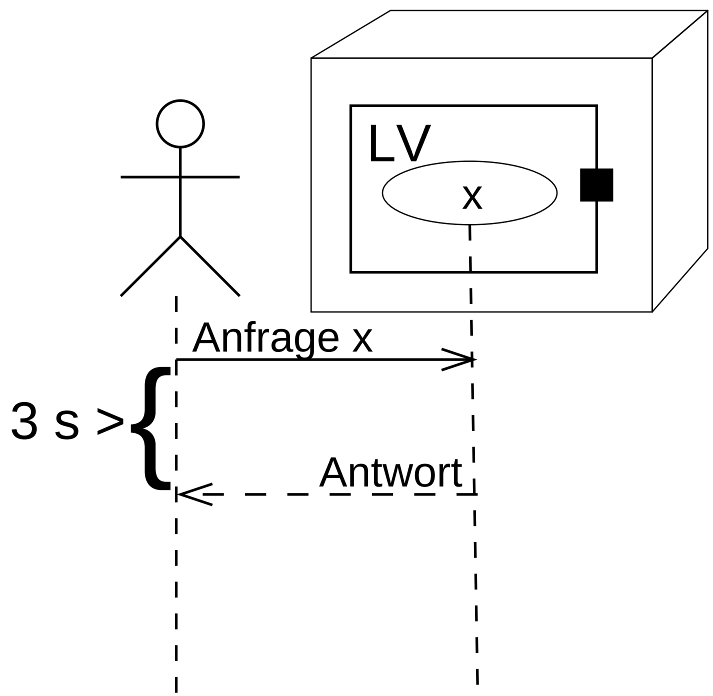
Das hat enorme architektonische Auswirkungen: Fat-Client-Anwendungen für Endkunden (also die C/S-Überbleibsel) sind bei üblichen Übertragungsraten von 56-128 Kbit/s nicht akzeptanzfähig.

Selbst in /schnellen) Unternehmensnetzen sind Antwortzeiten wichtig, da sie unmittelbar auf die Produktivität (d.h. Kosten & Wettbewerbfähigkeit) wirken.

Modellierung von quantitativen Aspekten

Nehmen wir also an, im Bibliothekssystem sei eine Anforderung, daß eine Anfrage x binnen 3 Sekunden beantwortet sein soll. Im Diagramm kann man das ausdrücken wie folgt:





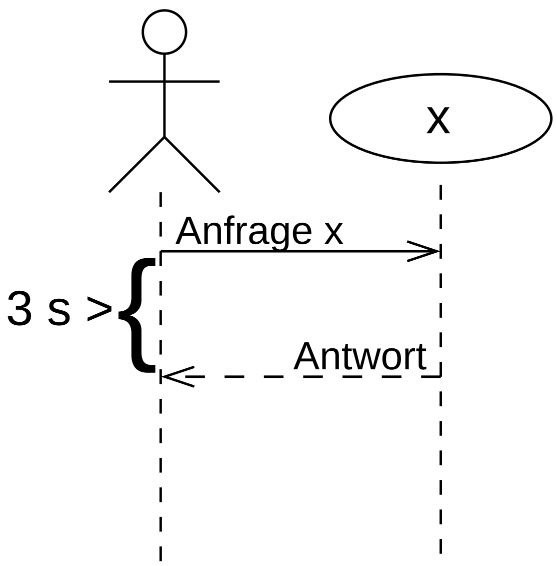
Last und Leistung: Formalisierung 3 - Alternativen

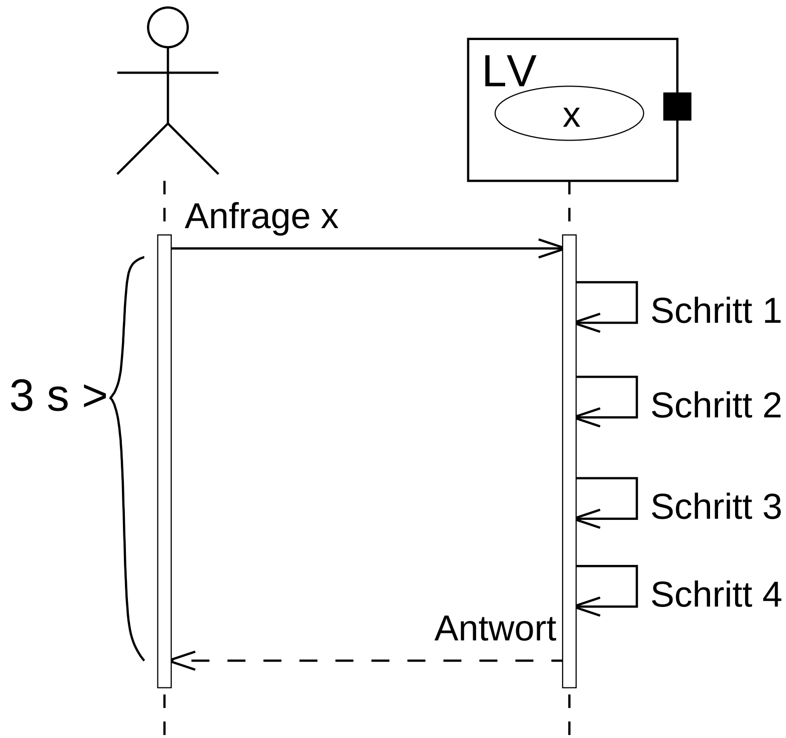
Wir haben jetzt verschiedene Möglichkeiten fortzufahren:

- **A:** Aus dem bisherigen Entwurf ist die strukturelle Verfeinerung des Subsystems LV bekannt (insbes. Protokollrolle des Ports, der x entgegennimmt)
- **B:** Wir kennen die Nutzfall-Szenarien für x, dargestellt als Interaktions- oder Aktivitätendiagramm
- **C:** Wir kennen die Verbindung zwischen Struktureller und funktionaler Verfeinerung, dargestellt als Nutzfallkarten

