

Studiengang: Informatik
Betreuer: Dr. Cora Burger
Prüfer: Prof. Dr. Kurt Rothermel

Beginn am: 01. März 2003
Beendet am: 31. August 2003

CR-Klassifikation: C.2.4, H.5.3, I.6.7, K.3.1

Studienarbeit Nr. 1891

Zustandsabhängige Verankerung von Zusatzinformation bei Protokollsimulationen

Robert Sauter

Fakultät Elektrotechnik, Informatik,
Informationstechnik
Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner
Abteilung Verteilte Systeme
Universitätsstr. 38
70569 Stuttgart

Zusammenfassung

Untersucht werden Anforderungen und Konzepte zur Erweiterung eines bestehenden Systems für Protokollsimulationen um die Möglichkeit der zustandsabhängigen Verankerung von Zusatzinformationen. Die elektronische Kollaborationsunterstützung steht dabei besonders im Vordergrund.

Dazu werden Einsatzszenarien definiert und daraus Anforderungen abgeleitet. Neben der Betrachtung bisheriger Ansätze für Annotationen in Protokollsimulationen werden auch Konzepte für Annotationen aus anderen Bereichen analysiert.

Der Schwerpunkt der Arbeit liegt auf der Diskussion von Lösungsansätzen für die Anforderungen und dem Entwurf des Gesamtsystems.

Inhalt

1	Einleitung	1
1.1	Motivation.....	1
1.2	Aufgabenstellung	1
1.3	Überblick	2
2	Grundlagen	4
2.1	Begriffe	4
2.2	Spin	5
2.3	Promela	6
2.4	HiSAP	7
2.5	HiSPIN.....	8
2.6	NUSS	8
2.7	Donut/ITS	8
2.8	ProDuctivE.....	9
2.9	Fazit	10
2.10	Quellen.....	11
3	Szenarien und Anforderungen.....	12
3.1	Phasen	12
3.1.1	Erstellungsphase.....	12
3.1.2	Diskussionsphase.....	12
3.1.3	Resultatsphase	12
3.2	Szenarien.....	12
3.2.1	Gruppenunterricht.....	13
3.2.2	Selbstlernumgebung	13
3.2.3	Lerngruppen	13
3.3	Szenarieneinordnung.....	13
3.4	Anforderungen	14
3.4.1	Dokumenttyp	15
3.4.2	Speicherung der Zusatzinformationen.....	15
3.4.3	Dokumentbezug.....	16
3.4.4	Verankerung von Zusatzinformationen	16
3.4.5	Weitergabe.....	17
3.4.6	Sichtbarkeit.....	18
3.4.7	Rechte	19
3.4.8	Art der Zusatzinformationen	19
3.4.9	Metainformationen über Zusatzinformationen	20
3.4.10	Darstellung	21
3.4.11	Übersicht und Statistiken.....	21
3.4.12	Verankerungsrichtung	21
3.4.13	Navigationsverankerung.....	22
3.4.14	Filterung	22
3.4.15	Bedienung.....	22
3.4.16	Generizität, Modularität und Erweiterbarkeit.....	22
3.5	Fazit	23

3.6	Quellen.....	23
4	Verwandte Arbeiten	24
4.1	Übersicht.....	24
4.2	Analyse Adobe Acrobat.....	25
4.2.1	Analyse hinsichtlich der Anforderungen.....	25
4.2.2	Fazit.....	27
4.3	Shared Powerpoint.....	27
4.3.1	Analyse hinsichtlich der Anforderungen.....	28
4.3.2	Fazit.....	30
4.4	ns Network Simulator	30
4.4.1	Analyse hinsichtlich der Anforderungen.....	30
4.4.2	Fazit.....	32
4.5	Ilias.....	32
4.5.1	Analyse hinsichtlich der Anforderungen.....	32
4.5.2	Fazit.....	34
4.6	Annotation System for Semantic Web (NCST, Bangalore).....	34
4.6.1	Analyse hinsichtlich der Anforderungen.....	34
4.6.2	Fazit.....	36
4.7	Annotea.....	36
4.7.1	Analyse hinsichtlich der Anforderungen.....	37
4.7.2	Fazit.....	39
4.8	Fazit.....	39
4.9	Quellen.....	39
5	Entwurf.....	41
5.1	Architektur	41
5.1.1	Komponenten	42
5.1.2	Kommunikation der Komponenten	44
5.1.3	Beispielsitzung	44
5.2	Verankerung.....	45
5.2.1	Repräsentation	45
5.2.2	Bezug.....	46
5.2.3	Dokumentbezug.....	48
5.2.4	Verankerungsrichtung	53
5.2.5	Navigationsanker.....	53
5.2.6	Zusammenfassung.....	53
5.3	Datenformat der Annotationen.....	53
5.3.1	Java Serialisierung.....	54
5.3.2	XML	54
5.3.3	Resource Description Framework (RDF).....	54
5.3.4	Fazit.....	55
5.4	Promeladoc	56
5.5	Annotationsart.....	56
5.5.1	Text.....	56
5.5.2	Hervorhebung	57
5.5.3	Verweise	57
5.5.4	Benennungen und Stichwortvorschläge durch den Verfasser	58
5.6	Kollaboration	58

5.6.1	Asynchron.....	58
5.6.2	Synchron.....	59
5.6.3	Offlinebetrieb	60
5.7	Metainformationen.....	60
5.8	Visualisierung	60
5.8.1	Export	61
5.8.2	Konfiguration	61
5.9	Filterung.....	61
5.10	Fazit	61
5.11	Quellen.....	61
6	Bewertung	63
6.1	Analyse hinsichtlich der Anforderungen	63
6.2	Fazit	64
7	Zusammenfassung und Ausblick.....	65
7.1	Zusammenfassung.....	65
7.2	Ausblick	65
7.3	Fazit	66
Anhang A: Projektplan.....		67
A.1	Einleitung.....	67
A.2	Beschreibung der Arbeitspakete	67
A.3	Zeitplan	68
A.4	Dokumente und Meilensteine	69

1 Einleitung

1.1 Motivation

Randnotizen und Unterstreichungen sind schon immer Werkzeuge für die intensive Beschäftigung mit Büchern und Artikeln gewesen. Sie sind dazu geeignet das Verständnis zu erleichtern, sich noch tiefer mit den Inhalten auseinanderzusetzen und auch ein späteres erneutes Befassen zu unterstützen. In vielen Büchern zum Thema Lernen im Allgemeinen und Textverständnis im Besonderen werden ausgefeilte Systeme für das Anbringen von Zusatzinformationen an Texte beschrieben. Es sind schließlich nicht nur einfache Notizen und Hervorhebungen möglich. Auch Bewertungen (wichtig, unklar,...) und Kategorisierungen (Frage, Zusammenfassung) einer Textstelle, oftmals mithilfe von Symbolen, sind wirkungsvolle Hilfsmittel. Auch graphische Elemente, vornehmlich zur Strukturierung eines Textes, werden gerne eingesetzt.

Neben der Papierform entsteht momentan in der Lehre eine neue Art von Lernmaterial mit interaktiven Elementen. Ein Beispiel dafür ist der Einsatz von Simulationssoftware. Zur Übermittlung komplexer Zusammenhänge wird die klassische Vorlesung in Verbindung mit Übungen zunehmend durch solche Hilfsmittel ergänzt. Gerade auch im Bereich der verteilten Systeme in der Informatik kann das Verständnis von Kommunikationsprotokollen durch Simulation und Visualisierung erleichtert werden. Außerdem kann die Motivation der Lernenden durch die Abwechslung, die aus dem Einsatz von Interaktionsmöglichkeiten entsteht, gesteigert werden.

In diesem Zusammenhang entsteht auch der Wunsch, das aus der realen Welt stammende Konzept der Anbringung von Zusatzinformationen in die virtuelle zu übertragen. Im Bereich der Papieranalogien wie Dokumente von Textverarbeitungen oder Präsentationsprogrammen sind Annotationen ein gängiges Funktionsmerkmal. Das Anbringen von Zusatzinformationen für Protokollsimulationen stellt dagegen besondere Anforderungen an ein Annotationssystem.

Daneben ist auch der kollaborative Aspekt von großer Bedeutung. Der Rechner soll die Zusammenarbeit von Dozenten und Studierenden unterstützen. Hierbei sind Erläuterungen, Fragen und Anmerkungen sowie deren Weitergabe relevant. Auch die Art von Zusatzinformationen wie die oben erwähnten aus der nichtvirtuellen Welt bekannten Konzepte müssen möglich sein und können noch durch die erweiterten Möglichkeiten des Rechnereinsatzes wie Audioaufnahmen ergänzt werden.

1.2 Aufgabenstellung

Im Rahmen des Projekts ITO (Information Technology Online) der Universität Stuttgart entsteht in der Abteilung Verteilte Systeme der Fakultät Informatik und Elektrotechnik eine Simulations- und Visualisierungsumgebung für Sicherheitsprotokolle (ProDuctivE). Studierende sollen sowohl im

Rahmen einer konstruktivistischen Selbstlernumgebung als auch in Praktika die Möglichkeit erhalten, sich interaktiv und kollaborativ mit Sicherheitsprotokollen auseinanderzusetzen.

Das Ziel dieser Arbeit ist es ProDuctivE zu erweitern, um die Kopplung von Zusatzinformationen an die Simulation zu ermöglichen. Neben den Besonderheiten der Verankerung bei Protokollsimulationen stehen dabei auch die kollaborativen Möglichkeiten im Mittelpunkt.

Als erstes sind verschiedene Szenarien zur Nutzung von Annotationen zu erstellen und zu analysieren. Dazu gehört zum Beispiel der Einbau von Hintergrundinformationen durch den Dozenten oder das Verfassen und Beantworten von Fragen. Aus diesen sind dann die Anforderungen an das System abzuleiten. Besonderes Augenmerk soll dabei auf die Möglichkeiten der Speicherung und Weitergabe der Annotationen gerichtet werden. Auch Probleme für den kollaborativen Einsatz wie Rechtemanagement sollen analysiert und Lösungsansätze dafür gefunden werden. Grundlegende Funktionen um Studierenden die Möglichkeit zu geben Fragen zu stellen und Anmerkungen zu verfassen sind in Form des Gespanns Donut und ITS (s. 2.7 Donut/ITS) bereits verfügbar. Es ist zu untersuchen, ob diese für die Nutzung privater Annotationen, weiterer Annotationstypen und erweiterter Kollaborationsmöglichkeiten modifiziert werden können. Es ist besonders zu untersuchen, ob eine Erweiterung zur Kopplung von Zusatzinformationen an *Protokollsimulationen* möglich ist.

Im Verlauf der Arbeit soll eine prototypische Implementierung der Lösungsansätze die Systeme ProDuctivE und Donut erweitern.

Der Entwurf sowie vor allem die einzelnen Lösungsansätze sind im Hinblick auf die eingangs entworfenen Anforderungen zu bewerten.

1.3 Überblick

Nach dieser Einleitung beschäftigt sich das zweite Kapitel mit den Grundlagen für diese Arbeit. Neben der Definition einiger fundamentaler Begriffe sind das vor allem die Arbeiten und Projekte, die das Fundament für diese Arbeit bilden. Von elementarer Bedeutung ist dabei natürlich das ProDuctivE-System, das wiederum auf einigen anderen Projekten aufbaut. Daneben sind noch das NUSS-Framework und das Donut-System bzw. ITS (Interaction Tracking System) wesentliche Ausgangspunkte.

Im anschließenden Kapitel werden einige konkrete Szenarien für den Einsatz von Annotationen in Protokollsimulationen sowie die davon abgeleiteten Anforderungen an das System vorgestellt.

Das Kapitel vier behandelt einige Ansätze, die sich ebenfalls mit Annotationen beschäftigen, begutachtet sie hinsichtlich der eruierten Anforderungen und setzt sie in Relation zu dieser Arbeit.

Im zentralen fünften Kapitel wird der Entwurf dieses Annotationssystems, gerade auch im Vergleich mit den bereits vorgestellten Projekten, erläutert. Einerseits stehen dabei verschiedene Lösungsansätze für die unterschiedlichen Probleme und Anforderungen im Vordergrund andererseits sind es natürlich

die Begründungen für die schlussendlich getroffenen Entscheidungen bei der Auswahl zwischen alternativen Handlungsmöglichkeiten, die diesen Teil der Arbeit charakterisieren.

Nach einer Bewertung im sechsten Kapitel, die den Bogen wieder hin zu den anfangs entworfenen Anforderungen spannt, folgen im letzten Kapitel noch die Zusammenfassung sowie ein Ausblick, welche Möglichkeiten in der Zukunft noch evaluiert und realisiert werden können.

2 Grundlagen

Wie in der Einleitung beschrieben geht es in dieser Arbeit um die Erweiterung eines bestehenden Systems: ProDuctivE. Dieses basiert auf anderen Projekten und stellt zusammen mit dem NUSS-Framework und dem Donut/ITS System die Grundlage dieser Arbeit dar. Der Definition der zentralen Begriffe folgt die Vorstellung und Analyse dieser Basisarbeiten.

2.1 Begriffe

Hier werden die zentralen Begriffe kurz aufgeführt:

- **Protokoll**
Ein Protokoll beschreibt die Teilnehmer und deren Verhalten, insbesondere Nachrichtenformate und deren Abfolge, für eine wohldefinierte Kommunikation beispielsweise zur Authentifizierung.
- **Protokollspezifikation**
Die Spezifikation ist das zentrale Lernmaterial: die Beschreibung eines Protokolls mit formalen Methoden.
- **Dokument**
Um einen Bogen zu Annotationssystemen für klassische Dateitypen wie Texten zu schlagen wird der Begriff Dokument für die Datei mit der die Protokollspezifikation verwendet. Diese Datei wird den Benutzern übermittelt. Im Falle von ProDuctivE ist das ein Promela Quelltext (zu Promela s. Kapitel 2.3 Promela).
- **Protokollsimulation**
Die Protokollsimulation ist die zentrale Funktion von ProDuctivE. Auf der Basis einer Protokollspezifikation ist eine Simulation des speziellen Verhaltens möglich.
- **Validation**
Die Validation ist die Überprüfung (durch den Rechner) auf die Einhaltung bestimmter (durch den Menschen) definierter Korrektheitskriterien.
- **Entität**
Entitäten sind die „Teilnehmer“ der Simulation einer Protokollspezifikation. Dazu gehören vor allem die Teilnehmer einer Kommunikation und die ausgetauschten Nachrichten.

