

Pattern-basierte Erstellung eines E-Learning Spiels am Beispiel Network Calculus

Andreas Faatz, Krishna Pandit und Ralf Steinmetz

Fachgebiet Multimedia Kommunikation
TU Darmstadt
Merckstr. 25
64289 Darmstadt

{Andreas.Faatz,Krishna.Pandit,Ralf.Steinmetz}@kom.tu-darmstadt.de

Abstract: Spiele sind ein Weg, um neuartige Begriffe in interaktiver Form kennenzulernen und in einen modellhaften Kontext zu bringen. Spiele können einen simulativen Charakter haben, abstrahieren aber gleichzeitig von der Wirklichkeit und weisen den Teilnehmern Rollen zu, die sie Ungewohntes entdecken lassen. Das vorliegende Papier zeigt, wie sich für einen bestimmten, seiner Natur nach theoretisch-mathematischen Unterrichtsstoff ("Network Calculus") aus der Hochschullehre systematisch ein Online-Spiel planen läßt. Die Vorgehensweise bei der Planung und Definition des Online-Spiels wird von vornherein durch den Einsatz pädagogischer Patterns geleitet. Wir zeigen - und dies ist der hauptsächliche Erkenntnisgewinn des vorliegenden Papiers - wie sich spezielle Eigenschaften des zu unterrichtenden Lehrstoffes in idealer Weise für die Planung pädagogisch wertvollen interaktiven Lehrmaterials heranziehen lassen.

1 Einleitung

Pädagogische Patterns stellen bewährte Lösungen aus der Didaktik in strukturierter Form zusammen. Das vorliegende Papier greift einen Teilbereich existierender Pädagogischer Patterns, nämlich die sogenannte Subsprache (Sublanguage) der Patterns für aktives Lernen als Beispiel auf und liefert ein Gedankenexperiment zu der Frage, inwieweit ein theoretisch-formaler Unterrichtsinhalt mit Hilfe dieser Pädagogischen Patterns im Rahmen eines didaktischen Online-Spiels umsetzbar ist. Der besagte Unterrichtsstoff der in diesem Spiel vermittelt werden soll, ist Network Calculus. Network Calculus ist eine Systemtheorie für deterministische Warteschlagensysteme [Cr91]. Es ist insbesondere geeignet, um Dienstgüte in paketvermittelten Netzen, wie dem Internet, zu analysieren. Dies ist von zunehmender Bedeutung, da der Anteil von Datenverkehr aus Dienstgüte benötigenden Multimediaanwendungen steigt. Der Grundgedanke ist, dass Dienstgütegarantien gegeben werden können wenn der Datenverkehr reguliert wird, das Schedulerverhalten bekannt ist und Zugangskontrolle erfolgt. Das vorliegende Papier gliedert sich folgendermaßen: zunächst stellen wir im zweiten Teil der Einleitung die Basiselemente des zu behandelnden Unterrichtsstoffes Network Calculus vor. In Abschnitt 2 geben wir einen Überblick über Design Patterns und ordnen diese bezüglich anderer Formen der Wissensrepräsentation

im E-Learning ein. Abschnitt 3 stellt dar, welche Anforderungen an ein Online-Spiel sich aus den pädagogischen Patterns ableiten lassen, woraufhin Abschnitt 4 die Regeln für ein Online-Spiel vorstellt, die wir basierend auf der Analyse in Abschnitt 3 definieren. Abschließend fassen wir die Ergebnisse zusammen und geben einen Ausblick.

1.1 Network Calculus

Eine wichtige Eigenschaft des Network Calculus ist, dass er auf Min-Plus Algebra aufbaut. Dies ist wenig intuitiv und stellt beim Lernen eine große Herausforderung dar. Eine bedeutende Operation im Network Calculus ist die Min-plus Faltung. Sie ist mit dem Operator $*$ gekennzeichnet und wie folgt definiert.

$$(f * g)(t) = \min_{0 \leq s \leq t} \{f(t-s) + g(s)\} \quad (1)$$

Im Folgenden sind die elementaren Resultate von Network Calculus beschrieben. Wie in der linearen Systemtheorie, besteht ein Network Calculus System aus einem Eingang, einer Übertragungsstrecke und einem Ausgang. Hierbei beschreibt der Eingang den Verkehr, welcher vom Startknoten gesendet wird. Die Übertragungsstrecke ist das Netz, das es durchläuft, und der Ausgang ist der Verkehr der beim Empfänger ankommt. Der Eingangsverkehr wird mit sogenannten Ankunftscurven beschrieben. Eine Ankunftscurve gibt den Verkehr an, den der Sender maximal senden darf. Dies unterscheidet sich von der allgemeinen statistischen Verkehrsannahme und ist begründet mit dem Ziel, deterministische Dienstgütegarantien zu erzielen. Mathematisch ausgedrückt ist eine Funktion $\alpha(t)$ eine Ankunftscurve für einen Datenstrom $x(t)$ wenn für alle $0 \leq s \leq t$ gilt:

$$\alpha(t) \leq x(t) - x(s) \quad (2)$$

Ein Netz oder Netzelement ist durch eine sogenannte Dienstcurve beschrieben. Die Dienstcurve gibt zu jedem Zeitpunkt an, wie viele Daten mindestens verarbeitet worden sein müssen. Eine Funktion $\beta(t)$ ist eine Dienstcurve für ein Netzwerkelement mit Eingang $x(t)$ und Ausgang $y(t)$, falls für alle t gilt

$$y(t) \geq (x * \beta)(t) \quad (3)$$

Das heißt, ein Netzwerkelement bietet einem Datenstrom die Dienstcurve, wenn der Ausgang kleiner oder gleich der Min-Plus Faltung des Eingangs mit der Dienstcurve ist. Die Dienstcurve eines aus mehreren Elementen bestehenden Netzes ist gegeben durch die Min-Plus Faltung der Dienstcurven der einzelnen Elemente. So ist die Dienstcurve $\beta(t)$ eines Pfades, bestehend aus zwei Teilpfaden mit den jeweiligen Dienstcurven $\beta_1(t)$ und $\beta_2(t)$ gegeben durch

$$\beta(t) = (\beta_1 * \beta_2)(t) \quad (4)$$

Die maximale Verzögerung die ein Paket einer Ankunftscurve $\alpha(t)$ beim Durchlaufen eines Netzes mit der Dienstcurve $\beta(t)$ erfährt, ist gegeben durch die maximale vertikale

Distanz von $\alpha(t)$ und $\beta(t)$.

$$d = \sup_{t \geq 0} \{ \inf_{s \geq 0} \{ \alpha(t) \leq \beta(t + sec) \} \} \quad (5)$$

Das Blind Multiplexing Resultat gibt die Dienstkurve an, die ein Datenstrom im ungünstigsten Fall erhält, falls noch andere Datenströme den Pfad teilen. Seien $\alpha_i, i = 1, 2, \dots, n$ alle Ankunftscurven die sich einen Link mit der Dienstkurve β teilen. Dann erhält jeder Datenstrom $\alpha_i(t)$ die Dienstkurve $\beta_i(t)$.

$$\beta_i(t) = \max \{ 0, \beta(t) - \sum_{j=1, j \neq i}^n \alpha_j(t) \} \quad (6)$$

Das vorliegende Papier stellt vor, wie man diesen Unterrichtsstoff durch eine e-Learning-Unterstützung anschaulich vermitteln kann. Bei der Gestaltung dieser e-Learning-Unterstützung eines Online-Spiel, greifen wir auf das Hilfsmittel der Pädagogischen Design Patterns (im weiteren Verlauf auch Pädagogische Patterns genannt) zurück.

2 Pädagogische Patterns

Der folgende Abschnitt beginnt mit einer allgemeinen Erläuterung von Design Patterns als systematische Lösungsaufbereitung eines Fachgebietes. Darauf folgt eine Übersicht über den aktuellen Stand zu sogenannten pädagogischen Patterns, die diesen Ansatz auf die Didaktik übertragen.

2.1 Design Patterns

Der Begriff des Design Patterns stammt aus der Architektur. Die dort angesiedelten Arbeiten Christopher Alexanders [AIS77] wurden in den achtziger Jahren auf Softwareentwurf und -erstellung übertragen. Ein Design Pattern formuliert in standardisierter Form Lösungen für ein bestimmtes Problem. Die Lösung wird allerdings bewusst nicht in algorithmischer Form oder durch formalisierte Modelle dargestellt, sondern natürlichsprachlich ausgedrückt. Das Ziel dieser Vorgehensweise ist ein Ausgleich zwischen der flexiblen Wiederverwendbarkeit und der exakt nachvollziehbaren Lösungsbeschreibung. Dieser für qualitativ hochwertige Patterns kennzeichnende Formalisierungsgrad spiegelt sich auch in der ursprünglichen Intention Christopher Alexanders und seiner Aussage, Zweck eines Design Patterns sei es, es Lösungsansätze millionenfach wiederzuverwenden, ohne jedoch jemals alle Lösungsdetails zu wiederholen[AIS77] wider.

Die erwähnte Standardisierung wird durch den standardisierten Aufbau eines Design Patterns erreicht. Typischerweise beschreibt ein Design Pattern immer die Problemstellung, die durch das Pattern gelöst werden soll, und es gibt den Kontext an, in welchem sich Lösungsvarianten abspielen können. Ein hochwertiges Design Pattern beschreibt als sogenannte Forces auch stets Wechselwirkungen der einzelnen Lösungsbestandteile, das heißt,