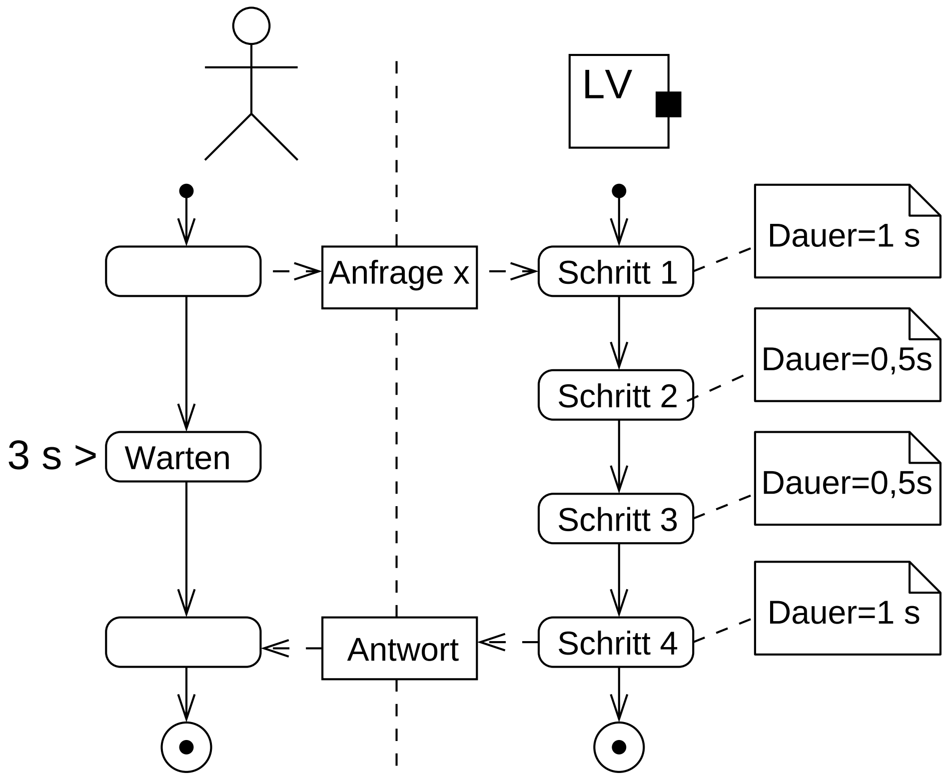
In allen Fällen liegen Verhaltensbeschreibungen vor. Welche davon in welcher Form man wählt, unterliegt pragmatischen Kriterien. Schauen wir uns drei Alternativen an.

Alternative A: Interaktionsdiagramm

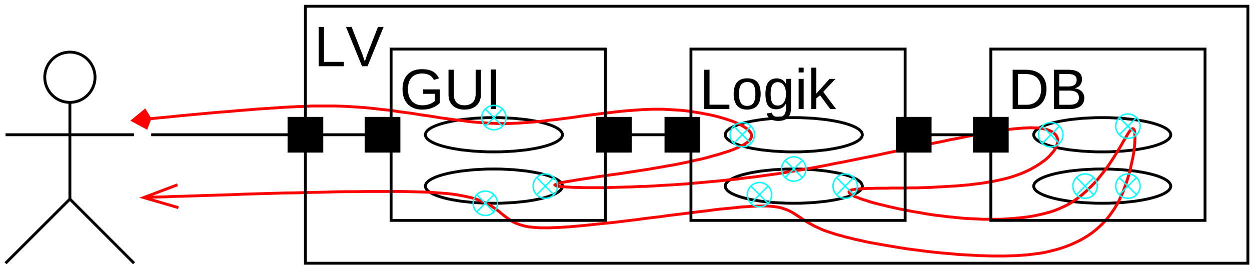




Alternative B: Aktivitätsdiagramm



Alternative C: Nutzfalkarte



Frage 1: Datenerhebung

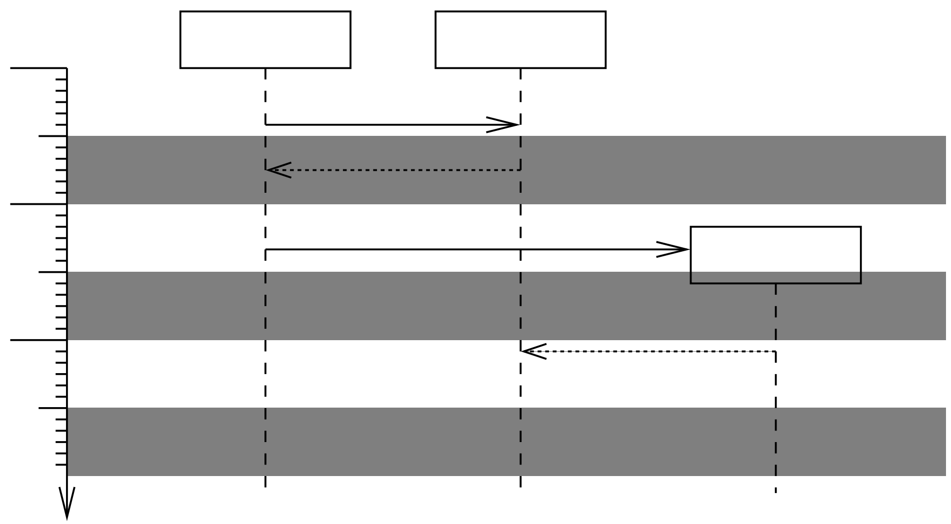
Zunächst zur ersten Frage. Es bieten sich drei Wege an, je nach Art des betreffenden Subsystems:

- ich weiß die Werte (z.B. vom Hersteller oder von Benchmarks, aber für welche Bauteile habe ich überhaupt Zahlen, wie zuverlässig/relevant sind sie?);
- ich messe sie am lebenden Objekt (Vorgehen und Instrumentierung ganz analog zu Test, aber wann kann ich so vorgehen, mit welchem Aufwand?);
- oder ich nehme mir meinen Entwurf vor, und fange an zu denken.

Fazit: zur heutigen Zeit kann man nur messen (Commodities, Legacies) und denken (Neuentwicklungen).

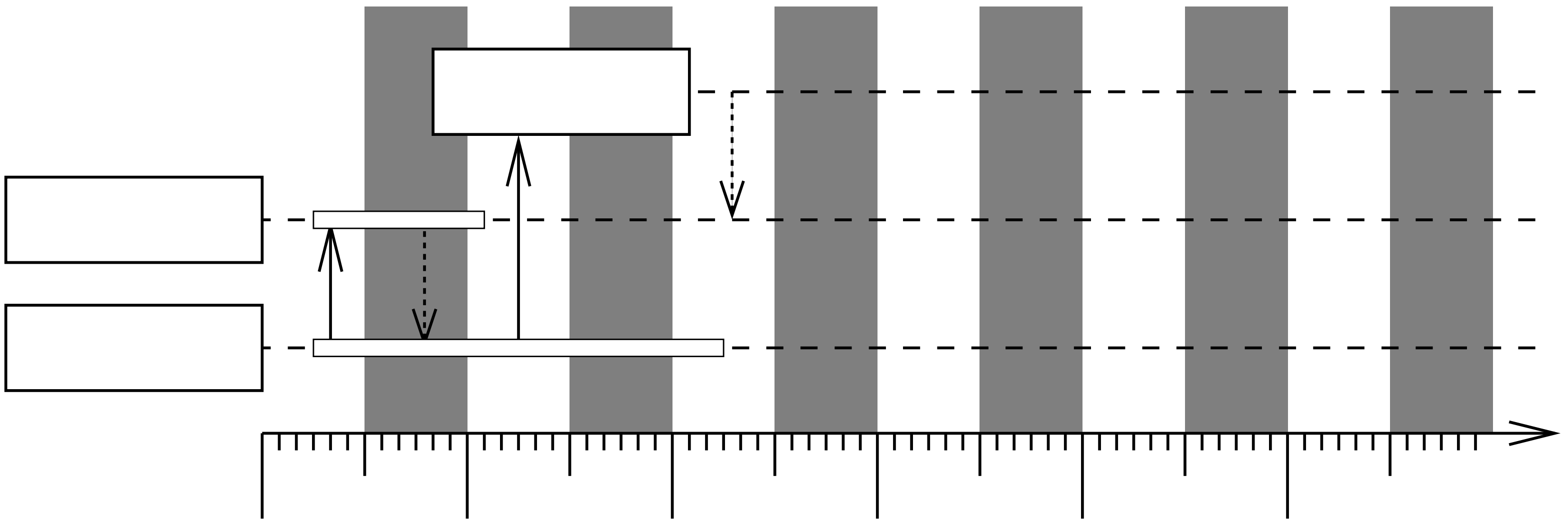
Metrische Interaktionsdiagramme

Metrische Interaktionsdiagramme werden im UML Standard nur am Rande erwähnt.