- **Ablauf**
Der Ablauf bezieht sich auf die Simulation und beschreibt eine spezielle Ausführung der Spezifikation.
- **Zustand**
Der zentrale Begriff während dem Ablaufen einer Simulation ist der Zustand derselben. Es kann dabei zwischen dem Gesamtzustand der Simulation und einem Teilzustand unterschieden werden. Wird das Verhalten aller Teilnehmer einer Kommunikation durch Automaten beschrieben, entspricht der Gesamtzustand der Menge der aktuellen Zustände aller Teilnehmer. Bei einem Teilzustand werden dementsprechend nur die Zustände eines Teils der Entitäten betrachtet.

2.2 Spin

Spin [HOLZ97] ist ein Werkzeug, das zur Simulation und Validation von Modellen verteilter Systeme (insbesondere Kommunikationsprotokolle) genutzt werden kann. Spin interpretiert dazu in der Sprache Promela (s. 2.3 Promela) verfasste Spezifikationen. Das System kann dabei in zwei Modi arbeiten:

- **Simulation**
Damit kann bei einem selbstentworfenen Protokoll ein erster Eindruck bezüglich des gewünschten Ablaufs gewonnen werden. Bei der Betrachtung von Protokollspezifikationen, die nicht selbst geschrieben wurden, können Simulationen Einblicke in das Verhalten liefern. Die graphische Oberfläche für die Simulation liefert dazu das in TCL/TK geschriebene Programm Xspin.
- **Validation**
Für die Validation generiert Spin aus den Promelaspezifikationen einen endlichen Automaten. Die Sprache Promela ist dabei so aufgebaut, dass alle Komponenten begrenzt sind um die Endlichkeit des resultierenden Automaten zu garantieren. Zur Validation benutzt Spin vom Verfasser der Protokollspezifikation definierte Korrektheitsanforderungen (LTL Formeln), die in ihrer negierten Form als never-claims im Promela-Quelltext angegeben werden. Da die Validation ein äußerst zeitintensiver Prozess ist, generiert Spin dazu aus der Promelaspezifikation einen C-Quelltext, welcher dann normal kompiliert und zur Ausführung gebracht wird. Falls Spin einen Fehler findet, wird dazu ein Trailfile generiert, welches die Ausführungsfolge, die zur Verletzung einer Korrektheitsbedingung geführt hat, aufzeichnet. Dieses Trailfile kann dann im Zusammenspiel mit der Simulationskomponente benutzt werden, um sich den Verlauf, welcher zur Verletzung einer Korrektheitsbedingung geführt hat, vor Augen zu führen.

Die große Stärke von Spin liegt sicherlich in seiner Validationskomponente begründet. Gerade für die Lehre bietet sie einige Vorteile. So ist beispielsweise ein Szenario denkbar, in dem das Grundgerüst einer Spezifikation oder eventuell eine fehlerhafte Spezifikation zusammen mit den notwendigen Korrektheitsbedingungen an die Studierenden ausgegeben wird. Diese ist dann zu komplettieren bzw. richtig zu stellen. Die Korrektheit der Lösung ist dabei leicht sowohl durch den Dozenten als auch durch den Studenten festzustellen. Letzteres bietet sich besonders für eine Selbstlernumgebung an.

2.3 Promela

Die Sprache Promela ist an C angelehnt beinhaltet allerdings Konstrukte um eine Spezifikation von Kommunikationsprotokollen und besonders auch die automatische Validation deren Korrektheit zu ermöglichen.

Da die Eigenschaften der Sprache Promela einen großen Einfluss auf die Möglichkeiten der Verankerung von Zusatzinformationen hat, soll hier etwas ausführlicher auf sie eingegangen werden.

Die zwei wichtigsten Konzepte von Promela sind Nebenläufigkeit und Indeterminismus, was der größte Unterschied zu „normalen“ Programmiersprachen ist.

Folgende Elemente können deklariert werden:

- Integervariablen

Promela unterstützt nur Integervariablen in verschiedenen Ausführungen (Byte, Short, ...)

- Symbolische Konstanten
- Prozesstypen

In diesen wird ähnlich wie in Unterprogrammen die eigentliche Programmlogik bestimmt. Sie können auch Parameter erwarten. Es können dabei mehrere Instanzen eines Prozesstyps während des Programmablaufs gleichzeitig existieren.

Der ausgezeichnete Prozess „init“ wird zu Beginn des Programmablaufs ähnlich einer Main-Funktion aufgerufen und kann dann die eigentlichen Prozesse aus den Prozessbeschreibungen heraus instanziiieren.

- Kanäle

Kanäle dienen zur Kommunikation zwischen Prozessen. Sie werden mit einer maximalen Größe und der Art der zu übermittelnden Nachrichten (ähnlich eines Records) typisiert. Jeder Prozess der einen Kanal kennt, kann über ihn Nachrichten senden und empfangen. Kanalreferenzen können dabei sowohl bei Prozessinstanzierungen als auch als Nachricht über Kanäle übergeben werden.

Kanäle sind also mit Queues vergleichbar.

- Records („typedef“) und Arrays

Die oben erwähnte Nebenläufigkeit wird einfach durch Instanziierung mehrerer Prozesse (eines oder mehrerer Typen) erreicht. Der bei Nebenläufigkeit stets auftretende Wunsch nach Atomizität eines Ausführungsabschnitts wird in Promela durch einen Atomic-Block erfüllt.

Eine Besonderheit bei Promela ist, dass jede Expression als Ersatz eines traditionellen if-Statements dienen kann. Nur wenn die Expression wahr ist, wird der Programmablauf fortgesetzt. Das if-Statement von Promela ist dagegen ein indeterministisches Konstrukt. Es können mehrere Bedingungen angegeben werden. Der Fortgang kann dann nach jeder beliebigen erfüllten Bedingung erfolgen. Promela verfügt natürlich noch über weitere Statements (Schleifen, etc.), auf die hier aber nicht näher eingegangen wird.

Die Mächtigkeit von Promela wird durch den Einsatz des C-Präprozessors noch erhöht. Diese Fähigkeit wird auch besonders von ProDuctivE benutzt.

Der Vorteil von Promela liegt darin, dass die Sprache auf die Spezifikation von Modellen verteilter Systeme und insbesondere Protokolle ausgerichtet ist. Es bietet die dazu nötigen Konstrukte und erleichtert durch eine gewisse Nähe zu C (ohne dessen große Stolperfallen) den Einstieg. Aufgrund der recht großen Verbreitung ist auch eine Vielzahl von fertigen Protokollspezifikationen im Internet verfügbar.

2.4 HiSAP

Das HiSAP Toolkit [PAPESCH03] hat sich die Simulation und Visualisierung von Protokollen zur Aufgabe gemacht. Protokolle werden dabei als Java-Applets implementiert und können die HiSAP Funktionen zur Darstellung nutzen. Für ProDuctivE wird allerdings nur die Visualisierungskomponente verwendet.

Im Folgenden werden die Hauptkonzepte von HiSAP beschrieben:

- Node: Knoten repräsentieren die Endpunkte von Verbindungen (connections). Ein Knoten kann dabei auf mehrere Verbindungen zurückgreifen
- Connection: Eine Connection verbindet genau zwei Nodes.
- Nachricht: Nachrichten sind beliebige Datenpakete, die in Form von Strings über connections ausgetauscht werden können.
- Strategy: Strategien bilden das Protokoll. Sie bestimmen wie Nodes nachrichten senden und empfangen.

Der große Vorteil von HiSAP liegt sicherlich in seinen ansprechenden Animationen. Durch die Tatsache, dass es in Java geschrieben und damit Applets realisiert werden können, sind die Anwendungen auch besonders einfach den Lernenden zugänglich zu machen.

2.5 HiSPIN

HiSPIN [PAPESCH03] ist ein in Java geschriebener Promelainterpreter. Außer der Kompatibilität zu Promela bietet HiSPIN auch die Nutzung von durch die Validationskomponente von Spin generierten Trailfiles. HiSPIN ist dazu ausgelegt als Komponente in einem größeren System und nicht nur als Standalonesoftware eingesetzt zu werden und den Zugriff auf die Eigenschaften einer Promelasimulation zu ermöglichen.

2.6 NUSS

Das NUSS-Framework [NUSS, NUSSVS] hat sich die Entwicklung einer Kommunikationsinfrastruktur für gemeinsame Anwendungsnutzung in der Lehre zum Ziel gesetzt. Die Absicht ist dabei sowohl grundlegende Dienste als auch Anwendungen zur Verfügung zu stellen. Das wesentliche Konzept ist eine Sitzung, die als elektronisches Pendant zu einer Veranstaltung angesehen werden kann. Die wichtigsten Dienste sind dabei:

- NAP

Der NUSS-Accesspoint bietet einen Registryservice an. Es können über ihn Sessions aufgebaut und abgefragt werden. NUSS unterstützende Programme können so dem Benutzer eine Liste mit den verfügbaren Veranstaltungen präsentieren, bzw. eine Art Launcher kann auch bei Anwahl einer Session die dazugehörigen Programme starten.

- Rechtemanagement

Das NUSS-Framework bietet ein flexibles Rechtemanagement. Die Rechte selbst werden dabei von den einzelnen Anwendungen definiert. Es sind dabei auch besonders für die Lehre wichtige Funktionen wie die Möglichkeit der Nachfrage beim Dozenten, ob ein Recht gewährt werden soll, implementiert.

2.7 Donut/ITS

Donut und das ITS (Interaction Tracking System) [NAGY03] entstanden ebenfalls im Rahmen des NUSS-Projekts. Es wurde erstmalig für Shared Powerpoint (s. 4.3 Shared Powerpoint) implementiert. Das Gespann dient vor allem der Unterstützung großer Vorlesungen. Der Dozent versieht dazu das Lehrmaterial mit Stichwörtern. Die Studenten können dann beim Stellen von Fragen oder beim Verfassen von Anmerkungen aus den aktuell gültigen Stichwörtern wählen. Das besondere liegt in der Darstellung für den Dozenten. Dieser erhält eine Zusammenfassung welche Stichwörter in welchem

Bereich des Dokuments besonders häufig angefragt werden. Er kann dann die Themen, für die besonderes Interesse besteht, auswählen.



Abbildung 1: Darstellung von Beiträgen im ITS (Quelle [NAGY03])

Eine Analyse von Donut/ITS ergab, dass eine Erweiterung der bestehenden Fähigkeiten und Anpassung an die Erfordernisse von Protokollsimulationen nur mit großem Aufwand möglich ist. Deshalb wird ein komplett neues System entwickelt, das aber selbstverständlich einige Konzepte nutzt.

2.8 ProDuctivE

ProDuctivE verbindet HiSAP und HiSPIN und bettet es mithilfe des NUSS-Frameworks in eine kollaborative Lernplattform ein. Während anfänglich HiSAP als Visualisierungsmittel für beliebige von HiSPIN simulierte Protokolle gedacht war, zeigte sich, dass dies nur sehr schwierig zu bewerkstelligen ist. Eine Anpassung an unterschiedliche Protokollarten scheint für eine ansprechende Visualisierung unumgänglich. ProDuctivE konzentriert sich dazu auf die Simulation von Sicherheitsprotokollen. Da die einzelnen Komponenten Simulation, Visualisierung und Kollaboration aber von unterschiedlichen Protokollarten unabhängig ist, muss zur Anpassung an eine neue Protokollart nur der „Kleber“ zwischen diesen angepasst werden.

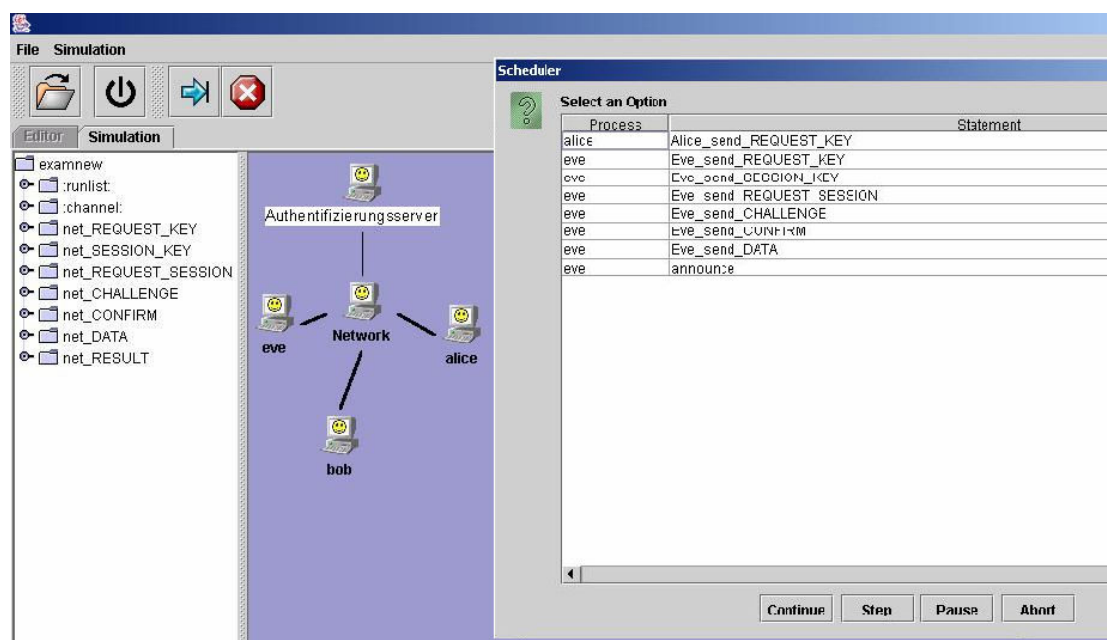


Abbildung 2: ProDuctivE (Quelle: [BURGER03])

Ein weiterer Grund für die Konzentration auf eine Simulationsart liegt in Promela begründet. Die Sprache ist zwar einerseits sehr mächtig und flexibel, stellt für den Einsteiger aber eine hohe Eintrittsschwelle dar.

Durch den Einsatz von Präprozessormakros und der Festlegung auf eine gemeinsame Konvention schafft es ProDuctivE, sicherlich auf Kosten der Flexibilität aber sehr zum Vorteil für den Einsatz in der Lehre, die Komplexität einer Promelaspezifikation vor den Studierenden zu verbergen. Der große Vorteil von Spin, die Validation, wird aber nicht aufgegeben.