den Einfluss einzelner Entscheidungen auf andere relevante Parameter. Abhängigkeiten zu anderen Lösungen können angezeigt werden, so dass aus mehreren Design Patterns ein größerer Lösungsansatz entstehen kann. Ein Beispiel hierfür sind die Arbeiten Schumachers [Sc03], der aus fachgebietsspezifischen Wissensrepräsentationen Schlussfolgerungen über den Einsatz spezieller Patterns trifft. Das allgemeine Anwendungsgebiet liegt bei Schumacher im Bereich der Sicherheitstechnologie, die Wissensrepräsentation erfasst vor allem die Wechselwirkung zwischen Angriffen auf Rechnersysteme. Im Gegensatz zu einer solchen hochstrukturierten Aufarbeitung eines Wissensgebietes stehen Sammlungen von Patterns, sogenannte Pattern Languages, wie wir sie beispielsweise im Bereich Didaktik vorfinden. Hier ist die Kopplung der Patterns allerdings nicht formalisiert. Wir ordnen diesen Ansatz nun in das Gebiet der Wissensrepräsentation im E-Learning ein.

2.2 Einsatz von Design Patterns in der Pädagogik

Verschiedene Ansätze innerhalb des e-Learning widmen sich der Frage einer wissensbasierten Unterstützung. So gehört die Erschließung eines zu unterrichtenden Fachgebietes zum Zwecke einer besseren Suche [FS03] oder zur Herstellung von Adaptivität der E-Learning-Materialien zum Anwendungsgebiet von Metadaten [Ho03] und Ontologien [FS03], [SSS01]. Wissensbasierte Ansätze e-Learning-spezifischer Auszeichnungssprachen widmen sich der genauen Beschreibung von Dokumenten und Dokumentteilen zur Erhöhung ihrer Flexibilität im Layout oder wiederum zur Verbesserung der Adaptivität [Fr02]. Diese Ansätze der Wissensrepräsentation im E-Learning beziehen sich auf die elektronischen Lernressourcen. Im Gegensatz dazu stehen Modellierungssprachen wie EML [Ko03]. EML abstrahiert von den eigentlichen Lerninhalten und Lernressourcen und stellt Mittel für eine formale Modellierung instruktionaler Designs zur Verfügung. Damit lassen sich Lehrveranstaltungen unterschiedlicher Art detailliert planen und didaktische Prozesse steuern. Ein ähnliches Ziel verfolgen die pädagogischen Design Patterns, die von den Teilnehmern des Pedagogical Patterns Project [PP06] gesammelt und in regelmäßigen Shepherding-Prozessen [BRP:04] evaluiert und iteriert werden. Im Gegensatz zum oben umrissenen stark formalisierten Ansatz im Sicherheitsbereich werden die pädagogischen Design Patterns als Sammlungen mit einheitlichem Vokabular, Angabe des Reifegrades eines Patterns und Querbezügen entwickelt. Der Begriff der Pattern Language ist somit als Sammlung und Überdeckung eines Wissensgebietes, hier der Didaktik, zu verstehen. Zudem beschäftigen sich die pädagogischen Design Patterns mit dem Bereich der Hochschullehre, in vielen ist von einem Seminarkontext die Rede. Dies und der umschriebene Formalitätsgrad lässt sie zu einer idealen Quelle für die Gestaltung eines didaktischen Online-Spiels zur Hochschullehre werden. Als Wissensrepräsentation sind pädagogische Design Patterns zusammenfassend betrachtet zwischen formalen und rein natürlichsprachlichen Ansätzen angesiedelt. Im Vergleich zu anderen Bereichen, die durch Design Patterns erfasst wurden, tendieren die pädagogischen Patterns in ihrem derzeitigen Entwicklungsstadium allerdings zu einem Text mit standardisierter Gliederung hin.

3 Ableitung von Designkriterien für ein Online-Spiel aus den pädagogischen Design Patterns

Der folgende Abschnitt greift eine Teilmenge (im englischen Sprachgebrauch: Sublanguage) der pädagogischen Design Patterns [PP06] heraus und leitet daraus direkt Anforderungen an ein Online-Spiel ab. Diese Design Patterns werden dann in Kriterien an das Spiel zum Network Calculus übersetzt. Die Übersetzung ist notwendig, da es noch keine explizite Anpassung der betreffenden Patterns an eine abstrakte Spielsituation gibt.

Wir widmen uns zunächst zwei Patterns, die die allgemeinsten beim Design eines Online-Spiels darstellen.