Analoges Vorgehen bei Aktivitätsdiagrammen möglich, aber offensichtlich nicht bei Kollaborationsdiagrammen und Nutfallkarten.

Gekippte (metrische) Interaktionsdiagramme



Kausalnetz

Ein Petrinetz $\langle P, T; F, \Sigma, \alpha\rangle$ heißt **Kausalnetz**, gdw.

es ist nicht Platz-verzweigend, i.e. $\forall_{p \in P} : \; p^\bullet \leq 1 \wedge {}^\bullet p \leq 1$,

F ist wohlfundiert und nicht zirkulär: $F^+ \cap (F^-)^+ = \emptyset$;

und endlich verzweigend, i.e. $\forall p \in P : \{t \in T \mid t \in p^\bullet\}$ ist endlich.

Verbindung zum Netz wird hergestellt durch eine Funktion r von den Netzelementen des Ablaufs auf die Netzelemente des Ursprungsnetzes.

Zur Erinnerung: α ist eine Etikettierungsfunktion.

Prozeß

Ein Ablauf eines Petrinetzes N heißt **Prozeß**, und kann wieder durch ein Kausalnetz R (für „Run“) dargestellt werden, welches zum Ursprungsnetz durch eine Funktion $r : X_R \to X_N$ in Verbindung gesetzt wird.

Die X sind jeweils die Netzelementen der beiden Netze.

Die Anfangs- und Endmarkierungen sind ebenfalls durch r aufeinander bezogen.

Beispiel: LV-Rolle

Partielle Worte

Die Prozeß-Semantik ist etwas unhandlich, denn sie enthält noch die Stellen, die in Prozessen wenig aussagen, und uns daher eigentlich gar nicht interessieren.

Eine einfachere Semantik erhält man, wenn man intuitiv folgendes Verfahren anwendet:

- Ersetze jede Transition durch ihr Etikett
- Entferne alle Stellen
- Verbinde stattdessen die Etiketten direkt

Die so entstandene Struktur heißt Partielles Wort. Partiell wg. der erhalten gebliebebeb Partiellen Ordnung des Prozesses, und Wort wegen der Ersetzung der Transitionenen durch Etiketten.

Partielle Worte

Ein **Partielles Wort** ist ein Tripel $\langle Q, \leq, \alpha \rangle$ mit

- $\leq$ eine partielle Ordnung auf Q ist (reflexiv, transitiv, antisymmetrisch);
- $\alpha : Q \to \Sigma$

Endliches PW: Q ist endlich (und damit die Relation $\leq$).
Impliziert: exisitiert Anfangskonfiguration.

Leistung

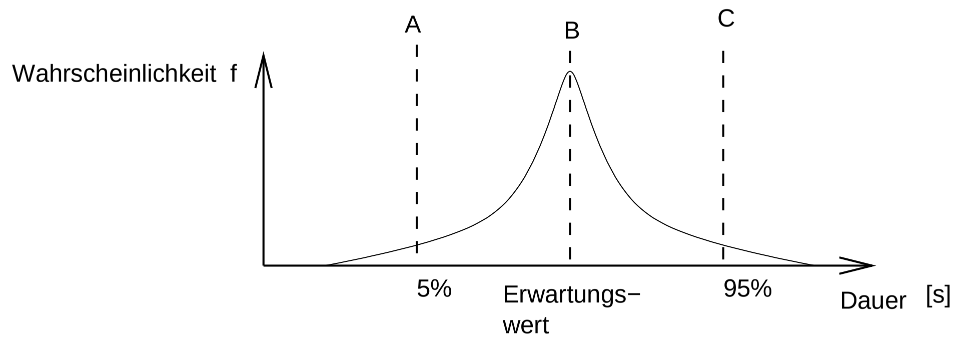
Mein Petrinetz-Modell soll ein Leistungsmodell sein, also gebe den Transitionen eine Dauer. In der Regel interessieren mich drei zeitliche Werte: Minimal- und Maximalwerte („dauert mindestens“ und „dauert längstens“), sowie der Normalfall (d.h. der Durchnitts- oder Erwartungswert).

Es gibt verschiedene Möglichkeiten:

- Gebe jeder Transition eine Dauer d (den Normalfall, ggf. 0).
- Zusätzlich könnte eine Schwankung ϵ spezifiziert (Transition dauert $d + / - \epsilon$)

Leistung

Allgemeiner ist jedoch folgende Lösung: Transition hat Verteilung f der Dauer (d.h. für jede Dauer gibt es eine Wsk.) Der Erwartungswert entspricht Normalfall, Minimum und Maximum wären die Dauern, in der z.B. höchstens 5%, bzw. mindestens 95% der Ereignisse (d.h. Transitionsabarbeitungsdauern) liegen.



$$A : \int_0^A f = 0,05$$

$$B : \int_0^B f = 0,95$$

Dies kann ich formal als Etikettierung des Netzes darstellen:
$\alpha : T \to \mathbb{R}^+$.

Dauer eines Ablaufes: Formale Definition

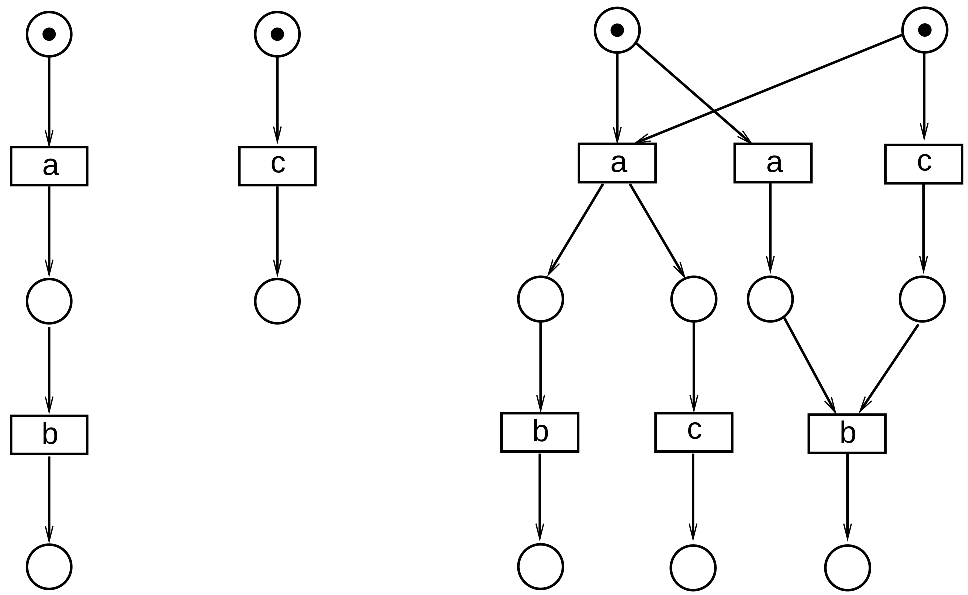
Jetzt kann ich zunächst die Dauer eines Ablaufes berechnen (lassen), indem ich ein PW als einen Graph betrachte.