ProDuctivE erlaubt es interaktive Sitzung aufzuzeichnen und wieder abzuspielen. Beim Abspielen ist nur ein Vor- und Zurückgehen möglich, während beim Aufzeichnen interaktiv Entscheidungen für die Prozesse getroffen werden können. Allerdings kann während des Abspielens einer Aufzeichnung auch wieder in den interaktiven Modus gewechselt werden.

Eine interessante Funktion von ProDuctivE ist der so genannte „Act-On-Behalf“ Modus. Dabei können Studierende oder auch Tutoren die Rollen von Kommunikationspartnern übernehmen, d.h. sie steuern einen Prozess der Simulation. Bei der Simulation von Sicherheitsprotokollen kann ein Anwender beispielsweise die Rolle des Angreifers übernehmen um zu versuchen eine Schwachstelle des Protokolls aufzudecken.

2.9 Fazit

In diesem Kapitel wurde neben der Definition zentraler Begriffe vor allem die Beschreibung der grundlegenden vorgenommen. ProDuctivE ist das System, dass um die Funktion zur zustandsabhängigen Verankerung von Zusatzinformationen erweitert werden soll. Es nutzt

verschiedene Komponenten und Konzepte. Donut und ITS stellen grundsätzliche Annotationsfunktionen für eine Powerpoint-basierte Anwendung zur Verfügung.

2.10 Quellen

- BURGER03** Burger, Cora; Papesch, Matthias
Eine interaktive und kollaborative Selbstlernumgebung für den Bereich der Sicherheitsprotokolle
Universität Stuttgart, 2003
- HOLZ97** Holzmann, Gerard
The Model Checker Spin
IEEE Transactions On Software Engineering, vol. 23, no. 5, Mai 1997
- NAGY03** Nagy, Michael
Zwischenfragen in Augmented Lectures
Universität Stuttgart, IPVS, Diplomarbeit, 2003
- NOLLER03** Noller, Jochen
Hierarchische Floorcontrol und Filterung von Primitiven - Generische Rechteverwaltung in hierarchischen Gruppen
Universität Stuttgart, IPVS, Studienarbeit, 2003
- NUSS** Universität Stuttgart
Notebook University Stuttgart
<http://www.campus-online.uni-stuttgart.de/nuss/>
- NUSSVS** Universität Stuttgart, IPVS, Abteilung VS
Projekt NUSS
<http://www.informatik.uni-stuttgart.de/ipvr/vs/de/projects/NUSS/>
- PAPESCH03** Papesch, Matthias
A Collaborative Environment for Learning Security Protocols
Universität Stuttgart, IPVS, Diplomarbeit, 2003

3 Szenarien und Anforderungen

Nach der Beschreibung der Basissysteme im vorangehenden Kapitel folgen hier die Definitionen der Phasen bei der Beschäftigung mit Protokollsimulationen, einiger Einsatzszenarien und die daraus abgeleiteten Anforderungen.

3.1 Phasen

Bei der Nutzung von Protokollspezifikation lassen sich drei unterschiedliche Phasen herausarbeiten, die durch ein Annotationssystem zu unterstützen sind.

3.1.1 Erstellungsphase

Die Erstellungsphase ist durch das Verfassen der Protokollspezifikation geprägt. In diesem Zusammenhang ist besonders der Einbau von Erläuterungen wichtig. Aber auch das Stellen von Fragen durch den Dozenten („Was läuft an dieser Stelle schief?“) ist denkbar. Die Erstellungsphase kann sowohl für den Dozenten als auch für die Studierenden relevant sein. So kann eine Übungsaufgabe die Studierenden zum Verfassen einer neuen Protokollspezifikation auffordern.

3.1.2 Diskussionsphase

In der Diskussionsphase wird eine Protokollspezifikation aktiv verwendet. Dabei stehen das Stellen von Fragen (vor allem durch die Studierenden) und das Verfassen von Antworten (Dozent und Studierende) und Anmerkungen im Vordergrund.

3.1.3 Resultatsphase

In der Resultatsphase stehen die Annotationen im Mittelpunkt. Besonders eine Zusammenfassung aller Notizen kann etwa zur Prüfungsvorbereitung nützlich sein.

Für den Dozenten ist es wichtig zu erfahren, an welchen Stellen Probleme für die Studierenden auftraten oder besonderes Interesse existierte. Dafür kann beispielsweise die Anzahl von Fragen ein Indikator sein. Diese Informationen können dann zur Verbesserung der Lernmaterialien verwendet werden.

3.2 Szenarien

Der Einsatz von Annotationen ist in unterschiedlichen Szenarien sinnvoll, die dementsprechend unterschiedliche Anforderungen an das System stellen. Zu beachten ist, dass im Regelfall nicht ein Szenario den Gesamt Ablauf einer Lehrveranstaltung beschreibt. Üblicherweise sind in deren Rahmen alle Szenarien vorhanden.

3.2.1 Gruppenunterricht

Im klassischen Gruppenunterricht führt ein Tutor mehrere Studierende durch eine Übung. Der Veranstaltung geht dabei die Erstellungsphase des Dozenten voraus. Wenn Studierende selbst erarbeitete Lösungen präsentieren ist auch deren Erstellungsphase abgeschlossen. Während der Übung steht die Diskussionsphase im Mittelpunkt. Die Kommunikation findet dabei synchron statt.

Im Anschluss der Veranstaltung folgt die Resultatsphase.

3.2.2 Selbstlernumgebung

Das Projekt ITO hat sich unter anderem die Bereitstellung einer konstruktivistischen Selbstlernumgebung zum Ziel gesetzt. Besonders die Unterstützung virtueller Lerngruppen steht im Vordergrund.

Die Phasen sind wie im Gruppenunterricht vorhanden. Allerdings steht jetzt die asynchrone Kollaboration zwischen Dozent und Studierenden im Vordergrund.

3.2.3 Lerngruppen

Lerngruppen, also Studenten, die sich zum Wissenserwerb zusammenfinden sind durch eine Mischung aus synchroner und asynchroner Kollaboration gekennzeichnet. Während die Studenten untereinander synchron kommunizieren verläuft der Kontakt zum Dozenten asynchron. Es ist aber auch eine Einschaltung des Dozenten in eine laufende Besprechung denkbar.

Auch in diesem Szenario sind wieder alle Phasen präsent.

3.3 Szenarieneinordnung

Die Szenarien lassen sich über folgende zwei Kategorien einordnen:

- **Hierarchie**
Hier ist zwischen dem Vorhandensein einer Hierarchie und deren Fehlen zu unterscheiden. Wenn Tutor und Studenten an einer Sitzung zusammen teilnehmen, sind diese mit unterschiedlichen Rechten ausgestattet.
- **Zeit**
Hier drückt sich die Art der Kommunikation aus. Es wird zwischen synchron und asynchron unterschieden.

Mit dieser Aufteilung ergibt sich folgendes Bild:

Hierarchie			
Tutor/Student	Gruppenunterricht	Selbstlernumgebung	
Keine	Lerngruppe		
	Synchron	Asynchron	Zeit

Tabelle 3-1: Szenarieneinordnung

3.4 Anforderungen

Die Anforderungen an ein Annotationssystem für Protokollsimulationen lassen sich in verschiedene Bereiche gliedern:

- Verankerung
 - Dokumenttyp
 - Dokumentbezug
 - Verankerung von Zusatzinformationen
 - Verankerungsrichtung
 - Navigationsverankerung
- Kollaboration
 - Speicherung der Zusatzinformationen
 - Weitergabe
 - Sichtbarkeit
 - Rechte
- Zusatzinformation
 - Art der Zusatzinformation
 - Metainformationen
- Benutzungsinterface
 - Darstellung
 - Übersicht und Statistiken
 - Bedienung
- Technische Anforderungen

- Generizität, Modularität und Erweiterbarkeit

Bei der folgenden Auflistung wurde eine etwas andere Reihenfolge gewählt, da sich einige Anforderungen erst aus Anderen ergeben.

3.4.1 Dokumenttyp

An erster Stelle steht natürlich die Frage, welcher Dokumenttyp annotiert werden kann. Das zu entwerfende Annotationssystem für dieses Projekt muss natürlich Protokollsimulationen unterstützen.

3.4.2 Speicherung der Zusatzinformationen

Eine Fragestellung ist die Art der Speicherung der Zusatzinformationen. Eine erste Wahl besteht zwischen der Trennung von Annotationen und Protokollspezifikation und deren gemeinsamen, d.h. dokumentinternen, Speicherung. Beide Möglichkeiten haben wie unten beschrieben ihre Vorteile und sind deshalb durch ein Annotationssystem zu unterstützen.

Ein wichtiges Kriterium ist die Möglichkeit für den Benutzer zu einem Dokument Annotationen aus verschiedenen Quellen gleichzeitig zu betrachten. Diese Zusatzinformationen können zum Beispiel von anderen Benutzern, die ebenfalls das Dokument besitzen, übermittelt worden sein.

Dokumentinterne Speicherung

Der größte Vorteil der dokumentinternen Speicherung liegt in einem einfachen Zugang für den Verfasser der Protokollspezifikation, da er direkt in dem für ihn angestammten Modell arbeiten kann. Dieses Verfahren eignet sich damit vor allem für die in Abschnitt 3.1.1 beschriebene Erstellungsphase des Dozenten. Eine Unterstützung durch das Annotationssystem ist somit sehr wünschenswert.

Ist die Spezifikation in Form eines Quelltextes zu verfassen (im Gegensatz zu einem Point & Click Interface) ist eine Einbettung der Zusatzinformationen im Sourcecode in der Art des Literate Programming [KNUTH92] oder von Javadoc [JAVADOC] und POD [POD] möglich und wünschenswert. Besonders Javadoc hat gezeigt, dass Programmierer zu dieser Art der Kommentierung ein besseres Verhältnis haben als zu separaten Dokumentationen. Und die Verankerung von Zusatzinformationen kann als Erweiterung des Konzepts der Kommentierung betrachtet werden. Stimmt das Modell der Visualisierung in der Simulation nicht mit dem Modell der Spezifikation überein (ProDuctivE etwa führt eine Vereinfachung durch), bietet es dem Verfasser den Vorteil in seinem angestammten Modell zu arbeiten.

Separate Speicherung

Eine separate Speicherung stellt eine unabdingbare Voraussetzung für die gleichzeitige Nutzung von Annotationen aus verschiedenen Quellen dar.

Ein weiterer Grund für die separate Speicherung kann die Größe des Dokuments sein. Dies stellt allerdings für dieses Projekt, bei dem die Dokumentgröße aufgrund der benutzten Protokollspezifikationsprache recht klein ist, kein Problem dar.

Auch für die synchrone Weitergabe ist eine separate Speicherung unumgänglich.

3.4.3 Dokumentbezug

Eine unabdingbare Voraussetzung für Zusatzinformationen ist natürlich der Bezug auf ein Dokument (in diesem Fall eine Protokollspezifikation). Zu beachten ist dabei, dass die Spezifikation im Regelfall und auch beim Einsatz von ProDuctivE nur als lokale Datei vorhanden ist und so a priori kein global gültiger Name vorhanden ist. Das stellt zwar kein Hindernis für private Annotationen dar, da der Pfadname als Verankerungsinformation verwendet werden kann. Für öffentliche Annotationen im kollaborativen Einsatz ist eine Übersetzung in eine global gültige Repräsentation nötig. Auch dieses Problem muss von einem Annotationssystem gelöst werden.

3.4.4 Verankerung von Zusatzinformationen

Das zentrale Problem, besonders im Bereich Protokollsimulation, ist auf was sich eine Annotation beziehen kann. Man kann grundsätzlich zwei verschiedene Bezugsarten unterscheiden: Bezug auf Entitäten und Bezug auf Zustände.

Bezug auf Entitäten

Der Bezug auf Entitäten ist der in klassischen Programmiersprachen heute durch Klassendiagramme und besonders im Java-bereich durch Javadoc stark verbreitete Ansatz bei dem Entitäten – Klassen, Methoden, Variablen – eben alles was im Quelltext deklariert wird auch kommentiert werden kann. Im Falle der Protokollsimulation sind es wie oben beschrieben Entitäten wie etwa die Endpunkte einer Kommunikationsverbindung, der Verbindungskanal selbst oder die ausgetauschten Nachrichten, auf die sich eine Annotation beziehen kann.

Bezug auf Zustände

Bei der zustandsabhängigen Verankerung bezieht sich eine Annotation auf den Fortgang eines Programms. Für Protokollspezifikationen ist das der Zustand der Simulation. Zu bemerken ist, dass sowohl der Bezug auf den Gesamtzustand als auch der Bezug auf einen Teilzustand wünschenswert ist. Des Weiteren ist der Bezug auf eine Zustandsfolge denkbar, bei dem nicht ein einzelner Zustand im Mittelpunkt des Interesses steht, sondern das Augenmerk auf einen Teilablauf (beispielsweise der Austausch der zentralen Nachrichten) des Protokolls gerichtet ist. Auch ein gesamter Ablauf kann interessant sein um zum Beispiel zwischen einem regulären (korrekten) Ablauf und einer Ausführung mit einem fehlerhaften Ergebnis zu unterscheiden.

Der Bezug auf Zustände kann noch sehr viel weiter verfolgt werden. Es ist denkbar sich auf den n-ten Ablauf einer Schleife zu beziehen oder sich auf einen Zustand nur dann zu beziehen, wenn ein ausgezeichneter anderer zuvor in diesem Ablauf auftrat.

Für ein sinnvolles Annotationssystem sind aber nicht alle diese Bezugsarten unbedingt notwendig und würden auch den Rahmen dieser Arbeit sprengen.

Orthogonalität

Zustandsabhängiger und entitätsabhängiger Bezug stehen im Wesentlichen orthogonal zu einander. Die Ausnahme besteht in der Existenz der Entitäten, da diese nicht zu jedem Zeitpunkt der Simulation bestehen müssen. Es ist selbstverständlich eine zentrale Anforderung für das Annotationssystem sowohl einen gleichzeitigen Bezug auf Entitäten und Zustände zu ermöglichen. Daneben ist aber auch der Bezug nur auf einen Zustand und nur auf eine oder mehrere Entitäten sinnvoll.