Das zentrale Pattern der Patternsprache Active Learning ist das Active-Student-Pattern. Jedes andere Pattern bezieht sich auf die Prinzipien dieses Patterns, das den Lernenden als Handelnden in den Fokus rückt. Das Pattern verlangt einen radikalen Bruch mit dem Frontalunterricht. Im Gegensatz dazu wird Wert auf eine ständige Aktivierung des Lernenden gelegt. Dies betrifft auch die Neueinführung von Begriffen, zu denen eine fachliche Wissensvermittlung erfolgt. Das Pattern geht so weit, reine Leseaktivitäten der Lernenden als möglicherweise zu passiv herauszustellen. Als Designkriterium für ein Spiel leiten wir daraus ab, dass die Regeln des Spiels vergleichsweise einfach formuliert sein sollten und gleichzeitig eine dem Unterrichtsthema angemessene Komplexität erlauben sollten. Zudem sollten sich einige der weiteren im Design zu berücksichtigenden Eigenschaften stärker aus dem Spielverlauf - beispielsweise als Elemente erfolgreicher Gewinnstrategien - und nicht durch die künstliche Einführung weiterer Regeln ergeben. Ein besonderes Kriterium zur Überprüfung der Klarheit und Einfachheit der Spielregeln ist die Einführbarkeit des Spiels als Solitärspiel. Gleichzeitig werden wir als Folge eines der spezielleren Patterns eine Variante als Mehrpersonenspiel fordern. Ein Spiel, das sowohl als Solitär- als auch als Mehrpersonenspiel funktionieren soll, kann im computerunterstützten Fall durch programmierte Agenten, die die Rolle einzelner Spielteilnehmer bei Bedarf übernehmen, realisiert werden. Dies stellt ein wesentliches Motiv dar, das Spiel als e-Learning-Umsetzung zu ermöglichen.

Als weiteres ausgereiftes und vergleichsweise allgemeines Pattern ist das Invisible-Teaching Pattern anzusehen, das die Vorteile eines In-den-Hintergrund-Tretens der Lehrkraft herausstellt. Wir sehen die Anforderungen, die durch dieses Pattern entstehen, durch Spiele an sich bereits weitestgehend erfüllt, es bleibt die sehr allgemeine Anforderung, dass das Spiel dazu in der Lage ist, fachliche Inhalte zu vermitteln.

Bei der Beschreibung der weiteren spezielleren Patterns und der resultierenden Anforderungsanalyse betrachten wir hauptsächlich diejenigen Patterns, die von der Pedagogical Patterns Organisation als wichtiger und ausgereifter angesehen werden. In der Notation der Patterns äußert sich dies in der Vergabe von zwei Sternen (**), auf das Pattern im Gegensatz zu Patterns, die erst mit einem Stern (*) ausgezeichnet wurden.

Aus dem Honor-Questions-Pattern geht hervor, dass es zur Aktivierung des Lernenden unbedingt hilfreich sei, in jeder Phase des Lernprozesses zu Fragen zu ermuntern. Die Fragen können an den Unterrichtenden, vorzugsweise aber an die anderen mit dem Unterrichtsstoff befassten Lernenden gerichtet sein. Das Pattern legt deutlich Wert darauf, dass es prinzipiell immer (zeitlich und thematisch gesehen) möglich sein muss, Fragen zu stellen. Für

die Planung des Network Calculus Spiels heißt das, dass komplexe Spielsituationen, die sich auch aus einfachen Regeln ergeben können, replizierbar sein müssen, um so eine präzise Grundlage des Fragens herzustellen. Eine Analogie ist etwa zum Schach zu sehen, wo eine standardisierte Notation [So81] es auch im nicht computergestützten Fall ermöglicht, einzelne Partien nachvollziehbar werden zu lassen. Die Ermöglichung einer exakten Fragestellung zu vergangenen Spielsituationen führt hier zu einer Kultur der Partianalyse und der Konkretisierung (bis hin zur Verwissenschaftlichung) von Fragestellungen. Network Calculus eignet sich zu einer im weiteren Sinne zugbasierten Darstellung da neue Datenströme in das Netz diskret und sequentiell aufgenommen werden können. Alternativ dazu - und in stärkerer Analogie zum Schachspiel - könnte die Diskretisierung des Spielverlaufes und die damit möglichen Anknüpfungspunkte für Fragen an die Entscheidungen der Spieler (Lernenden) gekoppelt sein. Die Spieler wiederum reagieren auf das Versenden von Paketen in einem durch das Spiel simulierten Kommunikationsnetz.