Ich summiere für jeden terminalen Pfae in diesem Graph die Werte an den Knoten, und ermittle von diesesn Werten Maximum, Minimum, und Mittel.

Wenn ich dies für alle Abläufe ausführe, kann ich dann auch die Dauer eines Netzes als Maximum, Minimum, und Mittel aller seiner Abläufe berechnen.

Nebenläufigkeits vs. Verschränkungssemantik

Eine rein sequentielle Semantik ist klarerweise immer die langsamste. Aber eine Verschränkungs-Semantik ist genauso langsam, denn ihre Abläufe sind immer sequentiell! Und selbst mit Schritt-Verschränkung ist eine Nebenläufigkeitssemantik nie langsamer, aber oft schneller:



Nebenläufigkeits vs. Verschränkungssemantik

Diese Netze haben:

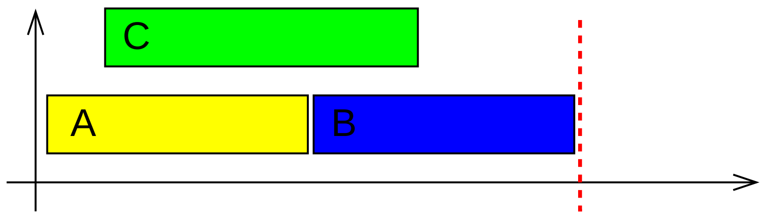
- die gleiche Sprache: $\{\mathsf{a.b.c}, \mathsf{a.c.b}, \mathsf{c.a.b}\}$
- verschiedene (aber beobachtungsäquivalente) ETS,
- aber unterschiedliche PWs:

$$\left\{ \left(\begin{array}{c} \mathsf{a} \rightarrow \mathsf{b} \\ \\ \mathsf{c} \end{array} \right) \right\} \qquad \left\{ \left(\begin{array}{ccc} \mathsf{a} & \rightarrow & \mathsf{b} \\ & \searrow & \\ & & \mathsf{c} \end{array} \right), \left(\begin{array}{ccc} \mathsf{a} & \rightarrow & \mathsf{b} \\ & \nearrow & \\ \mathsf{c} & & \end{array} \right) \right\}$$

Dieser Unterschied hat erhebliche praktische Auswirkungen. Das linke Modell ist realistischer als das rechte, und die Leistungsbewertungen unterscheiden sich. Um dies zu sehen, betrachten wir die Abläufe als zeitliche Intervalle.

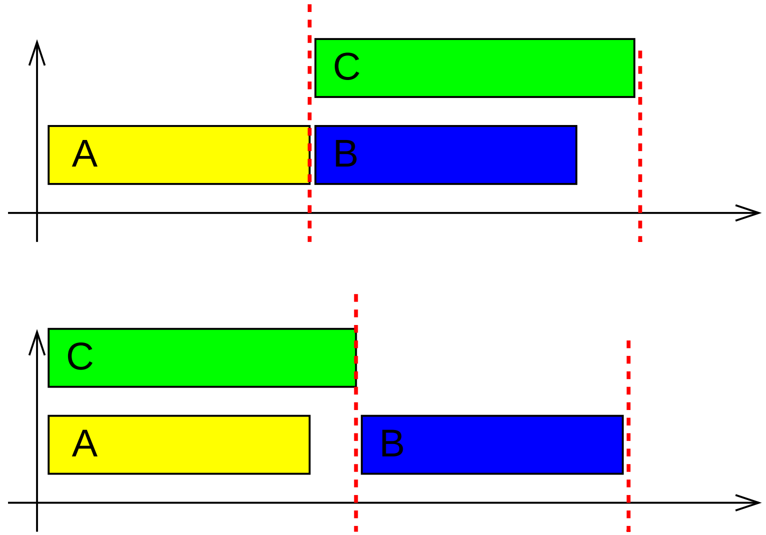
Nebenläufigkeits vs. Verschränkungssemantik

Variante (1)



Dauer: max(C,A+B)

Variante (2)

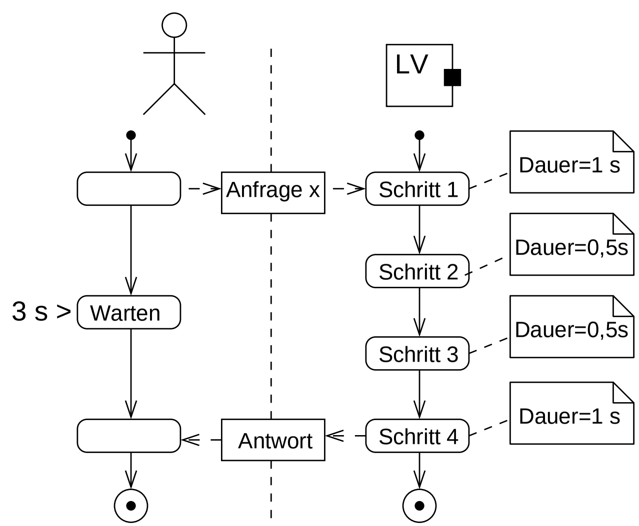


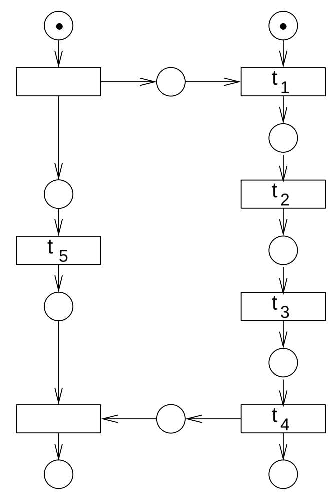
Dauer: max(A+max(B,C) , max(A,C)+B)

Wenn C> min(A,B) ist Variante (1) schneller als Variante (2),
ansonsten sind beide gleich schnell.