Robustheit

Unter Robustheit ist die Toleranz der Verankerung bei Veränderung des Originaldokuments zu verstehen. Es ist vorteilhaft, wenn einfache Modifikationen, wie im Extermfall nur die Änderung eines Kommentars in einem Quelltext, die Verankerung bestehender Annotationen nicht ungültig macht. Aber die Verankerungen sollen auch weitergehende Abwandlungen überstehen. Dazu gehören beispielsweise das Hinzufügen neuer Validationskriterien oder selbst kleinere Änderungen des eigentlichen Programms.

Die Robustheit kann allerdings auch übertrieben werden, was sogar zu sinnentstellenden Annotationen führen kann, wenn ihre Verankerung plötzlich auf eine gänzlich andere Stelle zeigt. Ein Beispiel dafür findet sich bei der Verwendung von Adobe Acrobat (s. 4.2).

Neben der Möglichkeit überhaupt Änderungen zu tolerieren ist es eine besondere Herausforderung für ein Annotationssystem eine geeignete Balance für die Robustheit im Spannungsfeld von Toleranz und Genauigkeit zu finden.

Allerdings ist für den Lehreinsatz ist die Wichtigkeit der Robustheit nicht überzubewerten, da beispielsweise Protokolldateien für Aufgabenstellungen in der Regel nicht mehr geändert werden. Trailfiles und Ablaufprotokolle sind ohnehin von Änderungen ausgeschlossen.

3.4.5 Weitergabe

Ein weiteres wesentliches Problemfeld für ein Annotationssystem ist die Weitergabe der Zusatzinformationen.

Es gibt die beiden folgenden Ausprägungen:

- Synchroner Weitergabe

Die synchrone Weitergabe, bei der neue Annotationen sofort für alle Beteiligten verfügbar sind, ist für Präsenzveranstaltungen und Lerngruppen unabdingbar.

- Asynchrone Weitergabe

Die asynchrone Weitergabe ist für die Selbstlernumgebung nötig. Aber auch Annotationen die der Dozent während seiner Erstellungsphase für eine Präsenzveranstaltung macht, müssen asynchron weitergegeben werden.

Neben dem zeitlichen Aspekt der Weitergabe ist auch das Mittel für die Weitergabe von Bedeutung. Die Weitergabe der Annotationen als Datei ist vor allem für den Einsatz von E-Mail relevant.

3.4.6 Sichtbarkeit

Für den kollaborativen Einsatz im Allgemeinen und für den Einsatz im Lehrbereich im Speziellen ist eine fein granulierte Steuerung der Sichtbarkeit nötig. Sichtbarkeit bezieht sich dabei allerdings nicht auf die aktuelle Anzeige (was zum Thema Darstellung gehört) sondern ist mit der Frage „welche Annotationen sind in einer kompletten Übersicht vorhanden“ verbunden. Die primären Einstellungen sind:

- Öffentliche Sichtbarkeit

Diese Annotationen sind für alle sichtbar. Hierzu gehören in erster Linie die Erläuterungen des Dozenten.

- Private Sichtbarkeit

Diese Annotationen sind nur für einen selbst sichtbar und sind so vor allem für private Notizen geeignet. Auch der Dozent kann so nur für ihn einsehbare Anmerkungen erstellen.

Aber auch feinere Abstufungen sind nötig. Zu den nächstliegenden Szenarien gehören:

- Nur für einen Studenten und den Dozenten sichtbar (private Frage)
- Nur innerhalb einer Gruppe sichtbar (etwa bei Teamarbeit), evt. zusätzlich für Dozent

Neben der Bestimmung der Sichtbarkeit durch einen Personenkreis sind aber auch weitere Möglichkeiten denkbar. So kann eine Annotation beispielsweise erst bei Erreichen eines Zustands sichtbar werden. Eine häufige Aufgabenstellung an die Studierenden ist das Finden des Fehlers einer Protokollspezifikation. In diesem Zusammenhang kann eine Annotation auftauchen, sobald es den Studierenden gelingt den fehlerhaften Zustand herbeizuführen. Auch eine zeitliche Steuerung ist denkbar. So können den Studierenden, sollten sie die Lösung noch nicht gefunden haben, nach und nach Hinweise gegeben werden.

Die zuletzt genannten Möglichkeiten gehen allerdings über den Rahmen dieser Arbeit hinaus.

3.4.7 Rechte

Neben den Überlegungen zur Sichtbarkeit ist auch eine Rechteverwaltung eine sinnvolle Komponente für ein Annotationssystem. Die zentralen Rechte sind:

- Lesen
Dieses Recht ist zur Steuerung der Sichtbarkeit geeignet.
- Erstellen
- Ändern
- Löschen

Zu beachten ist, dass besonders im Fall einer Hierarchie (s.o.) die Rechte asymmetrisch verteilt sind (dem Dozenten stehen mehr Möglichkeiten offen). Fehlt die Heterogenität der Rollen ist auch das Rechtemanagement von eher untergeordneter Bedeutung. Das einfache Prinzip, dass jeder nur mit seinen Annotationen alles machen kann, reicht in der Regel aus.

Für die erweiterte Unterstützung von Gruppenunterricht ist auch die einfache Änderung bestehender Rechte nötig. Dazu gehört beispielsweise das situationsabhängige Vergeben von Schreibrechten.

3.4.8 Art der Zusatzinformationen

Es sind verschiedene Arten von Zusatzinformationen sinnvoll:

Text

An erster Stelle liegen Textnotizen. Es sollen selbstredend auch Textformatierungen unterstützt werden. Eine besonders schätzenswerte Fähigkeit des Textformats stellen Links dar. Einerseits kann so auf Zusatzmaterial verwiesen werden; andererseits sind aber auch Links auf andere Annotationen oder andere Zustände ein wirkungsvolles Mittel.

Audio

Auch Audioaufnahmen (und für die Zukunft auch Videoaufzeichnungen) sind in der Lehre nahe liegend. Neben der Aufzeichnung einer Gruppenübung kann auch der Verweis auf die relevante Stelle der Vorlesung sinnvoll sein.

Graphik

Bei Textdokumenten sind graphische Annotationen wichtig. Bei Protokollsimulationen ist dagegen schon eine graphische Visualisierung vorhanden. Aber Dinge wie optisches Gruppierungen von Entitäten mit Kreisen oder Ähnlichem können sinnvoll sein, sind aber nicht notwendigerweise als Annotation zu verwirklichen, sondern können in der Regel eher durch die Benutzeroberfläche des

Simulationsprogramms verwirklicht werden. Andere graphische Darstellungen sind durch den Einsatz von Hervorhebungen möglich.

Hervorhebungen

Wie oben bereits erwähnt sind Hervorhebungen ein weiteres probates Mittel zur Unterstützung des Lehrbereichs. Sie können auf die neuralgischen Punkte eines Protokolls hinweisen. Aber auch das virtuelle Zeigen auf eine Entität stellt einen sinnvollen Einsatzzweck dar.

Verweise

Verweise auf externe Materialien sind eine hervorragende Möglichkeit um Hinweise zu weiterführenden Informationen zu geben. Ein interessantes Element sind Audioaufzeichnungen (von Vorlesungen oder Übungen), wie sie auch im Rahmen des NUSS-Projekts angedacht sind. In Zukunft sind sogar Videoaufzeichnungen denkbar. Aktuell sind aber schon Bilder (Diagramme, etc.) ein nahe liegender Wunsch. Für den Lehrbereich stehen vor allem Erläuterungen (beispielsweise Skriptstellen) im Vordergrund.

Eine enge Verknüpfung mit dem restlichen Lehrmaterial erleichtert das Umschalten zwischen traditionellen Vermittlungsmitteln und den neuen interaktiven Möglichkeiten.

Benennungen

Es bietet sich an Zuständen Namen zu geben. Auch dies kann durch eine Annotation erreicht werden. Hat der Dozent den relevanten Zuständen sinnige Namen gegeben, kann das in erster Instanz die Verständlichkeit des Protokolls steigern. Des Weiteren kann dies auch zu einer echten Unterstützung der „Real-Life-Kommunikation“ (Wortmeldungen) in Präsenzveranstaltungen verwendet werden.

Stichwortvorschläge

Stichwortvorschläge hängen eng mit Benennungen zusammen. Diese ermöglichen es dem Dozenten Stichwörter vorzugeben, die von den Studenten mit ihrer Frage oder Anmerkung verknüpft werden. Damit ist es dem Dozenten leichter möglich sich einen Überblick über die von den Studierenden nachgefragten Themen zu verschaffen.

Darüber hinaus können sie auch wieder dazu führen durch ein gemeinsames Vokabular die „Real-Life-Kommunikation“ zu erleichtern.

3.4.9 Metainformationen über Zusatzinformationen

Auch die Annotationen sollten um Metadaten erweitert werden können. Das sind beispielsweise Verfasser und Datum.

Ein ganz wichtiger Punkt sind Kategorien für die Annotation. Dazu gehören beispielsweise Frage, Antwort, Anmerkung, Erläuterung und Exzerpt.

Eine weitere nützliche Sache sind Stichwörter. Neben der Auswahl von Vorgegebenen (s. 3.4.8) ist auch eine freie Eingabe zu erlauben.

Auch die Vergabe eines Titels kann gerade bei umfangreichen Textnotizen einen schnellen Eindruck verschaffen.

3.4.10 Darstellung

Eine besondere Herausforderung bei der Annotation einer Protokollsimulation stellt die geeignete Darstellung der Anmerkungen dar. Die Darstellung hängt natürlich stark von der Visualisierung der Simulation ab. Nahe liegend ist die Nutzung von Tooltips bei Entitäten. Bei Annotationen, die für jeden Zustand gelten, kann das ausreichen. Bei Annotationen, die an bestimmte Zustände, womöglich unabhängig von einer Entität, verankert sind, ist dagegen ein Hinweis auf die Annotation die Minimalvoraussetzung und die direkte Anzeige oftmals wünschenswert. Vorzuziehen ist auf jeden Fall eine konfigurierbare Lösung.

Auch Hervorhebungen und Hinweise auf nicht direkt darstellbare Annotationen wie Audio-Dateien müssen geeignet visualisiert werden. Denkbar sind da vor allem Symbole die auch gleich unterschiedlichen Typs sein können. Auch Farbänderungen oder Blinken kann dafür eingesetzt werden. Auch hier ist es zweckmäßig, wenn sich eine Anwendung an die Bedürfnisse des Benutzers anpassen lässt.

3.4.11 Übersicht und Statistiken

Neben der einfachen Darstellung von Annotationen sind auch Übersichten und Statistiken wichtig. Eine listenartige Übersicht ermöglicht einen Überblick über alle Zusatzinformationen und kann bei entsprechender Beschaffenheit der Notizen auch einen Überblick über das Protokoll liefern. Besonders der Export der Übersicht kann beispielsweise zur Prüfungsvorbereitung genutzt werden.

Statistiken sind wie beschrieben vor allem für den Dozenten interessant. Er kann damit herausfinden wo Informationsdefizite bei den Studenten bestehen. Besonders in der Verbindung mit benennenden Annotationen und Stichwortvorschlägen (s.o.) unterstützt dieses Hilfsmittel den Tutor dabei sich besser auf die Bedürfnisse der Studenten einzustellen.

3.4.12 Verankerungsrichtung

Neben dem „Anbringen“ einer Zusatzinformation an einen Zustand ist auch der Weg zurück aus einer Übersicht sinnvoll. Dabei soll die Simulation in einen Zustand versetzt werden. Dies kann einerseits wie oben erwähnt die umständliche Beschreibung eines Zustands ersetzen. Andererseits wird damit auch die Möglichkeit der Nutzung von Annotationen als Navigationsmittel eröffnet.

Es ist aber eher ein Problem der Simulationsanwendung, ob eine so weitreichende Steuerung überhaupt möglich ist.

3.4.13 Navigationsverankerung

Auch eine Verankerung ohne zugehörige Annotation kann sinnvoll verwendet werden. Sie kann dann beispielsweise zur Übermittlung eines Zustands verwendet werden (s. „Rückkanal“). Es ist dabei wünschenswert eine Ähnlichkeit zu URLs aus dem World Wide Web zu erhalten um den Anwender durch bekannte Konzepte zu unterstützen. Außerdem sind URLs bekanntermaßen bequem zu benutzen und weiterzugeben.

3.4.14 Filterung

Die Möglichkeit der Filterung ist sowohl für die Darstellung als auch für die Übersicht und die Generierung von Statistiken relevant. Vor allem die Metainformationen einer Annotation sind dabei die primären Kandidaten für Filterkriterien. Neben dem klassischen Ausblenden unerwünschter Informationen (vor allem bei der Übersicht) ist auch die Veränderung der Visualisierung wünschenswert. Dinge wie unterschiedliche Farben für unterschiedliche Autoren oder – besonders für den Tutor interessant – für unterschiedliche Kategorien sind definitiv sinnvoll.

Ein weiterer verwandter Punkt ist die Suche nach Annotationen insbesondere auch nach Textnotizen über ihren Inhalt.

3.4.15 Bedienung

Bedienung (und Darstellung) sind das zentrale Bindeglied zwischen dem Annotationssystem und dem Benutzer. Die Anforderungen wie intuitive Bedienung sind zwar bei jedem interaktiven Programm ähnlich. Eine Protokollsimulation stellt aber doch einige besondere Anforderungen an die Bedienung. Die besondere Schwierigkeit liegt bei der Identifizierung und dem Erfassen des gewünschten zu annotierenden Zustands. Dazu ist jedoch hauptsächlich die Simulationssoftware gefragt um durch Interaktionsmöglichkeiten wie einen Einzelschrittmodus und dem Zurücknehmen von Schritten den Anwender dabei zu unterstützen den gewünschten Zustand herbeizuführen.

3.4.16 Generizität, Modularität und Erweiterbarkeit

Generizität und Modularität sind zwei Eigenschaften die als prinzipiell erstrebenswert im Software Engineering angesehen werden. Sie sind aber auch von besonderer Relevanz in Bezug auf ein Annotationssystem. Einerseits ist die Möglichkeit zur Anbringung von Zusatzinformationen für sehr viele Anwendungen (z.B. alle mit Viewer-Charakter) vorteilhaft andererseits stellt sie im Normalfall keine Kernfunktionalität dar und soll deshalb auf minimalinvasive Weise mit der Hauptanwendung gekoppelt werden.