Das Groups-Work-Pattern stellt darauf ab, wie wichtig Gruppenarbeit für den Lernprozess ist. Eine Gruppe Lernender ist durch ihre Interaktion dazu in der Lage, sich auch neuen Unterrichtsstoff gegenseitig zu vermitteln. Die Gruppen sollten nach dem Groups-Work-Pattern für die Dauer eines Kurses oder Semesters nicht starr festgelegt sein. Übertragen auf das Network Calculus Spiel erzeugt dies die Anforderung, ein Spiel zu definieren, in dem Gruppenarbeit nötig ist. Diese Gruppen entstehen im Idealfall aus den Notwendigkeiten des Spielverlaufes heraus. Zusammen mit der wichtigen Anforderung, zu Fragen zu ermutigen, entsteht hier für das Network Calculus Spiel das Ziel einer Spielgestaltung, die den Weg zum Spielerfolg sehr stark durch Gruppenkommunikation fördert. Die spontan gebildete Gruppe von Spielern, die intensiv über ihre Strategien kommuniziert, soll mit größerer Wahrscheinlichkeit auch die Sieger des Spiels hervorbringen. Hier verschränken sich auch kooperative und kompetitive Elemente des Spiels. Als Online-Spiel muss ein Kanal für synchrone Kommunikation (schriftlich, das heißt als Chat, oder audiovisuell) geschaffen werden, der es Gruppen beliebiger Größe während des Spielverlaufes ermöglicht, abgeschirmt von anderen Gruppen und Einzelteilnehmern zu kommunizieren. Eine Gruppe kann über die Aufnahme eines weiteren Mitgliedes in die abgeschirmte Kommunikation entscheiden, als Möglichkeiten wären hier Mehrheitsentscheide oder Konsensentscheide in technisch unterstützter Form denkbar. Die geforderte Form der Gruppenkommunikation stellt ein weiteres wesentliches Motiv dar, das Spiel als e-Learning-Umsetzung zu planen. Zwar enthält die eingeführte Teilmenge auch ein Pattern für ein sogenanntes Kriegsspiel (War Game), allerdings lässt dieses Pattern beim derzeitigen Stand einige Fragen offen. So erscheint allein schon der Name irreführend und die Konzeption des Patterns läuft auf feste, vordefinierte Gruppen hinaus. Zur Arbeitsteilung dieser Gruppen und zu kooperativen Elementen des Spiels im Gegensatz zur Betonung kompetitiver Elemente wird im War-Game-Pattern nichts Genaueres ausgeführt. Auch stellt das Pattern derzeit noch auf eine stark Intervention des Lehrenden bei der Vermittlung des Regelwerks ab. Das kann bei Spielen wie dem im Pattern referenzierten Monopoly hilfreich sein, da sie in ihrem Gehalt hinreichend weit von realen Wirtschaftsabläufen entfernt und gleichzeitig mit Allgemeinbildung der Spielteilnehmer gut zu erfassen sind. Je mehr allerdings das Spiel selbst die direkte Anwendung von tatsächlich zu unterrichtenden Konzepten verlangt und je abstrakter diese Konzepte sind, desto mehr läuft eine genaue Instruktion zum Spiel auch eine genaue Instruktion zu den fachlichen Inhalten hinaus. Wir sehen hier eine Konkurrenz zum

Ziel des bereits beschriebenen Patterns zum selbstbestimmten Entdecken neuer fachlicher Konzepte während des Lernvorgangs. Somit ist die in diesem Abschnitt getroffene Neukonstruktion von Anforderungen an ein Spiel aus den einzelnen Patterns der Subsprache für aktives Lernen notwendig und kann als Konkretisierung des auch aus der Sicht der Pedagogical Patterns Gruppe noch nicht völlig ausgereiften War-Game-Patterns verstanden werden.

4 Das Network-Calculus-Spiel

Im Folgenden werden die Regeln des Spiels dargestellt und anhand eines Beispiels gezeigt, wie diese Regeln mit den Anforderungen des Abschnittes 3 korrespondieren.

4.1 Regeln

Ziel des Spiels ist es, Datenströme mit einer möglichst geringen Worst Case Verzögerung durch das Netz zu routen, das heißt, entlang eines bestimmten Weges zu leiten. Das Netz richtet sich nach der Anzahl der Spieler und besteht aus m Knoten, die mit Großbuchstaben gekennzeichnet sind. Für jedes Spiel wird eine Netzwerktopologie zufällig gewählt, in dem Verbindungen zwischen Knoten auf dem als Rechteck zu visualisierenden Spielfeld gesetzt werden. Verbindungen sind gerichtet und es gibt keine rückwärts gerichteten Verbindungen. Jede Verbindung ist durch die gleiche, vorher festgelegte Dienstkurve beschrieben. Das Spiel besteht aus n Spielrunden. Zu Beginn jeder Spielrunde wird ein Paar von Sende- und Empfängerknoten zufällig gebildet. Die Sendeknoten sind in der ersten Spalte und die Empfängerknoten in der letzten. Jeder Spieler muss nun einen Datenstrom vom Sende- zum Empfängerknoten routen. Jeder Datenstrom genügt einer vorher festgelegten Ankunftskurve. Die Route ist frei wählbar. Nachdem jeder Spieler seine Route (den Weg, entlang dessen ein Paket geleitet wird) eingegeben hat, wird der Zustand des Netzes neu berechnet. Jede Verbindung erhält eine neue Dienstkurve, die sich berechnet aus der Initialdienstkurve minus der Summe aller Ankunftskurven die diese Verbindung verwenden. Dies ist das Ende einer Spielrunde. Nach n Spielrunden, werden für jeden Spieler die Ende-zu-Ende Verzögerungen aller seiner Datenströme aufaddiert. Die Ende-zu-Ende Verzögerung eines Datenstroms ist die horizontale Distanz zwischen der Ankunftskurve und Ende-zu-Ende Dienstkurve. Die Ende-zu-Ende Dienstkurve ergibt sich aus der Min-plus Faltung aller Dienstkurven der Verbindungen die der Datenstrom durchläuft. Natürlich sind die einzelnen Verbindungsdienstkurven unterschiedlich je nach Anzahl der Datenströme die ihn durchlaufen. Der Spieler mit dem geringsten Wert hat gewonnen.