Zurück zum Beispiel

Dies waren die Anforderung und das daraus resultierende Petrinetz:





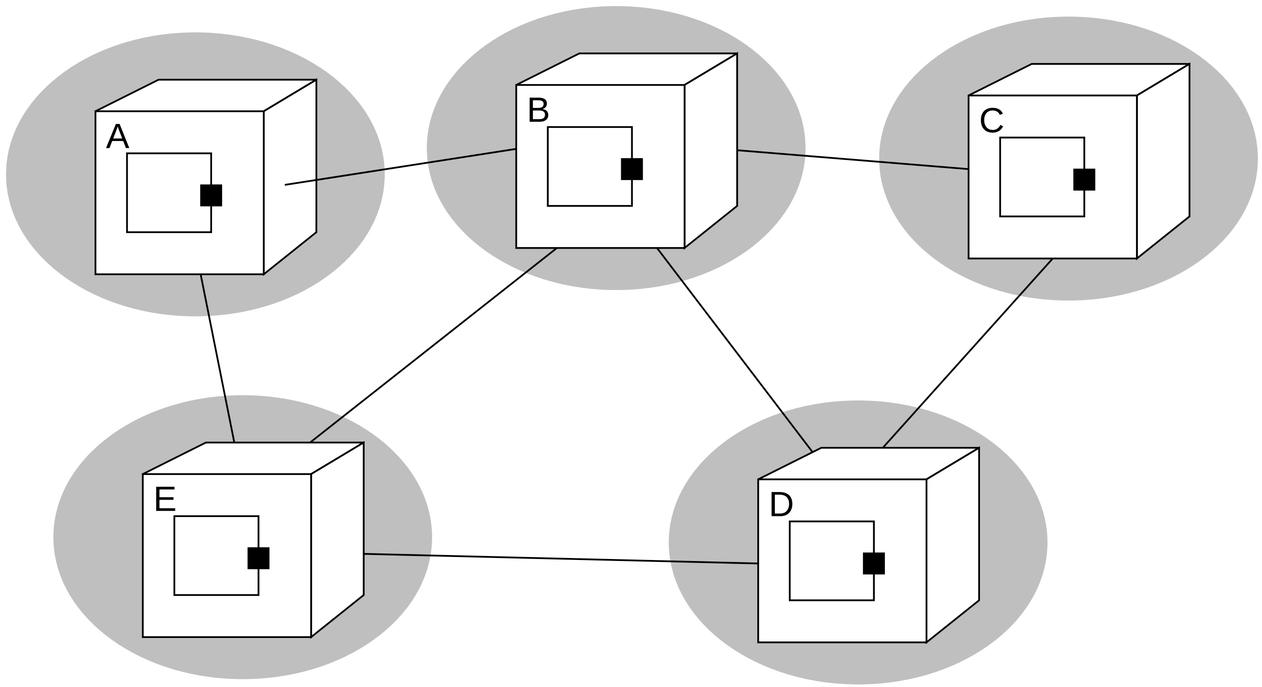
Etikettierung: $\{t_1 \mapsto 1, t_2 \mapsto 0,5, t_3 \mapsto 0,5, t_4 \mapsto 1\}$, die anderen Transitionen werden mit 0 etikettiert. Man könnte z.B. t_5 auch it dem Maximalwert 3 etikettieren, und fragen, ob das Netz konsistent ist.

Verkehrsanalyse

Ziel der Leistungsanalyse war die Antwortzeit – in die obige Rechnung sind aber bislang nur Rechenzeiten eingegangen, nicht jedoch Wartezeiten, z.B. aufgrund von Verdrängung durch nebenläufige Prozesse, verteilungsbedingte Wartezeiten und Verwaltungsaufwand (z.B. des Betriebssytems).

Was die Wartezeiten angeht, so ist festzustellen, daß es Nutzfälle gibt, die nur im Verbund mit anderen Netzknoten bearbeitet werden können. Um den Netzverkehr und daraus entstehende Wartezeiten zu berechnen, müssen wir zunächst das Netz modellieren.

Verkehrsanalyse



Hier nehmen wir an, daß es in unserem Bibliotheksverbund V fünf Filialen gibt, die mit A bis E bezeichnet werden ($V = \{A, B, C, D, E\}$).

Modellierung von Kanälen

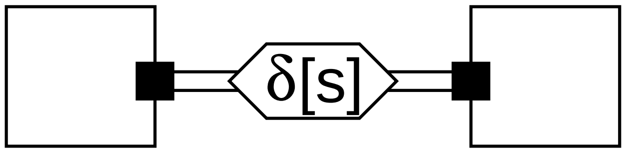
Offenbar spielt die Charakteristik der Verbindungskanäle eine große Rolle für das Leistungsverhalten des Netzes. Daher kümmern wir uns jetzt zunächst um Arten von Kanälen und deren Eigenschaften.

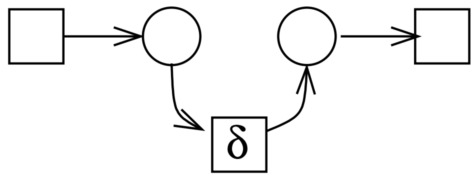
Der Ausgangspunkt ist die synchrone Kommunikation, also kein eigentlicher Kanal.

Der (von der Semantik her) einfachste Kanal modelliert eine asynchrone Kommunikation mit einem Kanal als Stelle.

Kanal mit Verzögerung

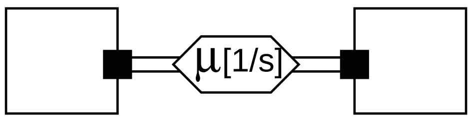
Nachrichten haben oftmals eine Laufzeit, d.h. die Übertragung als solche hat eine Dauer.

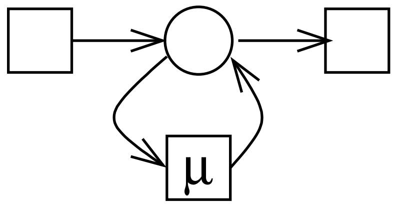


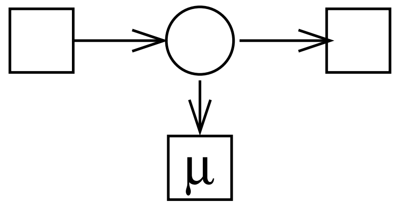


Kanal mit Fehlern

Z.B. Umordnung, Verfälschung, Verlust





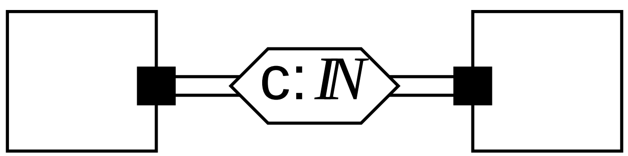


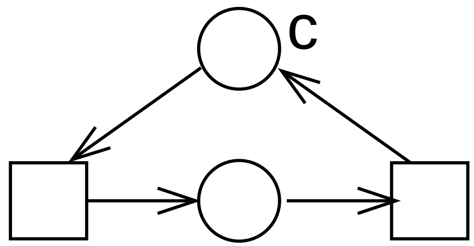
Dieses Modell zwar bei weitem zu detailliert für unsere Zwecke, aber so könnte man es machen.

Bedienstrategie: Tafel

Kanal mit Kapazität

Pufferkapazität (Zahl der Nachrichten, die abgeschickt wurden, aber noch nicht angekommen sind).