Auch aus den bisher aufgestellten Anforderungen ist ein klarer Vorteil für eine modulare Architektur erkennbar, da die gleiche Funktionalität oftmals in mehreren Ausprägungen notwendig ist. Beispiele

dafür sind der Gegensatz zwischen asynchroner und synchroner Kommunikation oder die Unterstützung von unterschiedlichen Arten von Zusatzinformationen.

Eine einfache Erweiterbarkeit erhöht dazu die Möglichkeit das System für zukünftige Wünsche anpassen zu können.

3.5 Fazit

Aus den Phasen und Szenarien zur Nutzung von Annotationen und Protokollsimulationen wurden in diesem Kapitel die Anforderungen an das Annotationssystem abgeleitet.

3.6 Quellen

KNUTH 92 Knuth, Donald E.
Literate programming
Stanford : CSLI, 1992

JAVADOC Sun Microsystems, Inc.
Javadoc Tool Home Page
<http://java.sun.com/j2se/javadoc/>

POD Wall, Larry
PerlPod
<http://www.perldoc.com/perl5.6.1/pod/perlpod.html>

4 Verwandte Arbeiten

Nachdem im vorangehenden Kapitel die Anforderungen an das zu entwerfende Annotationssystem ermittelt wurden, werden in diesem Kapitel bestehende Projekte und System im Hinblick auf diese Anforderungen analysiert.

4.1 Übersicht

Es wurden folgende Systeme untersucht. Es zeigt sich dabei, dass fast alle zur Nutzung im Bereich Protokollsimulation ungeeignet sind. Deshalb liegt der Fokus vor allem auf Konzepten und Lösungen für Teilbereiche.

- Adobe Acrobat

Adobe Acrobat ist eine bekannte und weit verbreitete Software, die auch für ihre Möglichkeiten der Annotation bekannt ist.

- Shared Powerpoint

Shared Powerpoint (SPP) ist ein Projekt der Abteilung Verteilte Systeme an der Universität Stuttgart im Rahmen des NUSS-Projekts und erweitert die bekannte Microsoft Powerpoint Software um die Möglichkeit des synchronen kollaborativen Einsatzes vor allem für Vorlesungen.

- ns Network Simulator

Der ns Network Simulator ist ein kooperatives Projekt von UC Berkeley, LBL, USC/ISI und Xerox PARC.

- Ilias

Ilias ist eine webbasierte Lernplattform.

- Annotation System for Semantic Web (NCST, Bangalore)

Das „Annotation System for Semantic Web“ ist ein Webannotationssystem.

- Annotea

Ein LEAD (Live Early Adoption and Demonstration) Projekt des W3C Konsortiums. Es ist keine Software sondern ein Protokoll und Datenmodell.

Diese Systeme werden auf die Anforderung aus Kapitel 3 hin überprüft. Auf die technischen Anforderungen wird dabei nur in besonderen Fällen eingegangen. Auf die Verankerung von

Zusatzinformation wird nur im Fall des ns Network Simulators eingegangen, da sie bei den anderen Systemen im Hinblick auf Protokollsimulationen irrelevant sind.

4.2 Analyse Adobe Acrobat

Adobe Acrobat ist ein Programm zur Erstellung und Bearbeitung von PDF-Dokumenten. Es dient als Beispiel für viele Desktopprogramme die Annotationen für traditionelle Dokumente (Text, Graphik, usw.) unterstützen. Betrachtet wurde Adobe Acrobat 5.0.

4.2.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Es werden ausschließlich PDF-Dokumente unterstützt.

Speicherung

Grundsätzlich speichert Acrobat die Zusatzinformationen innerhalb des PDF-Dokuments. Es besteht aber die Möglichkeit die Annotationen zu exportieren und wieder zu importieren. Exportierte Annotationen können direkt geöffnet werden und führen zum Nachladen der annotierten Datei. Es ist somit auch möglich sich Annotationen aus mehreren Quellen gleichzeitig anzusehen.

Dokumentbezug

Der Dokumentbezug wird über den Dateinamen hergestellt. Interessanterweise können Annotationen manchmal (nicht nachvollziehbar) auch in Dateien importiert werden, die nichts mit der Ursprungsdatei zu tun hat. Dies führt zu recht seltsamen Effekten wie verirrt wirkende Hervorhebungsbalken.

Weitergabe

Es wird nur die asynchrone Weitergabe unterstützt. Allerdings kann wie oben beschrieben sowohl das PDF-Dokument inklusive Annotationen als auch nur die Annotationen selbst weitergegeben werden. Beides sind einfache Dateien.

Sichtbarkeit

Die Sichtbarkeit von Annotationen kann nicht gesteuert werden.

Rechte

Es ist kein Rechtemanagement implementiert.

Arten von Zusatzinformationen

Es sind zwei Arten von Textnotizen möglich. Einmal eine Post-It ähnliche Variante und des Weiteren ein Freitext. Beim Freitext sind einige wenige Formatierungsmöglichkeiten (Schriftart und –Größe, Hintergrundfarbe) für die gesamte Annotation möglich. Bei der Post-It Version sind gar keine Einstellungen möglich.

Auch Audio-Dateien können angehängt werden.

Daneben gibt es einige graphische Stilmittel wie einfache Zeichnungsprimitive (Bleistift, Kreis, usw.) und eine Stempelfunktion.

Es sind Hervorhebungen möglich. Dabei wird zwischen Hervorhebungen im engeren Sinne (vergleichbar eines Textmarkers), Unterstreichungen und Durchstreichungen unterschieden.

Es ist außerdem möglich ganze Dateien in das PDF-Dokument einzubetten.

Metainformationen

Verfasser und Datum einer Annotation werden automatisch gespeichert. Zur Filterung können ferner die Art der Zusatzinformation und die Seite, auf die sich eine Annotation bezieht, herangezogen werden. Darüber hinaus sind keine weiteren Metainformationen möglich.

Darstellung

Die Darstellung erfolgt einfach direkt auf dem Dokument.

Übersicht und Statistiken

Es ist möglich eine Zusammenfassung aller Kommentare zu erstellen und als PDF-Datei zu speichern. Eine Sortierung und Filterung ist möglich. Ansonsten ist aber keine Konfigurationsmöglichkeit der Darstellung der Übersicht vorgesehen.

Überdies kann im Bereich der Anzeige für das Inhaltsverzeichnis eines PDF-Dokuments stattdessen eine hierarchische Liste (nach Verfasser, Datum, Typ und Seite) angezeigt werden.

Verankerungsrichtung

Aus der hierarchischen Liste heraus kann navigiert werden.

Navigationsanker

Die Repräsentation einer Verankerung ist für den Benutzer nicht einsichtig. Da der Bezug im Wesentlichen aus einer Seitenzahl besteht, ist dies auch nicht notwendig.

Filterung

Eine Filterung nach Annotationstyp, Verfasser und Erstellungsdatum ist sowohl bei der Darstellung als auch bei der Erstellung einer Übersicht möglich. Für die Navigationsansicht ist allerdings nur eine Sortierung vorgesehen.

Eine Suche nach dem Text einer Annotation ist ebenfalls möglich.

Bedienung

Die Bedienung ist intuitiv aber im Hinblick auf die Besonderheiten von Protokollsimulationen praktisch irrelevant.

Erweiterbarkeit

Die Erweiterbarkeit von Adobe Acrobat gehört eigentlich zu den Stärken des Systems. Es gibt eine Unmenge an Zusatzmodulen und -programmen. Allerdings ist das PDF-Format das unverrückbare Zentrum des Systems. Nur der Umgang *damit* kann erweitert werden.

4.2.2 Fazit

Die besonderen Vorteile von Adobe Acrobat liegen in der intuitiven Bedienung, dem einfachen Generieren von Übersichten und der Benutzung von Sortierungen und Filtern. Auch die Navigationsansicht für Annotationen ist äußerst gelungen.

Die Gestaltungsmittel für Annotationen sind etwas eingeschränkt und orientieren sich stark am Papiervorbild der realen Welt.

Das System ist auf asynchrone Weitergabe beschränkt. Die Möglichkeiten des Imports und Exports von Annotationen bieten eine gewisse Flexibilität wie beispielsweise die wichtige Nutzung mehrerer Quellen gleichzeitig.

Im Hinblick auf die Anforderungen wird vor allem die Kollaboration viel zu wenig unterstützt. Durch die mangelnde Erweiterbarkeit ist auch eine Anpassung zur Unterstützung von Protokollsimulationen unmöglich.

4.3 Shared Powerpoint

Shared Powerpoint entstand während einer Studienarbeit [CERMANN 03] im Rahmen des NUSS-Projekts der Universität Stuttgart. Dabei wurde Microsoft Powerpoint zusammen mit der NUSS-Infrastruktur benutzt und erweitert um Notebook-Vorlesungen zu unterstützen.

Dieses System zeichnet sich im Vergleich zu den anderen besonders durch seine Unterstützung der synchronen Kommunikation und der expliziten Ausrichtung auf die Lehre aus.

4.3.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Es werden ausschließlich Powerpoint Präsentationen unterstützt.

Speicherung

Die Annotationen werden innerhalb der Powerpoint Datei gespeichert und dabei nicht als solche gekennzeichnet.

Dokumentbezug

Ist nicht relevant, da nur interne Speicherung unterstützt wird.

Weitergabe

Das Hauptaugenmerk liegt auf der synchronen Weitergabe während einer Präsenzveranstaltung. Aber es ist natürlich auch möglich die gesamte Powerpoint-Datei inklusive aller Annotationen weiterzugeben. Es ist dann aber nicht möglich Annotationen aus mehreren Quellen gleichzeitig zu betrachten.

Sichtbarkeit

Es werden öffentliche, private und gruppenöffentliche Annotationen unterstützt. Es ist nicht möglich nachzuverfolgen, welcher Beitrag von welcher Person kam.

Es ist möglich die Gruppenzusammensetzungen während einer laufenden Sitzung zu ändern.

Rechte

Das Rechtemanagement erfolgt in Verbindung mit dem NUSS-Framework (s. Kapitel 2.6 NUSS) und ist somit sehr flexibel. Es ist möglich Rechte während einer Sitzung zu vergeben und zu entziehen. Außerdem ist das Zusammenfassen von Rechten zu Rollen möglich.

Arten von Zusatzinformationen

Neben Textannotationen sind einige graphische Primitive möglich. Ähnlich wie bei Adobe Acrobat gibt es für eine Textannotation einige wenige Formatierungsmöglichkeiten, die sich auf die gesamte Annotation beziehen.

Es wird die Hervorhebung in Form eines Zeigers unterstützt. Außerdem ist es möglich einen „temporären Stift“ zu benutzen, dessen Ergebnisse beim Umblättern entfernt werden.

Der Verfasser einer Präsentation kann darüber hinaus weitere Informationen über die Notizfelder zu den Folien anzugeben. Dazu gehört beispielsweise die Zeit, die für diese Folie veranschlagt ist. Während der Präsentation ist es dann für den Dozenten möglich eine Uhr einzublenden um den

Fortgang zu kontrollieren. Diese Informationen werden in Powerpoint eingegeben und von SPP interpretiert.

Eine weitere sehr interessante Möglichkeit ist die Angabe von Stichwörtern für Bereiche einer Folie. Auf diese Stichwörter kann sich der Studierende beziehen, wenn er mithilfe von Donut (s. Kapitel 2.7 Donut/ITS) Fragen und Anmerkungen verfasst.

Im asynchronen Betrieb, in dem die Ergebnisse einer SPP-Sitzung verwendet werden, sind alle Gestaltungsmittel von Powerpoint benutzbar.

Metainformationen

Es werden im Moment keine Metainformationen zu den Annotationen gespeichert.

Darstellung

Die Darstellung erfolgt einfach direkt auf dem Dokument.

Übersicht und Statistiken

SPP unterstützt Donut und das ITS (s. 2.7 Donut/ITS).

Verankerungsrichtung

Da die Annotationen nicht von den Elementen des Powerpoint-Dokuments getrennt sind ist kein Rückkanal möglich.

Navigationsanker

Die Repräsentation einer Verankerung ist für den Benutzer nicht einsichtig. Da der Bezug im Wesentlichen aus einer Seitenzahl besteht, ist dies auch nicht notwendig.

Filterung

Eine Filterung ist nicht vorgesehen.

Bedienung

Die Bedienung ist intuitiv, aber im Hinblick auf die Besonderheiten von Protokollsimulationen praktisch irrelevant.

Erweiterbarkeit

Das Augenmerk der Arbeit liegt auf der Fernsteuerung von Anwendungen. Es wurde untersucht und am Beispiel von Powerpoint implementiert inwiefern Anwendungen „aufgebrochen“ und um kollaborative Funktionen erweitert werden können. Die Erweiterbarkeit des Systems bezieht sich also

auf das Nutzen verschiedener Anwendungen. Zur Nutzung von Annotationen müssten aber die Anwendungen dies bereits unterstützen.

Der Fokus liegt darüber hinaus auf der synchronen Kommunikation.

4.3.2 Fazit

Die große Stärke von Shared Powerpoint aus Sicht eines Annotationssystems ist das ausgefeilte Rechtemanagement. Ein weiteres hervorstechendes Merkmal ist die Nutzung von ITS zu Bereitstellung von Übersichten.

Das Hauptaugenmerk liegt auf der synchronen Weitergabe. Die asynchrone Weitergabe wird nur leidlich und letztlich unter Umgehung des Systems unterstützt. Vor allem fehlt die Möglichkeit Annotationen aus verschiedenen Quellen gleichzeitig zu betrachten.

Auch bei diesem System geht die Erweiterbarkeit in eine falsche Richtung um es für Protokollsimulationen einzusetzen.

4.4 ns Network Simulator

Der ns Network Simulator [NAM] ist wie der Name schon sagt eine Netzwerksimulationssoftware. In Verbindung mit dem Netzwerkanimator Nam wird das System auch gerne in der Lehre eingesetzt.

4.4.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Es werden Simulationen für Protokollspezifikationen in Form von Tcl-Skripten unterstützt.

Speicherung

Nur die dokumentinterne Speicherung im Quelltext wird unterstützt. Die Annotation wird dabei durch einen eigenen Befehl realisiert.

Dokumentbezug

Ist nicht relevant, da nur interne Speicherung unterstützt wird.

Verankerung von Zusatzinformationen

Es gibt nur die Möglichkeit des Bezugs auf die Simulationszeit.