4.2 Beispiel

Es nehmen 3 Spieler teil und das Spiel besteht aus $n = 2$ Spielrunden. s_i indiziert den i ten Spieler. Abbildung 1 zeigt das zugrunde liegende Netz.

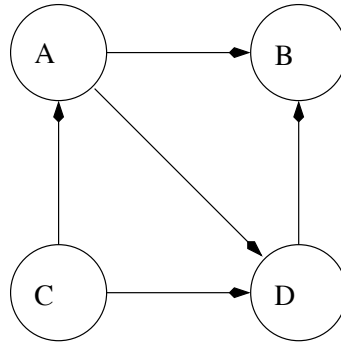


Abbildung 1: Beispielnetzwerk

Jede Verbindung hat die Dienstkurve

$$c(t) = \begin{cases} 0 & t \leq 1 \\ 6t & t > 1 \end{cases} \quad (7)$$

Die Funktion $v_{ab}(i)$, $a, b \in \mathcal{S}$, $i = 1, \dots, r$ beschreibt den Zustand der Verbindung zwischen den Knoten a und b nach der i ten Spielrunde.

Die Datenströme sind in der folgenden, als Token Bucket bekannten, Form

$$f(t) = \begin{cases} 0 & t \leq 0 \\ 3 + t & t > 0 \end{cases} \quad (8)$$

f_i gibt den Datenstrom der i ten Spielrunde an. Der Pfad den der Spieler s_j in der i ten Spielrunde wählt ist beschrieben durch $p_j(i)$. $h_j^i(k)$, $k \geq i$ gibt die Ende-zu-Ende Dienstkurve an, die der Datenstrom f_i des Spielers s_j nach der k ten Spielrunde sieht. Der Wert $d_j^i(k)$, $k \geq i$ gibt die Verzögerung für den Datenstrom f_i nach der k ten Spielrunde des Spieler j an.

$z(i)$ gibt das Paar von Start- und Zielknoten der i ten Spielrunde an.

Sei $z(1) = (A, B)$. Die ersten beiden Spieler wählen den direkt Pfad von A nach B , während der Spieler s_3 den Pfad über D geht.

$$p_1(1) = p_2(1) = (A, B) \quad (9)$$

$$p_3(1) = (A, D, B) \quad (10)$$

Demnach ergeben sich folgende Zustände auf den Verbindungen. Der Begriff Zustand bedeutet hier, dass eine neue Ankunftscurve auf der Verbindung die errechnete Dienstkurve

sieht.

$$v_{AB}(1) = [c - 2f]^+ = [-12 + 4t]^+ \quad (11)$$

$$v_{AD}(1) = v_{DB}(1) = [c - f]^+ = [-9 + 5t]^+ \quad (12)$$

$$v_{CA}(1) = v_{CD}(1) = c \quad (13)$$

Die Ende-zu-Ende Dienstkurven die die Datenströme sehen ergeben sich wie folgt. Für die Spieler s_1 und s_2 teilen sich die Verbindung v_{AB} und müssen daher beide damit rechnen dass der andere immer zuerst bedient wird. D.h., beide sehen die Ursprungsdienstkurve minus einem Datenstrom.

$$h_1^1(1) = h_2^1(1) = [c - f]^+ = [-9 + 5t]^+ \quad (14)$$

Nun lassen sich die Verzögerungen der Datenströme der Spieler berechnen. Die Ende-zu-Ende Verzögerung des Datenstroms f_i des Spielers s_j nach der k ten Spielrunde sei gegeben durch $d_j^i(k), k \geq i$. So ist $d_1^1(1)$ die maximale horizontale Distanz zwischen f_i und $h_1^1(1)$ und somit $d_1^1(1) = \frac{12}{5}$. Da $h_2^1(1) = h_1^1(1)$ ist $d_2^1(1) = d_1^1(1) = \frac{12}{5}$. Die Ende-zu-Ende Dienstkurve des Spielers 3 ist

$$h_3^1(1) = [v_{AD}(1) + f]^+ * [v_{DB}(1) + f]^+ = [-12 + 6t]^+ \quad (15)$$

Daraus ergibt sich Ende-zu-Ende Verzögerung für den Spieler s_3 für den Datenstrom f_1 nach der ersten Spielrunde.

$$d_3^1(1) = f \oplus h_3^1(1) = 5 \quad (16)$$

In der zweiten Spielrunde ist ein Datenstrom von C nach B zu realisieren, d.h., $z(2) = (C, B)$. Der Spieler s_1 wählt den Pfad über D , während die beiden anderen Spieler s_2 und s_3 den Pfad über A bevorzugen.

$$p_1(2) = (C, D, B) \quad (17)$$

$$p_2(2) = p_3(2) = (C, A, B) \quad (18)$$

Die Zustände der Verbindungen sind danach wie folgt.