Übertragungskapazität (maximale Zahl von übertragbaren Nachrichten pro Zeiteinheit) ist abhängig von der Abgangsrate der Wartesysteme, d.h. der Empfangsrate (vgl. Handshake-Synchronisation).

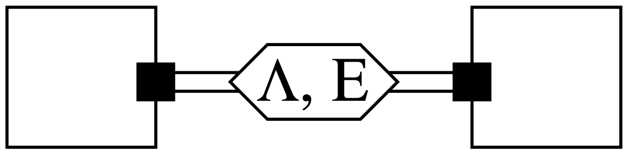
Kanal mit Verkehrsprofil

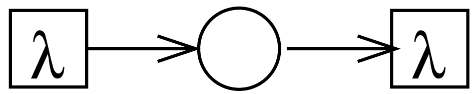
Wir betrachten nun einen Kanal als elementares Wartesystem, d.h. für Zu- und Abgang gibt es Ereignisraten. Sinnvollerweise wird zugrundegelegt, daß das System sich im Flußgleichgewicht befindet (gilt z.B. für das LV-System). Weiter nehmen wir an, daß m Protokollrollen beteiligt sind, und daß es gibt n verschiedene Nachrichtentypen gibt (Extension des Nachrichtentyps ist n).

Wir modellieren also das Verkehrsprofil für einen Konnektor mit einer Matrix Λ wie folgt:

$$\Lambda = \begin{pmatrix} \lambda_{r_1,a_1} & \cdots & \lambda_{r_1,a_n} \\ \vdots & & \vdots \\ \lambda_{r_m,a_1} & \cdots & \lambda_{r_m,a_n} \end{pmatrix}$$

Das Netzmodell hat die Form:





Solche Systeme sind bereits früher, unter dem Namen Warteschlangen-Theorie untersucht worden (kommt gleich noch im Detail).

Der Einfachheit halber beschränken wir uns auf binäre Konnektoren, und nehmen an, daß der Nutzinhalt der Nachrichtenpakete uninteressant ist, d.h. es gibt eine einzige Äquivalenzklasse. Also ist

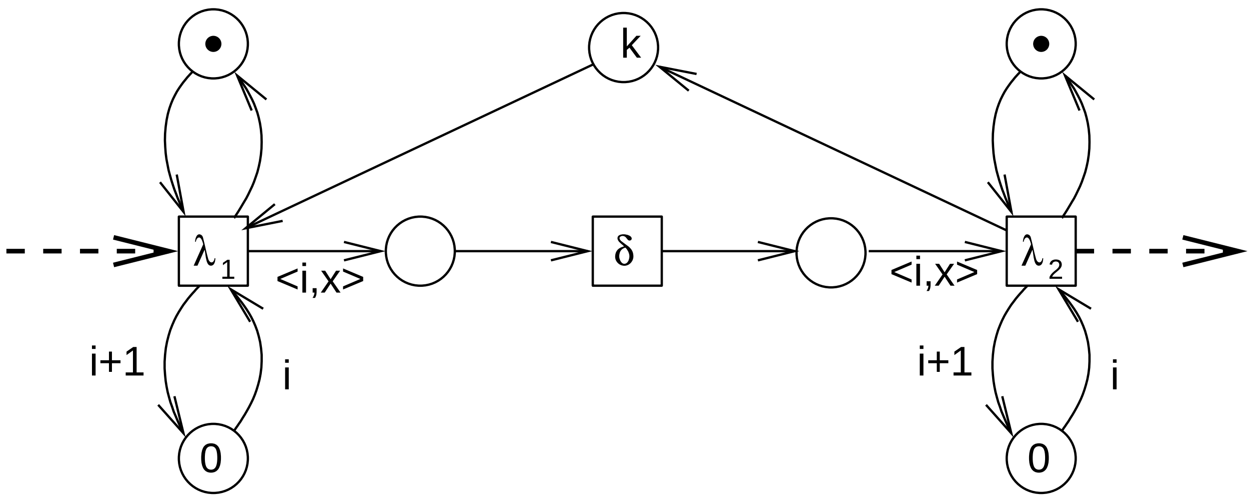
$$\Lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}.$$

Kombination

Bislang haben wir die verschiedenen Eigenschaften jeweils für sich modelliert. In der Regel treten sie aber in Kombination auf.

Sind die verschiedenen Eigenschaften, die wir oben diskutiert haben, in diesem Sinne miteinander kombinierbar? Betrachten wir zunächst den einfacheren Fall eines Simplex-Kanals, d.h. der Übertragung in nur eine Richtung. Die verschiedenen Eigenschaften kombinieren sich wie folgt.