Weitergabe

Nur asynchron zusammen mit Protokollspezifikation möglich.

Sichtbarkeit

Die Steuerung der Sichtbarkeit wird nicht unterstützt.

Rechte

Ein Rechtemanagement wird nicht unterstützt.

Arten von Zusatzinformationen

Nur ein einfacher String ist möglich.

Metainformationen

Metainformationen werden nicht unterstützt.

Darstellung

Die Darstellung erfolgt in einem Unterfenster der Simulationssoftware.

Übersicht und Statistiken

Es werden abhängig von der Fenstergröße die letzten Annotationen angezeigt, aber eine richtige Unterstützung für eine Übersicht ist nicht vorhanden.

Verankerungsrichtung

Es wird kein Rückkanal unterstützt.

Navigationsanker

Die Simulationszeit ist eine einfache Gleitkommazahl.

Filterung

Eine Filterung ist nicht vorgesehen.

Bedienung

Annotationen können nur direkt im Quelltext der Protokollspezifikation eingegeben werden.

Erweiterbarkeit

Eine Erweiterung des Systems um eine umfangreichere Annotationsfunktionalität ist denkbar aber mit sehr hohem Aufwand verbunden.

4.4.2 Fazit

Die einfachen Textausgaben sind sicherlich ein einfacher Ansatz für Simulationen. Die Anforderungen dieser Arbeit sind damit aber nicht zu erfüllen. Eine Erweiterung des Systems ist im Rahmen dieser Arbeit nicht möglich.

4.5 Ilias

Ilias [ILIAS] ist ein Projekt der Universität Köln. Es ist eine webbasierte Lernplattform und bietet das Veröffentlichen von Lernmaterialien über das Web an. Daneben kann zum Beispiel auch ein Quiz realisiert werden. Es bietet für die Studenten ein Kursmanagementsystem sowie Chats, Foren und ein internes Nachrichtensystem.

Ilias ist ein weit verbreitetes System und wird hier als ein Vertreter webbasierter Lernplattformen besprochen.

4.5.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Direkt werden HTML Seiten unterstützt. Es ist aber auch möglich beliebige Dateien als Lernmaterial auf dem Server abzulegen.

Speicherung

Die Speicherung der Annotationen erfolgt ausschließlich auf dem Server.

Dokumentbezug

Der Dokumentbezug wird intern verwaltet.

Weitergabe

Eine Weitergabe von Annotationen ist nicht vorgesehen. Es besteht nur die Möglichkeit die Übersicht auszudrucken oder als HTML-Seite zu speichern und weiterzugeben.

Sichtbarkeit

Es sind nur private Annotationen möglich.

Rechte

Mit den eigenen Annotationen ist alles erlaubt.

Arten von Zusatzinformationen

Es gibt nur einfache Textnotizen.

Metainformationen

Es gibt vier eingestellte Kategorien (Wichtig /Unklar / Gut/ Schlecht) die angewählt werden können. Diese beziehen sich nicht auf die Annotation, sondern auf die annotierte Seite.

Darstellung

Auf der annotierten Seite wird das Vorhandensein nur durch die Abänderung eines kleinen 12x13 Pixel großen Symbols in der Toolbar angezeigt. Wenige horizontale Linien in einem Rechteck sollen ein beschriebenes Blatt suggerieren. Man muss allerdings schon danach Ausschau halten um zu erkennen, dass eine Annotation vorhanden ist.

Die Seite wird während der Änderung der Notiz dargestellt. Die Notiz selbst wird in einem einfachen HTML-Formularfeld angezeigt. Während eine Notiz angezeigt wird, ist allerdings keine Navigation zwischen den Seiten ausführbar. Somit ist es nicht möglich die Annotationen ständig einblenden zu lassen.

Übersicht und Statistiken

Es ist eine nützliche Übersicht über die Annotationen zu einem Kurs verfügbar. Aus dieser kann direkt auf die annotierten Seiten gewechselt werden.

Außerdem gibt es faszinierenderweise an dieser Stelle die Möglichkeit Seiteninhalt und Notiz zusammen anzuzeigen. Dabei werden alle annotierten Seiten in ihrer originalen Reihenfolge zusammen mit der dazugehörigen Notiz angezeigt.

Verankerungsrichtung

s. voriger Abschnitt.

Navigationsanker

Die Repräsentation einer Verankerung ist für den Benutzer nicht einsichtig.

Filterung

Eine Filterung ist nicht vorgesehen.

Bedienung

Die Bedienung ist nicht unbedingt intuitiv. Es irritiert, dass die Notiz nicht zusammen mit der Seite dargestellt wird. Außerdem führt der Text „Notizen“ in der Toolbar zu der Notizübersicht. Ein kleines Icon direkt daneben führt dagegen zur Änderung der Notiz.

4.5.2 Fazit

Die Möglichkeiten des Ilias-Systems im Bereich der Annotationen sind extrem beschränkt. Der größte Mangel liegt im „Verstecken“ einer Annotation hinter einem unauffälligen Icon. Als einigermaßen gelungen kann eigentlich nur die Übersicht bezeichnet werden.

Kollaborative Aspekte werden überhaupt nicht unterstützt. Nicht mal eine einfache Weitergabe von Annotation ist vorgesehen.

4.6 Annotation System for Semantic Web (NCST, Bangalore)

Das „Annotation System for Semantic Web“ [NCST] ist ein Projekt des National Center for Software Technology (NCST) Bangalore in Indien. Das primäre Ziel ist es Annotationen für Webseiten zu ermöglichen. Dazu wurde ein Annotationsserver und ein Internet Explorer Plug-In entwickelt. Es können damit nicht nur Webseiten sondern alle im Internet Explorer darstellbaren Dokumenttypen (PDF, Word, usw.) mit Zusatzinformationen verknüpft werden.

Dieses System ist ein recht ausgefeilter Vertreter für die inzwischen recht zahlreichen Webannotationssysteme.

4.6.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Es werden alle Dokumenttypen unterstützt, die im Internet Explorer dargestellt werden können. Das sind insbesondere Webseiten und PDF-Dateien.

Speicherung

Die Speicherung kann auf einem öffentlichen oder privaten Server oder auf der lokalen Festplatte geschehen. Die Annotationen werden immer separat von der Webseite gespeichert.

Es kann eine Verbindung zu mehreren Servern gleichzeitig aufgebaut werden. Damit ist es möglich Annotationen aus verschiedenen Quellen simultan zu betrachten.

Weitergabe

Wenn die Annotationen auf einem für andere Benutzer zugänglichen Server gespeichert werden, können auch diese auf die Annotationen zugreifen. Eine Weitergabe im engeren Sinn ist nicht möglich, da nur ein systemexternes Benachrichtigen (beispielsweise per E-Mail) eines anderen Anwenders möglich ist.

Es ist keine synchrone Weitergabe vorgesehen.

Sichtbarkeit

Eine Steuerung der Sichtbarkeit wird nicht unterstützt. Es kann allerdings, wenn auch etwas umständlich, durch Einsatz unterschiedlicher Server für unterschiedliche Gruppen und lokaler Speicherung von privaten Annotationen eine begrenzte Steuerung der Sichtbarkeit erreicht werden.

Rechte

Nur der Verfasser einer Annotation hat das Recht diese zu ändern. Ein weiteres Rechtmanagement ist nicht implementiert.

Arten von Zusatzinformationen

Es sind nur einfache textuelle Annotationen vorgesehen. Es wird keinerlei Formatierung unterstützt.

Metainformationen

Es gibt eine Reihe von Metainformationen zu jeder Annotation. Dazu gehören Verfasser und Erstellungs- sowie Änderungsdatum. Aber auch ein Titel und eine Bewertung können vergeben werden. Die Bewertung bezieht sich allerdings nicht auf die Annotation sondern auf die annotierte Web-Seite.

Es gibt zwei Arten von Kategorien. Einerseits kann der Typ der Annotation (Kommentar, usw.) beschrieben werden, andererseits kann auch eine Kategorie mit selbstdefinierten Begriffen eher im Sinne von Stichwörtern gewählt werden.

Darstellung

Die Darstellung einer Annotation erfolgt getrennt vom Dokument in einem eigenen Dialogfenster. Bei Webseiten wird im Dokument durch ein Symbol auf die Existenz einer Annotation hingewiesen.

Übersicht und Statistiken

Es gibt eine gute hierarchische Übersicht geordnet nach Annotationsserver und Webseite. Es sind keine Statistiken verfügbar.

Im Gegensatz zu den anderen Lösungen und auch jenseits der Anforderungen werden hier die Annotationen zu mehreren Dokumenten angezeigt. Dies entspricht sicherlich eher dem Vorgehen im WWW, bei dem in der Regel nicht genau eine Webseite im Mittelpunkt steht. Während zwar auch bei Ilias mehrere Webseiten in der Übersicht auftauchen, so sind diese doch stets Teil eines Kurses. Bei diesem System müssen die Webseiten dagegen in keinem Verhältnis zueinander stehen.

Verankerungsrichtung

Aus der oben genannten Übersicht heraus kann auf die annotierte Web Seite navigiert werden. Allerdings steht hier wie beschrieben nicht der Sprung zu einem Teil innerhalb eines Dokuments sondern zu dem Dokument an sich im Zentrum.

Navigationsanker

Es wird die URL der Web-Seite benutzt und auch angezeigt. Diese kann also ganz einfach benutzt werden. Es ist dazu allerdings nicht die Unterstützung des Annotationssystems nötig.

Filterung

Die Anzeige von Annotationen kann nach verschiedenen Kriterien wie Annotationsserver oder Stichwörter gefiltert werden. Es ist außerdem eine Suche nach bestimmten Annotationen über ihren Inhalt möglich.

Bedienung

Die Bedienung ist intuitiv.

Erweiterbarkeit

Durch die Begrenzung auf den Internet Explorer hält sich die Erweiterbarkeit in engen Grenzen und es resultieren daraus auch unannehmbare Anforderungen für die Einsetzbarkeit des Systems. Leider ist das Kommunikationsprotokoll nicht offen gelegt, so dass auch nicht der Server allein verwendet werden kann.

4.6.2 Fazit

Die Vorteile des „Annotation System for Semantic Web“ liegen in der einfachen Bedienung, den ausgefeilten Filtermöglichkeiten und in der simultanen Unterstützung verschiedener Quellen. Die Gestaltungsmöglichkeiten sind allerdings etwas eingeschränkt.

Die Kollaborationsfähigkeiten lassen vor allem Unterstützung für die synchrone Kommunikation vermissen. Auch das Rechte management und die Steuerungsmöglichkeiten der Sichtbarkeit sind unzureichend.

Die mangelnde Offenheit verhindert auch die Nutzung einer Teilkomponente.

4.7 Annotea

Annotea [ANNOTEa, ANNOPROT] ist kein komplettes Softwaresystem sondern besteht aus einem Kommunikationsprotokoll und einem Datenformat. Es wurde allerdings mit der Zielsetzung entwickelt Teil eines Webannotationssystems zu sein. Es befindet sich noch in der Entwicklung und

besonders über die Art der Verankerung wird gerade diskutiert [ANNODISK]. Bisher liegt das Augenmerk auf XML-Dokumenten, auf deren Elemente per XPointer verwiesen werden kann.

Das Datenformat basiert auf RDF (Resource Description Framework) und bietet so eine große Flexibilität (s. 5.3.3 Resource Description Framework (RDF)).

Das Protokoll basiert auf HTTP. Es werden die gängigen Operationen GET, POST und DELETE umgedeutet. Die Integration in einen bestehenden Webserver ist möglich. Die Server stellen dabei nicht viel mehr als eine RDF-Datenbank zu Verfügung. Nur Annotationsthreads werden gesondert behandelt.

Annotea kann von verschiedenen Anwendungen für verschiedene Dokumenttypen verwendet werden.

4.7.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Grundsätzlich werden keine Einschränkungen für den Dokumenttyp aufgestellt. In der Anfangsphase standen allerdings XML-Dokumente im Mittelpunkt (s. Verankerung).

Speicherung

Es wird nur die separate Speicherung auf einem Server unterstützt. Der gleichzeitigen Benutzung mehrerer Quellserver werden durch das Protokoll keine Steine in den Weg gelegt. Das bleibt allein Sache der Anwendung.

Dokumentbezug

Annotea sieht die Verwendung von URIs (Uniform Resource Identifier) vor. Es liegt aber letztendlich im Ermessen des Anwendungsentwicklers wie er dies handhabt.

Verankerung von Zusatzinformationen

Wie oben bereits erwähnt finden gerade in diesem Bereich zurzeit Diskussionen statt. Vorgesehen sind bisher XPointer Ausdrücke. Auch bei den Serverimplementierern, die nicht Mitglieder des Projektteams sind, gab es zu diesem Thema abweichende Meinungen. Inzwischen werden aber von den Meisten keine besonderen Anforderungen mehr an dieses Merkmal gestellt.

Weitergabe

Das Protokoll stellt ähnlich wie das „Annotation System for Semantic Web“ keine Möglichkeit der Weitergabe per se zur Verfügung. Zwar werden Annotationen auf einem Server gespeichert die Anwender müssen sie aber aktiv von diesem beziehen (pull statt push). Auch hier bleibt nur die Möglichkeit einer systemexternen Benachrichtigung zwischen den einzelnen Anwendern.

Sichtbarkeit

Wird durch die Anwendung bestimmt.

Das Protokoll selbst bietet keine Vorkehrungen.

Rechte

Wird durch die Anwendung bestimmt.

Das Protokoll selbst bietet keine Vorkehrungen.

Arten von Zusatzinformationen

Wird durch die Anwendung bestimmt.

Metainformationen

Wird durch die Anwendung bestimmt. Durch das RDF-Format sind beliebige möglich. Im Schema vorgegeben ist beispielsweise das Element Autor. Das Protokoll sieht eine explizite Möglichkeit der Speicherung von Metainformationen durch den Server vor. Dieser kann etwa Datums- und Verfasserangaben, auch wenn sie vom Client übermittelt werden, überschreiben.

Darstellung

Wird durch die Anwendung bestimmt.

Übersicht und Statistiken

Wird durch die Anwendung bestimmt. Es ist keine serverseitige Funktionalität vorgesehen.

Verankerungsrichtung

Wird durch die Anwendung bestimmt.

Navigationsanker

s. Verankerung

Filterung

Wird durch die Anwendung bestimmt.