$$v_{AB}(2) = [c - 4f]^+ = [-18 + 2t]^+ \quad (19)$$

$$v_{AD}(2) = v_{CD}(2) = [c - f]^+ = [-9 + 5t]^+ \quad (20)$$

$$v_{CA}(2) = v_{DB}(2) = [c - 2f]^+ = [-12 + 4t]^+ \quad (21)$$

Nun lassen sich die endgültigen Verzögerungen aller Datenströme berechnen. Da sich die Zustände auf den Verbindungen geändert haben, müssen die Verzögerungen der Datenströme aus der ersten Runde neu berechnet werden.

$$\begin{aligned}
d_1^1(2) &= f \oplus h_1^1(2) = f \oplus (v_{AB}(2) + f) = [3 + t]^+ \oplus [-15 + 3t]^+ = 6 \text{ (} \\
d_2^1(2) &= f \oplus h_2^1(2) = f \oplus (v_{AB}(2) + f) = 6 \text{ (} \\
d_3^1(2) &= f \oplus h_3^1(2) = f \oplus ((v_{AD}(2) + f) * (v_{DB}(2) + f)) = f \oplus [-14 + 5t]^+ = \frac{17}{5} \text{ (}
\end{aligned}$$

Die Verzögerungen der Datenströme aus der zweiten Runde sind wie folgt.

$$d_1^2(2) = f \oplus h_1^2(2) = f \oplus ((v_{CD}(2) + f) * (v_{DB}(2) + f)) = \frac{17}{5} \quad (25)$$

$$d_2^2(2) = f \oplus h_2^2(2) = f \oplus ((v_{CA}(2) + f) * (v_{AB}(2) + f)) = \frac{39}{5} \quad (26)$$

$$d_3^2(2) = f \oplus h_3^2(2) = f \oplus ((v_{CA}(2) + f) * (v_{AB}(2) + f)) = \frac{39}{5} \quad (27)$$

Das Endresultat des Spielers s_j ist y_i .

$$y_j = \sum_{i=1}^n d_j^i(n) \quad (28)$$

In diesem Beispiel, ergibt sich als Endresultat für jeden Spieler die Summe der Verzögerungen seiner beiden Datenströme.

$$y_1 = d_1^1(2) + d_1^2(2) = 6 + \frac{17}{5} = \frac{47}{5} \quad (29)$$

$$y_2 = d_2^1(2) + d_2^2(2) = 6 + \frac{39}{5} = \frac{69}{5} \quad (30)$$

$$y_3 = d_3^1(2) + d_3^2(2) = \frac{17}{5} + \frac{39}{5} = \frac{56}{5} \quad (31)$$

D.h., Spieler s_1 hat das Spiel gewonnen.

5 Bezug zu den Design Patterns

Der Ablauf des Spiels zeigt stets die tatsächliche Wirkungsweise des Network Calculus und dient damit fachlich zur Vermittlung seiner Konzepte. Auch die anderen Anforderungen aus Abschnitt 3 sehen wir gut erfüllt.

Zunächst sei die Wirkung von Kooperation zwischen Spielern exemplarisch erläutert. Wie im obigen Beispiel sei Abbildung 1 das zugrunde liegende Netz. Es seien alle Annahmen übernommen mit einer Ausnahme. Das Spiel besteht nur aus einer Spielrunde $z(1) = (C, B)$. Die Spieler s_1 und s_2 sprechen sich ab um zu verhindern dass sie sich eine Verbindung teilen müssen. So nimmt Spieler s_1 den Pfad $p_1(1) = (C, A, B)$ und Spieler s_2

den Pfad $p_2(1) = (C, D, B)$. Spieler s_3 hat drei Möglichkeiten. Entweder, er nimmt den Pfad (C, A, B) . In diesem Fall teilt er sich die Verbindung mit Spieler s_1 , während Spieler s_2 eine Verbindung für sich alleine hat und somit das Spiel gewinnt. Oder er wählt den Pfad (C, D, B) , in welchem Fall Spieler s_1 das Spiel gewinnt. Die dritte Möglichkeit ist der Pfad (C, A, D, B) . Dann allerdings teilen sich die Spieler s_1 und s_2 den Sieg und Spieler s_3 ist der alleinige Verlierer. Es bleibt festzuhalten, dass es durch die Kooperation der Spieler s_1 und s_2 unmöglich für Spieler s_3 wird, das Spiel zu gewinnen. Die Anreize zur Kooperation sind also dem Spiel inhärent.