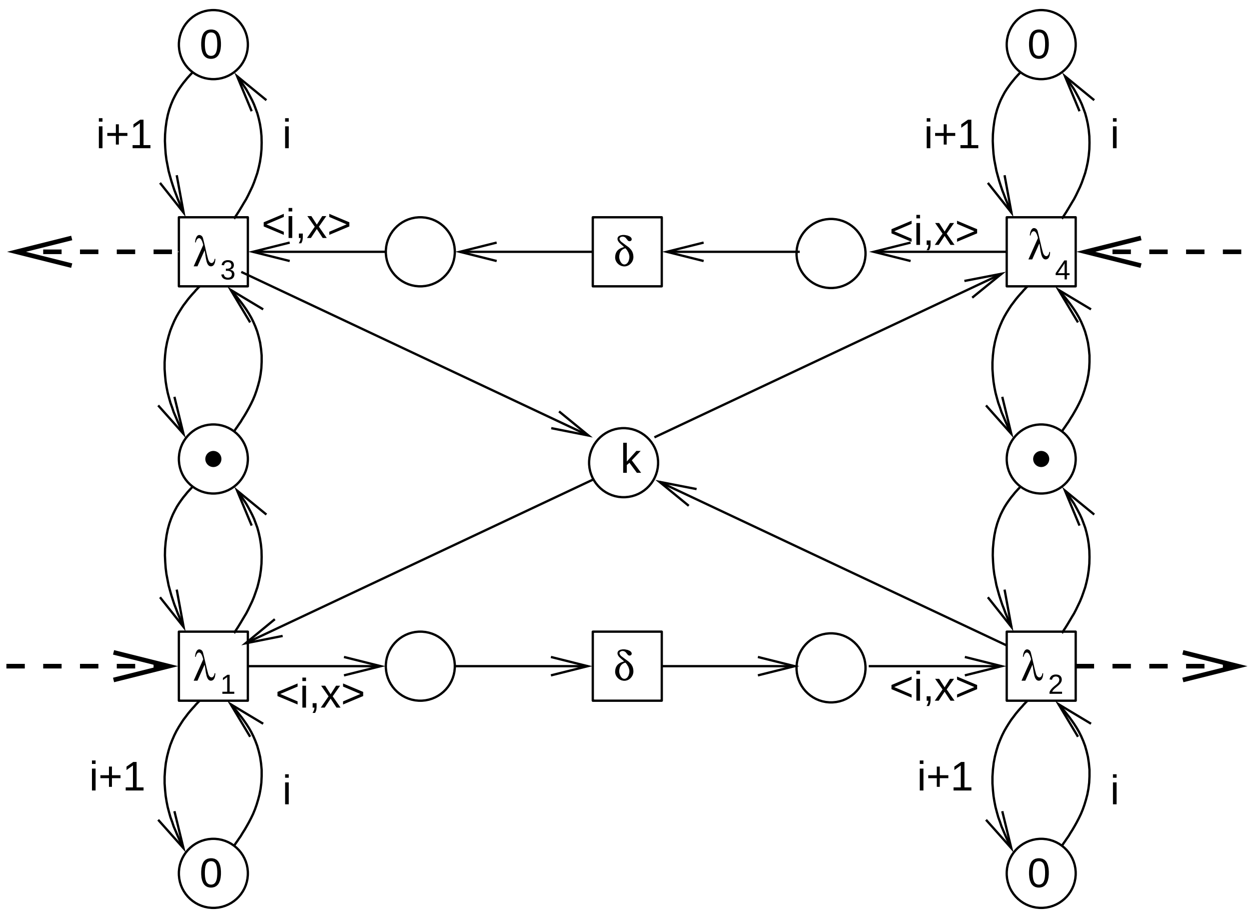
Abschluß: Simplex-Kanal



Dieses Netz stellt einen Simplex-Kanal mit Verzögerung δ, FIFO-Bedienstrategie, einer Kapazität k, sowie Zu- und Abgangsraten λ_1 und λ_2 dar. Der Kanal arbeitet fehlerfrei. Zu jedem Zeitpunkt kann jede Protokollrolle nur genau eine Nachricht senden bzw. empfangen.

Abschluß: Duplex-Kanal

Um nun einen Duplex-Kanal mit den entsprechenden Werten zu modellieren, werden einfach die entsprechenden Bestandteile analog hunzugefügt. Wir erhalten folgendes Netz.


0
i+1
i
<i,x>
λ3
δ
λ4
k
λ1
λ2
0

Grundlagen: Warteschlangentheorie

Betrachten wir ein System mit einer abzählbar unendlichen Menge $\mathcal{S}$ von Zuständen. Genau in einem davon ist das System zu jeder Zeit. Welcher es ist, wird durch eine Zufallsvariable $S(t), t \in I\!R^+$ angegeben, und man sagt, „$S(t)$ bildet einen zustandsdiskreten stochastischen Prozeß“.

Die Wsk. des Überganges von Zustand s_i nach Zustand s_j im Intervall $[t, t + \delta t]$ wird aufgeschrieben als $P_{s_i s_j}(t + \delta t)$. Sie beträgt $P[S(t + \delta t) = j | S(t) = i]$ (Bayes!).

Wenn $P_{s_i s_j}(t + \delta t)$ unabhängig von der Vorgeschichte ist, d.h. wenn gilt für alle a gilt

$$P[S(t+\delta t) = s_j | S(t) = s_i \wedge S(t-a) = s_k] =$$
$$P[S(t+\delta t) = s_j | S(t) = s_i]$$

heißt der zugrundeliegende Prozeß (zustandsdiskreter) Markov-Prozeß (oder: Markov-Kette).

Wenn zusätzlich gilt

$$P[S(t+\delta t) = s_j | S(t) = s_i] = P_{ij}(\delta t)$$

heißt die Markov-Kette homogen.

Nur solche betrachten wir im Folgenden! Für alle anderen drohe heftige Differentialgleichungen.

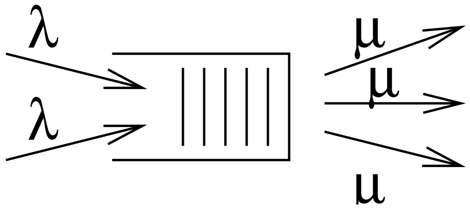
Eine Markov-Kette ist ein ETS mit Übergangswahrscheinlichkeiten an den Kanten.

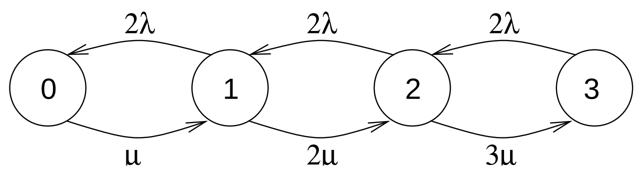
Beispiel

Annahmen:

- homogene Markov-Kette,
- Puffer mit Kapazität 3,
- jeweils 2 bzw. 3. zeitunabhängige Zugangs- und Abgangsprozesse mit den Raten λ und μ.
- Flußgleichgewicht (d.h. $3\mu \geq 2\lambda$)

Wie groß ist die mittlere Füllung des Puffers?





Neue Ereignisse können verloren gehen, wenn der Puffer voll ist

Beispiel: Übergangswahrscheinlichkeiten

Beachte: es wird ein stationärer, also zeitunabhängiger Pozeß betrachtet, also gilt $p_i(t) = p_i$.

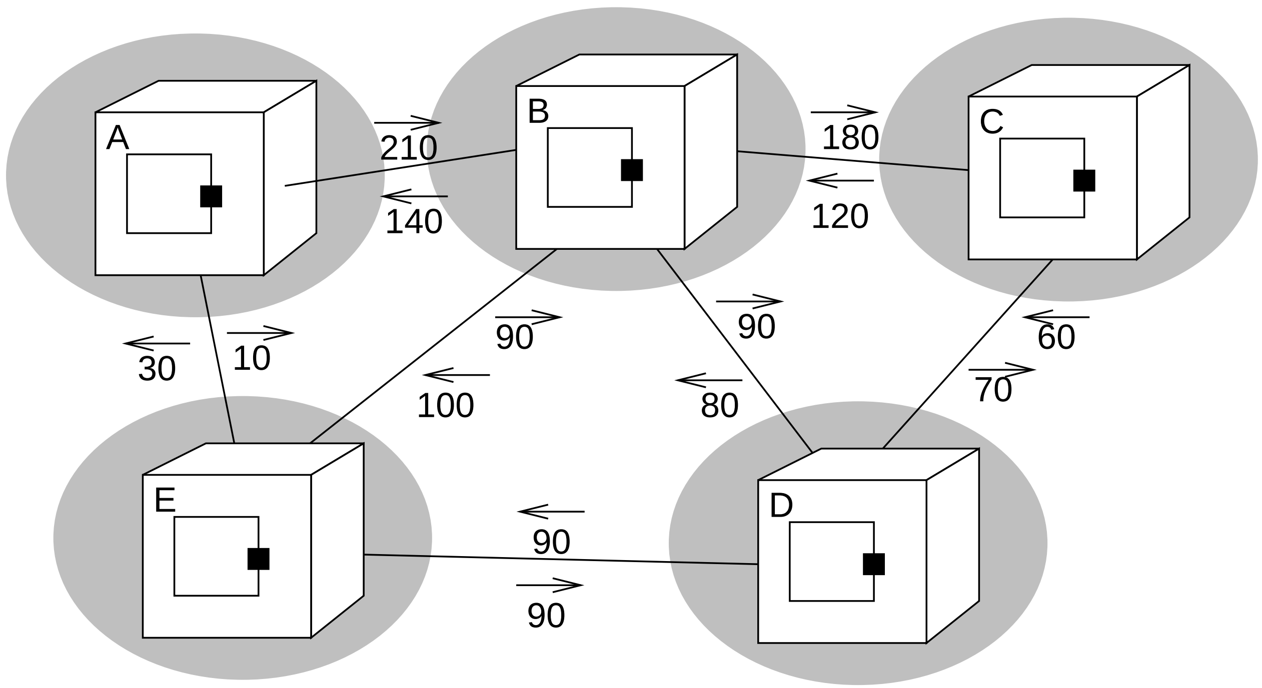
$$\begin{aligned} p_0 \cdot 2\lambda &= p_1 \cdot \mu \\ p_1 \cdot (\mu + 2\lambda) &= p_0 \cdot 2\lambda + p_2 \cdot 2\mu \\ p_2 \cdot (2\mu + 2\lambda) &= p_1 \cdot 2\lambda + p_3 \cdot 3\mu \\ p_3 \cdot 3\mu &= p_2 \cdot 2\lambda \end{aligned}$$