Die Autoren des Protokolls führen in einem „informal Appendix“ die Möglichkeit der Benutzung von Algae [ANNOPROT] auf. Algae ist eine allgemeine Abfragesprache für RDF-Dokumente und kann somit zur serverseitigen Filterung eingesetzt werden.

Bedienung

Wird durch die Anwendung bestimmt.

Erweiterbarkeit

Die Erweiterbarkeit ist natürlich der Kern dieses Projekts. Dies ist beim Datenmodell uneingeschränkt von Vorteil. Beim Kommunikationsprotokoll machen sich allerdings der Verzicht auf Authentifizierungs- und Autorisierungsmechanismen auch negativ bemerkbar. Einerseits eröffnet sich dadurch zwar die Möglichkeit die besonderen Umstände des Einsatzgebietes einer Anwendung optimal zu berücksichtigen. Andererseits schadet es der Koexistenz verschiedener Anwendungen und verschiedener Hersteller.

4.7.2 Fazit

Das Annotea-Projekt bietet einen interessanten Ausgangspunkt um ein Annotationssystem zu realisieren. Ein nicht zu unterschätzender Aspekt ist dabei die Beheimatung bei dem W3 Konsortium. So besteht die Chance auf eine Standardisierung und der Unterstützung durch Dritthersteller.

Negativ fällt das Fehlen von Rechte- und Benutzermanagement auf. Allerdings bleibt so eine große Flexibilität erhalten. Leider fehlt auch eine Unterstützung synchroner Kommunikation.

Die Möglichkeiten zur Erweiterung gerade im Bereich der Verankerungsdefinition lassen allerdings die Verwendung als Komponente für ein Annotationssystem für Protokollsimulationen zu.

4.8 Fazit

Die Analyse bestehender Systeme ergab leider keine verwendbaren Lösungen für die zustandsabhängige Verankerung. Es ließ sich aber eine Reihe von Konzepten, vor allem hinsichtlich der Bedienungsschnittstelle und der Nutzung von Übersichten, eruieren. Annotea bietet mit seinem Datenmodell und seinem Kommunikationsprotokoll eine mögliche Basis für die Entwicklung eines Annotationssystems.

4.9 Quellen

- ANNOPROT** Swick, Ralph et al
Annotea Protocols
W3C, 2002
<http://www.w3.org/2002/12/AnnoteaProtocol-20021219>

- ANNOTEA** W3C
Annotea Project Homepage
<http://www.w3.org/2001/Annotea/>
- ILIAS** Universität Köln
Ilias Homepage
<http://www.ilias.uni-koeln.de/ios/>
- NAM** University of Southern Caolina
The Network Simulator - ns-2
<http://java.sun.com/j2se/javadoc/>
- NCST** NCST, Bangalore
Annotation System for Semantic Web Home Page
<http://www.ncb.ernet.in/groups/dake/annotate/index.shtml>
- CERMANN 03** Cermann, Holger
Fernsteuerung von Anwendungen
Universität Stuttgart, IPVS, Studienarbeit, 2003

5 Entwurf

Nach der Vorstellung verschiedener Ansätze für Annotationssysteme im letzten Kapitel geht es in diesem Kapitel nun um den Entwurf dieser Arbeit. Nach einem Überblick über die Gesamtarchitektur folgen die Diskussionen und Entscheidungen zu Lösungsansätzen für die in Kapitel 3 dargestellten Anforderungen.

5.1 Architektur

Wie in der Anforderungsanalyse deutlich geworden ist, stellt die Erweiterbarkeit des Annotationssystems ein wichtiges Ziel dar. Einerseits sind weitere Annotationsinhalte und vor allem Metainformationen denkbar andererseits ist auch der Einsatz dieses Systems für andere Anwendungen, die nicht aus dem Simulationsbereich stammen, interessant. Auch die Anpassung an neue Kommunikationsstrukturen, so wird etwa NUSS noch stark weiterentwickelt, wird durch einen modularen Entwurf erleichtert. Deshalb kommt der Verwendung einer flexiblen Architektur eine besonders große Bedeutung zu.

5.1.1 Komponenten

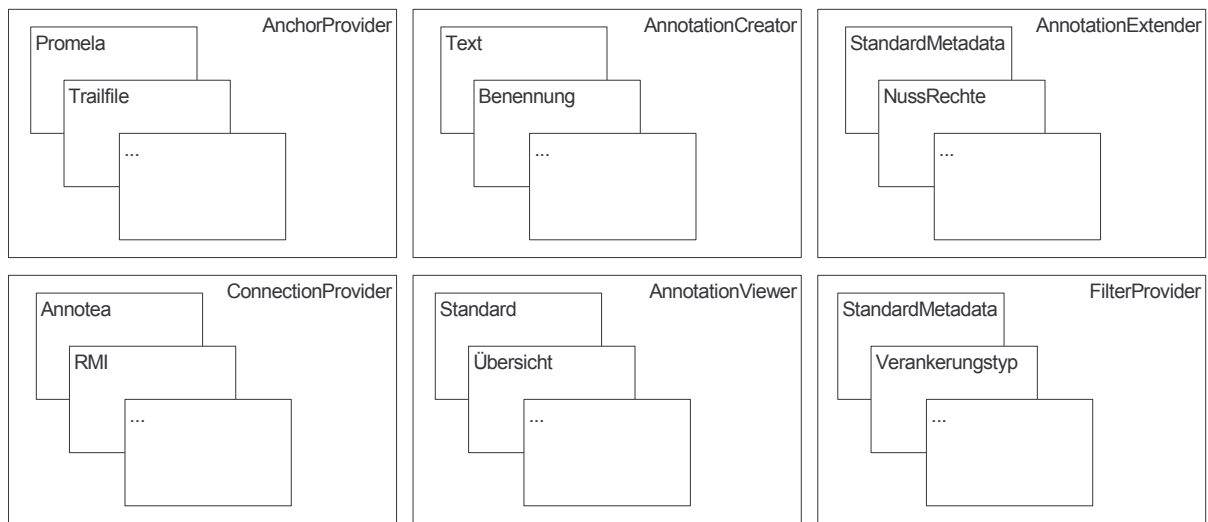
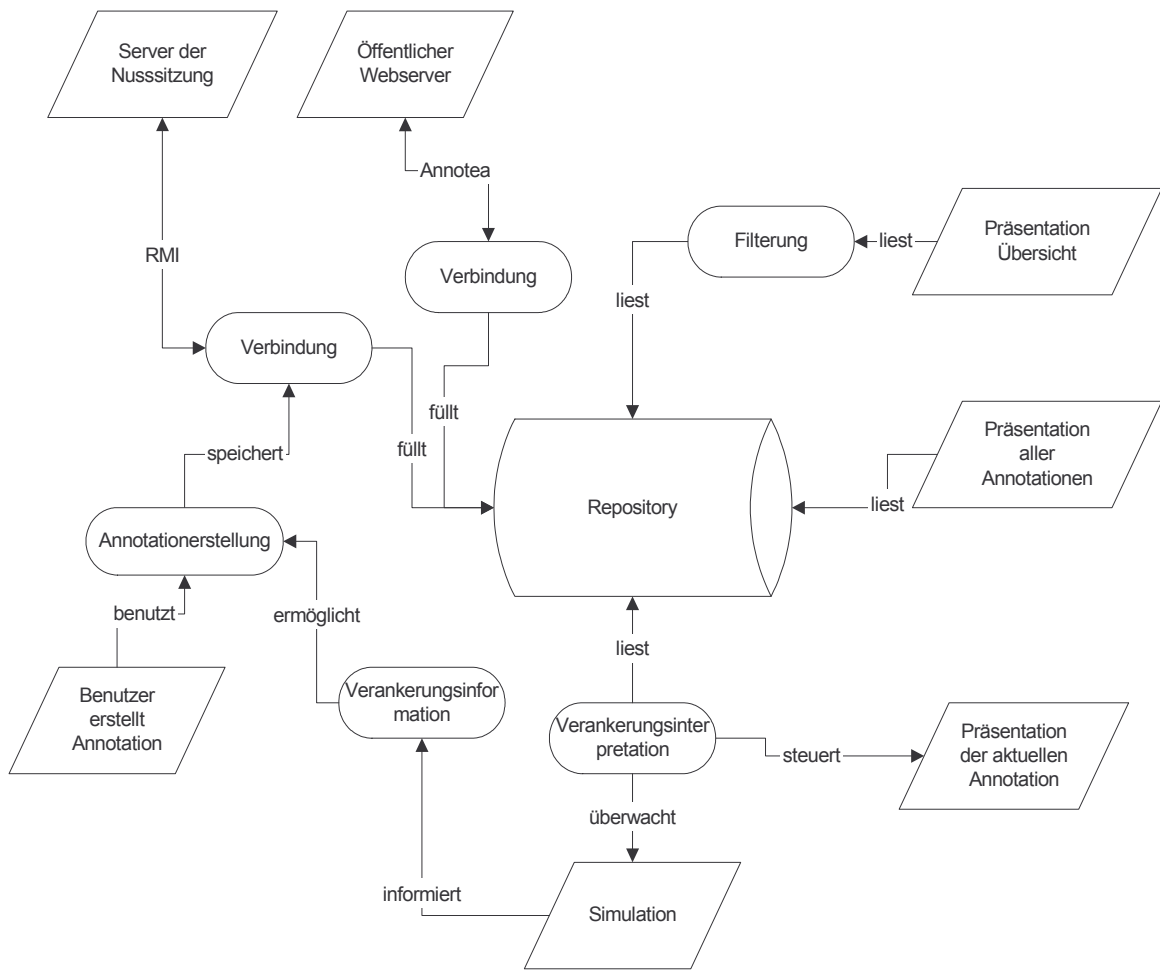


Abbildung 5-1: Architektur

Die Abbildung 5-1 zeigt die Architektur des Annotationssystems. Da es mit den Annotationen ein klares Datenzentrum gibt ist der Repository orientierte [SEGE01] Entwurf sinnvoll. Das System ist in folgende Komponenten unterteilt:

Repository

Im zentralen Repository sind alle Annotationen für die aktuell geladenen Dokumente (Protokollspezifikation, Trailfile, ...) geladen.

Verbindung

Eine Verbindung stellt die Kommunikation zu verschiedenen Quellen bereit. Zu den möglichen Quellen gehören sowohl Annotationsserver, etwa ein Annoteserver, aber auch lokale Dateien. Wie in den Anforderungen erläutert gehört die gleichzeitige Betrachtung von Annotationen aus verschiedenen Quellen zu den elementaren Ansprüchen an das Annotationssystem. Bei der Erstellung einer neuen Annotation kann die Verbindung, über die die Speicherung erfolgen soll, gewählt werden.

Die unterschiedlichen Szenarien stellen außerdem auch verschiedene Anforderungen an die Art der Kommunikation. Dazu gehört vor allem die Unterscheidung zwischen synchroner und asynchroner Kommunikation. Dies wird dadurch Rechnung getragen, dass unterschiedliche Implementierungen („ConnectionProvider“) eingesetzt werden können.

Verankerungsinterpretation

Die Verankerungsinterpretation stellt den Bezug zwischen Simulation und Annotationen her. Diese Komponente überprüft, ob ein annotierter Zustand eingetreten ist und übermittelt diese Information an die Darstellungskomponente.

Verankerungsinformation

Diese Komponente ermöglicht die Angabe der Verankerung. Dazu gehört beispielsweise die Wahl von Entitäten, auf die sich eine Annotation bezieht. Verankerungsinterpretation und –information sind in einer Komponente AnchorProvider zusammengefasst.

Annotationserstellung

Die Annotationserstellung führt mehrere Komponenten in einem gemeinsamen Dialogfenster zusammen:

- AnnotationCreator

Ein AnnotationCreator ist das zentrale Element für das Verfassen einer Annotation. Diese Komponente stellt eine Möglichkeit zur Inhaltseingabe bereit. Im Falle von Textinformationen ist das beispielsweise ein Editierfeld. Bei Hervorhebungen kann die gewünschte Art eingestellt werden.

- AnnotationExtender

Diese dienen zur Erweiterung einer Annotation. In diesen Bereich gehören etwa die Metainformationen wie Kategorien und Stichwörter. Aber auch das Rechtemanagement wird über eine solche Komponente implementiert.

- AnchorProvider

Diese Komponente liefert die möglichen Verankerungen wie die Auswahl einer Entität und natürlich besonders den aktuellen Zustand. Dabei wird auch zwischen dem Bezug auf die Protokollspezifikation, einem etwaigen Trailfile oder einer ProDuctivE-Aufzeichnung unterschieden.

Die resultierende Annotation wird über eine vom Benutzer gewählte Verbindung gespeichert. Diese Verbindung ist für die Aktualisierung des Repositorys zuständig.

Darstellung

Für verschieden Darstellungsarten sind die AnnotationViewer Komponenten zuständig. Zu diesen gehören neben den Gegenstücken zu den AnnotationCreator Komponenten auch die Implementierungen der Übersicht sowie der ITS-ähnlichen Anzeige. Diese greifen dabei auf „normale“ Annotationen bzw. deren Metainformationen zurück.

Filterung

Die Filterung besteht aus zwei Teilen. Einer Benutzungsschnittstelle und der eigentlichen Filterfunktionalität. Für beide Elemente sind die FilterProvider Komponenten zuständig. Das Ergebnis ist eine gefilterte (programminterne) Sicht auf die im Repository verfügbaren Annotationen. Diese kann von anderen Komponenten, vor allem den zur Darstellung dienenden verwendet werden.

5.1.2 Kommunikation der Komponenten

Die Kommunikation der Komponenten erfolgt primär durch Ereignisse. So registrieren sich beispielsweise die Darstellungsmodule bei der Verankerungsinformation um über die aktuell darzustellende Annotation informiert zu werden.

Der zweite wichtige Kommunikationskanal ist die Benutzungsschnittstelle. So stellen beispielsweise AnnotationCreator Komponenten ein Register-Dialogfenster zur Verfügung, das AnnotationExtender Komponenten um eigene Registerkarten erweitern können. Besonders die AnnotationViewer Komponenten integrieren sich auch in die Hostapplikation (ProDuctivE).

5.1.3 Beispielsitzung

Der typische Ablauf für die Nutzung des Annotationssystems beginnt der Erstellung einer Verbindung durch den Benutzer. Die ConnectionProvider Komponente überprüft, welche Dateien (Protokollspezifikation, Trailfile, usw.) von ProDuctivE geladen sind. Daraufhin stellt sie eine

Anfrage an den Server bzw. liest die lokale Datei und füllt das Repository mit den passenden Annotationen.