Generell kann durch Absprachen zum Teilen der Ressourcen verhindert werden, dass Spieler dieselbe Verbindung nehmen und sich dabei gegenseitig stören. Absprachen müssen allerdings durch einen vertraulichen Kommunikationskanal (z.B. Chat) geschaffen werden. Zudem benötigt man als Spieler zunächst programmierte Werkzeuge, die die nötige Berechnung unterstützen. So wird es möglich, im Spielverlauf eine stärkere Intuition für den Network Calculus zu entwickeln. In einer schwierigeren Stufe des Spiels können die Spieler angewiesen werden, die Berechnungen selbst durchzuführen. Es ist dem Lehrenden möglich, sich weitgehend aus dem Spielverlauf zurückzuziehen, und die Einführung als einfaches Solitärspiel ist umsetzbar, indem die Wahl der Routen durch das Netz für den Computergegner zufällig erfolgt (und die entsprechenden Verzögerungen ähnlich wie in unserem Beispiel dem Einzelspieler als Berechnung angezeigt werden).

Fragen zum Spielverlauf werden zum einen durch die inhärente Kooperation, andererseits durch die prinzipielle Wiederholbarkeit einzelner Partien möglich. Die Wiederholbarkeit kann durch eine Visualisierung der Spielzüge auf dem zufällig gewählten Spielbrett unterstützt werden. So ist es denkbar, die Datenraten und eintreffenden Pakete sowie ihre Verarbeitung bei der Spielwiederholung als (aufgrund des Spielverlaufes ad hoc generierten) Film anzuzeigen, wobei Datenpakete unterschiedlicher Spieler unterschiedlich markiert sein müssen.

Schließlich sind die Regeln des Spiels so definiert, dass auch größere Gruppen in größeren Netzen spielen können, aus denen sich spontan kooperierende Teilgruppen bilden würden. In diesem Sinne ist eine Skalierbarkeit des Spiels möglich.

6 Zusammenfassung und Ausblick

Das Papier stellt einen ersten konzeptionellen Versuch dar, theoretisch-mathematische Unterrichtsinhalte systematisch mit Hilfe von Design Patterns als Online-Spiel zu erschließen. Wir haben uns auf grundlegende Aspekte des Network Calculus beschränkt und unsere Analyse nur für einen Teilbereich der heute zur Verfügung stehenden Design Patterns aus der Pädagogik durchgeführt. Trotzdem ist uns die Definition eines Spiels gelungen, das nun implementiert und auf seinen pädagogischen Gehalt hin überprüft werden kann. Der Erkenntnisgewinn besteht außer in der konkreten Definition des Spiels auch im Gestaltungsprozess, der in den Abschnitten 3 und 4 sichtbar wurde. Dieser Prozess ist in zukünftigen Arbeiten in zweierlei Hinsicht auszuweiten: eine Vielzahl ähnlich abstrakter Unterrichtsinhalte im Kommunikationsbereich kann auf ihre spielerische Umsetzbarkeit überprüft werden. Andererseits könnten die anderen Teilmengen (Sublanguages) pädago-

gischer Design Patterns zu ähnlichen Anforderungsanalysen und Gestaltungsprinzipien von Online-Spielen herangezogen werden.

Literatur

- [AIS77] C. Alexander, S. Ishikawa, M. Silverstein. A pattern language, New York: Oxford University Press, 1977
- [BRP:04] R. Baggetun, E. Rusman, C. Poggi. Design Patterns For Collaborative Learning: From Practice To Theory And Back, Proceedings of ED-MEDIA, 2004
- [Cr91] R. Cruz. A calculus for network delay, part I: network element in isolation, IEEE Transactions on Information Theory 37:11, 114-131, 1991
- [FS03] A. Faatz, C. Seeberg, and R. Steinmetz. Ein Begriffsnetz für ein medizinisches Online-Lernprojekt. In Proceedings des LEARNTEC '03 Ü WORKSHOPS Aufbau und Einsatz von Begriffsnetzen zur semantischen Suche von Wissensmaterialien", pages 7-11, March 2003
- [Fr02] B. Freitag. LMML - Eine XML - Sprachfamilie fuer eLearning Content, GI Jahrestagung, 2002
- [Ho03] Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic, Oliver Merkel, Andreas Faatz, and Ralf Steinmetz. Building Structures of Reusable Educational Content Based on LOM. In Proceedings of The 15th Conference On Advanced Information Systems Engineering, Workshop on Semantic Web for Web-based Learning, June 2003
- [Ko03] R. Koper. Combining re-usable learning resources and services to pedagogical purposeful units of learning. In A. Littlejohn (Ed.), Reusing Online Resources: A Sustainable Approach to eLearning, London: Kogan Page, 2003
- [BT01] J.Y. Le Boudec, P. Thiran. Network Calculus - A Theory of Deterministic Queuing Systems for the Internet. LNCS 2050, Springer Verlag, 2001.
- [PP06] Pedagogical Patterns Project. <http://www.pedagogicalpatterns.org>
- [Sc03] M. Schumacher. Security Engineering with Patterns. Lecture Notes in Computer Science, LNCS 2754, Springer, 2003
- [So81] A.P. Sokolskii. Your first move: chess for beginners, Moscow: Raduga Publishers, 1981/1984
- [SSS01] L. Stojanovic, S. Staab, R. Studer. eLearning based on the Semantic Web, WebNet2001-World Conference on the WWW and Internet, 2001