Außerdem gilt ja noch $p_0 + p_1 + p_2 + p_3 = 1$. Dadurch ist dieses lineare Gleichungssystem lösbar (Aufgabe!). So erhalte ich für jeden Zustand die (stationäre) Aufenthaltswahrscheinlichkeit (d.h., die p_i), und damit den Erwartungswert als $\sum_{i=0}^{3} i \cdot p_i$.

Durchsatz

Jetzt können wir zurück zu unserem Ausgangspunkt, nämlich der Berechnung des Durchsatzes unseres Bibliotheksnetzes.

Nehmen wir an, statistische Auswertungen des bisherigen Recherche- und Fernleihverkehrs lassen erwarten, daß in etwa folgendes Verkehrsaufkommen zu erwarten ist:



Mit anderen Worten, es gilt die folgende Zugangsmatrix:

	Nach				
Von	*A*	*B*	*C*	*D*	*E*
A		70	120	20	10
B	50		60	70	100
C	80	40		20	40
D	10	70	20		50
E	30	90	50	40	

Die Werte in der Matrix seien jeweils „Nachrichten pro Minute“, wobei eine Nachricht eine Anfrage sein kann, eine Antwort darauf, oder eine sonstige Information. Wenn die Zahlen groß erscheinen: beachte, daß z.B. Anfragen in der Regel nicht einzeln, sondern als Kataloganfragen, d.h. gleich über Mengen von Medien abgesetzt werden.

Summiert man jeweils die einzelnen Zeile, so erhält man den

Zugang je Knoten $E[Z_i] = \sum_{j \in V}$. Summiert man über die gesamte Matrix, so erhält man den Durchsatz des Netzes $E[D_N] = \sum_{i,j \in V} E[Z_{i,j}]$. Beide Werte sind in folgender erweiterter Zugangsmatrix dargestellt.

	Nach					Summe :
Von	A	B	C	D	E	Zugang/Knoten
A		70	120	20	10	220
B	50		60	70	100	280
C	80	40		20	40	180
D	10	70	20		50	150
E	30	90	50	40		210
Summe (Netzdurchsatz)						1040

Routing und Durchsatz von Nachrichten

Wir nehmen an, daß die meisten Bücher nur in einem Exemplar vorhanden sind, also in einer Filiale. Die Ausleihdaten dieses Exemplares (ist es ausgeliehen, wenn ja wann und von wem, ist es vorbestellt, etc.) werden in der jeweiligen Filiale verwaltet.

Allerdings ist das Netz kein vollständiger Graph, so daß manche Anfragen weitergeleitet weden müssen. Wir nehmen an, daß es folgende Verweismöglichkeiten gibt.

Von	Nach	führt über
A	D	B
A	C	B
C	E	D

Dadurch ergeben sich höhere Belastungen der Leitungen mit Nachrichten:

	Nach					*Summe*
Von	*A*	*B*	*C*	*D*	*E*	
A		210	−−	−−	10	220
B	140		180	90	100	510
C	−−	120		60	−−	180
D	−−	80	70		90	240
E	30	90	−−	90		210
						1360

Der Verkehrsengpaß (d.h. die Leitung mit der größten Belastung) ist offenbar die Leitung von A nach B. Ihre Auslastung (und damit die Netzauslastung insgesamt), also das Verhältnis von verfügbarer und benötigter Leistung, beträgt:

$$\text{Netzauslastung} = \frac{210 \text{ Pakete}}{684 \text{ Pakete}} = 0,3$$

Dieser Wert ist in Ordnung – mit anderen Worten, wir haben unser Netz gemäß dem Verkehrsaufkommen angemessen dimensioniert.

Durchsatz

Nehmen wir an, daß Datenpakete die Größe von 1 Kbit besitzen. Zusätzlich braucht jedes Paket 0,5 Kbit Verwaltungsinformation, zusammen also $P = 1,5$ Kbit je Paket. Bei einem Grenzdurchsatz (d.h. die maximale Leistung der Leitungen) von $c_L = 10$Mbit ergibt sich also für die Laufzeit L_P von Paketen

$$L_P = \frac{P}{c_L} = \frac{1,5 \text{ Kbit}}{1 \text{ Mbit/s}} = 1,46 \text{ ms}.$$

Der Durchsatz D_P ist der Kehrwert hieraus, also gut 684 Pakete/s.

Eine typische Nachricht umfasse 6 Kbit. Damit benötigt sie 6 Pakete zu $P = 1,5$ Kbit, also eine Bandbreite N von 9 Kbit. Daraus ergibt sich ein effektiver Grenzdurchsatz für die Applikation

c_a von abgerundet

$$c_A = \frac{1 \text{ Mbit/s}}{9 \text{ Kbit/Nachricht}} = 11,3 \text{ Nachrichten/s}$$

und eine Laufzeit L_N von Nachrichten entsprechend mit

$$L_N = \frac{N}{c_L} = \frac{9 \text{ Kbit}}{1 \text{ Mbit/s}} = 8,79 \text{ ms.}$$

Etappen

Dazu dividieren wir die Leitungsdurchsätze (Wege) durch die Zugangszahlen (Ziele), und erhalten die Etappenzahl (jeweils die Erwartungswerte hiervon).

$$E[\text{Etappenzahl}] = \frac{\sum_{i,j\in V} E[D_{i,j}]}{\sum_{i,j\in V} E[Z_{i,j}]} = \frac{1040}{1360} = 1,31$$

Rückblick

Wir haben

- unsere Modellierungssprache um den Leistungsaspekt erweitert
- dafür Semantiken angegeben
- und diese formalen Abbilder unserer Modelle mathematisch analysiert