Während des Ablaufs einer Simulation, übermittelt das ProDuctivE System der Komponente für die Verankerungsinterpretation alle Zustandsänderungen. Da das ProDuctivE bereits das entsprechende Versenden von Ereignissen unterstützt, fügt sich dieser Ansatz leicht in das bestehende System ein. Die Verankerungsinterpretation überprüft welche Annotationen veraltet und welche nun aktuell sind und übermittelt das den Darstellungskomponenten. Diese reagieren entsprechend. Bei der Darstellung einzelner Annotationen wird nur die neue angezeigt, bei der Übersicht wird die aktuell gültige Annotation hervorgehoben.

Wenn der Anwender eine Annotation erstellen möchte, so benutzt er dazu entweder das Hauptmenü der Anwendung oder das Kontextmenü einer Entität. Diese Integration in die Benutzungsoberfläche von ProDuctivE ist ein wesentliches Bindeglied zum Annotationssystem. Der entsprechende Bezug ist dann voreingestellt, kann aber verändert werden. Neben der Wahl der Annotationsart (Text, usw.) und der Angabe des Inhalts kann der Benutzer noch Metainformationen über die Annotation eingeben. Diese werden durch AnnotationExtender Komponenten zur Verfügung gestellt. Der Anwender kann außerdem die gewünschte Zielverbindung für die Speicherung der Annotation bestimmen. Die Verbindungskomponente ist dann für die Übermittlung der neuen Annotation an den Server bzw. der Aktualisierung der lokalen Datei zuständig. Darüber hinaus aktualisiert sie auch das zentrale Repository.

Während einer Sitzung kann der Anwender verschiedene Darstellungsarten (Übersicht, usw.), welche durch AnnotationViewer Komponenten realisiert werden, aktivieren. Diesen kann dabei eine Filterung, deren Einstellung und Ausführung durch FilterProvider-Komponenten realisiert werden, vorangestellt werden.

Mit dem Schließen aller Verbindungen endet die die Ausführung des Annotationssystems.

5.2 Verankerung

Die Fragen rund um die Verankerung stellen das größte Problemfeld dar. Die untersuchten Systeme ergaben keine nennenswerten Ansätze zur Verankerung von Zusatzinformationen in Protokollsimulationen.

5.2.1 Repräsentation

Die Repräsentation einer Verankerung ist vor allem für den Einsatz als Navigationsverankerung (s. Anforderungen) relevant. In diesem Fall ist sie für den Benutzer sichtbar und erfordert damit ein geeignetes Format.

Es bieten sich drei Möglichkeiten an:

- URL
- URL

Mit der Darstellung als URL sind die damit hinlänglich bekannten Vorteile verbunden. Vor allem die einfache Handhabung (z.B. in E-Mails) sprechen für diese Speicherung. Eine kurze Zeichenkette mit einem charakteristischen Erscheinungsbild ist dem Benutzer vertraut und kann einfach eingesetzt werden.

- Opaque

Die Repräsentation hat für den Anwender keinerlei Informationsgehalt. Bei geeigneter Größe und Codierung (z.B. Base64) kann der Anwender den Verweis zwar ebenso in E-Mails einfügen, es fehlt aber der Wiedererkennungswert. Aus technischer Sicht ist dies aber sicherlich der einfachste Weg.

- Mischung

Ähnlich wie bei URLs vieler E-Commerce-Seiten kann der Verweis wie eine URL beginnen und somit den klassischen Wiedererkennungswert besitzen; die wichtigen Daten können aber kodiert angehängt werden. Es ist allerdings die Beschränkung auf eine handhabbare Länge der Zeichenkette notwendig.

Da die „Mischung“ die Vorteile beider Lösungen miteinander verbindet, ist dieser der Vorzug zu geben. Die URL entsteht dabei durch Konkatenation des Dateibezugs (s.5.2.3) und des Bezugs (s. nächster Abschnitt). Eine Klartextrepräsentation wird dabei auch in der Mischung so weit wie möglich verfolgt.

5.2.2 Bezug

Beim Bezug gibt es, wie in den Anforderungen dargestellt, den Bezug auf Entitäten (deklarativ) und auf Zustände (prozedural).

Der deklarative Ansatz stellt dabei das bedeutend kleinere Problem dar. Im Wesentlichen können im Promela-Quelltext vergebene Namen (von Prozessen, Variablen, etc.) verwendet werden. Dass in der Visualisierung durch ProDuctivE eine Abbildung und eine Reduzierung der angezeigten Entitäten vorgenommen wird, ist dabei auch kein Problem, da dann nur unterschiedliche Namensräume für Promela und ProDuctivE Deklarationen verwendet werden müssen. Ein gewisses Problem tritt auf, wenn der Zustand auf die Auswahl von Entitäten Einfluss hat, einfach dadurch, dass mehrere Prozesse desselben Typs existieren. Durch Anhängen eines Array-Indexes bzw. dessen Fehlen kann aber leicht zwischen den beiden Arten unterschieden werden.

Größere Schwierigkeiten bereitet der Bezug auf Zustände. Dabei sind die folgenden beiden Arten zu unterscheiden:

Gesamtzustand und Zustandsfolgen

In diesem Bereich treten die meisten Probleme auf. Die einzige Möglichkeit den Zustand eindeutig zu beschreiben besteht in der kompletten Speicherung des Zustands des Simulators. Dazu gehören bei Promela die Instanzvariablen der Prozesse (auch die Kanäle) und der Instruktionszeiger innerhalb jedes Prozesses. Diese Speicherung wäre zwar möglich, ist aber mit Problemen verbunden. Das Nächstliegende ist der Platzverbrauch. Bei der Beschränkung auf die für ProDuctivE vorgesehenen Sicherheitsprotokolle ist zwar die Menge für heutige Massenspeicher sicherlich kein Problem, aber eine Speicherung als handhabbare Zeichenkette ähnlich einer URL ist damit nicht mehr möglich. Diese aber ist wie beschrieben besonders für die Weitergabe wünschenswert.

Auch für die Geschwindigkeit der Simulation sind beim Einsatz der kompletten Speicherung des Zustand Einbußen zu erwarten, da in jedem Schritt alle Variablen überprüft werden müssen, ob jetzt ein annotierter Zustand (von vielen) erreicht ist. Es sind hier sicherlich Optimierungen möglich, die aber den Simulator stark verkomplizieren würden. Und eine Änderung des Simulators ist wenn möglich zu vermeiden.

Neben der normalen interaktiven Simulation bietet ProDuctivE aber noch zwei weitere Betriebsarten, die den Bezug auf einen Gesamtzustand ermöglichen:

- **Trailfile**

Im Gegensatz zur interaktiven Simulation, die von der Dynamik her einem Programmablauf entspricht ähnelt die Interpretation eines Trailfiles dem Betrachten einer Folienpräsentation. Es ist nur möglich sich schrittweise vor und zurück zu bewegen, bzw. zu einer Stelle zu springen. Der Zustand kann dabei ganz einfach durch die Zeilennummer des Trailfiles repräsentiert werden. Damit ist der Bezug auf einen Gesamtzustand in diesem Betriebsmodus möglich.

- **Speicherung des Ablaufs**

Der Ablauf einer interaktiven Simulation in ProDuctivE kann gespeichert werden. Das kann etwa zur Speicherung einer Lösung für eine Übungsaufgabe („Finden Sie einen fehlerhaften Ablauf“) oder zum Wiederholen einer Präsenzveranstaltung dienen. Die Möglichkeiten bei der Betrachtung dieses Ablaufs entsprechen denen bei der Verwendung eines Trailfiles. In diesem Modus ist der Bezug zu einem Gesamtzustand durch die Verwendung der Schrittnummer möglich.

Teilzustand

Auch der Bezug auf einen Teilzustand (s. 3.4.4), vor allem die Ausführung einer bestimmten Anweisung in einem Prozess, ist unverzichtbar. Besonders das Senden oder Empfangen einer Nachricht ist ein interessanter Bezugspunkt für eine Annotation. Als Erstes kommt einem die Position

im vom Simulator aufgebauten Syntaxbaum als Anker in den Sinn. In einem durchnummerierten Baum kann einfach die Nummer der Anweisung zur Verankerung genutzt werden. Das Hauptproblem besteht dann in der Robustheit der Verankerung. Was passiert, wenn sich der Quelltext ändert? Im Beispiel von HTML ist das kein Problem, da Anker dort benannt sind. Und das ist auch die Lösung für Promela. Promela unterstützt bereits Sprungmarken und diese können damit auch als Anker benutzt werden. Sollte an der Stelle der Anweisung keine Marke stehen, kann immer noch eine Mischung aus dem Bezug auf die letzte vorhergehende Marke und der relativen Position dazu die Robustheit des Ankers steigern. Der Autor des Promela-Quelltextes kann natürlich auch an den neuralgischen Punkten der Spezifikation Sprungmarken definieren.

Eine weitere Steigerung der Robustheit ist mit der Beachtung des Anweisungstyps zu erreichen. Falls statt „n-te Anweisung“ „n-te Anweisung eines Typs“ (z.B. das Versenden einer Nachricht) als Anker benutzt wird, ist besonders bei eher selten verwendeten aber wichtigen und damit auch annotationswerten Anweisungsarten (wiederum der Versand von Nachrichten) eine Erhöhung der Robustheit erreichbar. Diese Maßnahme ist natürlich mit der oben Genannten (Sprungmarken) kombinierbar.

Im Laufe dieser Arbeit hat sich die Spezifikation von ProDuctivE dahin gehend geändert, dass nur noch mit Sprungmarken versehene Anweisungen visualisiert werden. Somit reicht es für das Annotationssystem Zustände mit Marken zu unterstützen. Diese können dann auch einfach in der URL kodiert werden.

Orthogonalität

Da die deklarative Sichtweise im Wesentlichen orthogonal zur prozeduralen (Ausnahmen „Existenz von Entitäten“ s. o.) steht, ist eine kombinierter Bezug auf Entitäten während eines Zustands eine natürliche Verankerungsart. Diese kann durch eine geeignete Kodierung der URL erreicht werden. Analogien sind beispielsweise zu den URLs für das Microsoft Active-Directory oder zu der Kodierung von Formularparametern für HTTP/HTML erkennbar.

5.2.3 Dokumentbezug

Neben dem Bezug auf Stellen innerhalb eines Protokolls ist natürlich der Bezug auf die Protokollspezifikation selbst von Bedeutung. Auch der Bezug auf Trailfiles und ProDuctivE-Aufzeichnungen fällt in diese Kategorie. Das Problem hierbei ist es einen global gültige Referenz herzustellen. Während sich dies bei Web-Annotationsdiensten leicht durch Nutzung der URL der zu annotierenden Web-Seite lösen lässt, ist durch die Verwendung lokal gespeicherter Dateien a priori keine globale Sicht möglich. Da die Dateien auf jedem Rechner in einem anderen Pfad gespeichert sein können, ist eine Instanz zur Abbildung auf einen globalen Namen nötig, über den *alle* Clients übereinstimmen.

Es sind mehrere Lösungen denkbar:

- **Zusatzinformation im Dokument**

In der annotierenden Datei kann ein global gültiger Name gespeichert sein, der als Referenz benutzt wird. Dies kann beispielsweise in einem einer speziellen Konvention folgenden Kommentar geschehen. Die Vorteile liegen klar auf der Hand: Es sind keine zusätzlichen Vorkehrungen bei der Weitergabe zu treffen, da der global gültige Name untrennbar (unabsichtlich) mit der Datei verbunden ist. Außerdem ist der Name gegen Änderungen am Dokument immun.

Die Nachteile sind einerseits eine möglicherweise zu große Robustheit, da so auch trotz großer Änderungen der globale Name derselbe bleibt. Besonders für das übliche copy-and-edit Vorgehen zur Erstellung neuer Dateien ist das problematisch. Der Hauptnachteil liegt aber in der Notwendigkeit der Unterstützung durch das und für das Dateiformat begründet. Einerseits muss der globale Name auf „unschädliche“ Art (etwa in einem Kommentar) in das Dokument einbezogen werden andererseits muss er auch in unterschiedlichen Dateiformaten (Promela-Quelltext, Trailfile, ...) auf unterschiedliche Weise integriert werden.

- **Inhaltsbezogen**

Der globale Name kann auch inhaltsbezogen generiert werden. Dafür werden üblicherweise Hashingverfahren herangezogen. Auch hier gibt es eindeutige Vorteile: Die Vergabe des Namens erfolgt völlig automatisiert, benötigt keine Unterstützung durch das Dateiformat und muss auch nicht separat weitergegeben werden. Der Hauptnachteil liegt in der fehlenden Robustheit. Jede Änderung des Inhalts hat eine Änderung des globalen Namens zur Folge. Dies kann in bestimmten Anwendungsfällen durchaus gewollt sein, bei der Beschäftigung mit Protokollen vor allem in Zusammenarbeit mit anderen ist das aber eher hinderlich.

Eine gewisse Verminderung dieser Anfälligkeit kann durch weitergehende Interpretation des Dokuments erfolgen. So können in einem Quelltext beispielsweise Kommentare aber auch die Namen von Variablen, usw. ignoriert werden. Aber da diese eben nicht den Charakter eines Programms bestimmen, werden sie in der Regel auch nicht (unabhängig vom Rest) geändert.

Es sind auch noch viel aufwendigere Verfahren denkbar: Neben statistischen Analysen kann auch versucht werden den Charakter eines Programms zu erkennen. Dies dürfte aber im Moment noch über die technologischen Möglichkeiten hinausgehen. Ganz sicher liegt es jenseits des Augenmerks dieser Arbeit.

- **Bezug auf Metainformationen**

Der Name kann auch aus Metainformationen der Protokolldatei generiert werden. Da aber im normalen Dateisystem plattformübergreifend nur Dateiname, Datum (u.U. mehrere Daten) und Größe

dafür in Frage kommen, ist dies keine praktikable Möglichkeit. Das Datum wandelt sich häufig bei einer Weitergabe, der Name allein ist nicht ausreichend und die Größe ändert sich in der Regel bei Modifikationen.

- **Globaler Name kodiert im lokalen Dateinamen**

Eine Möglichkeit ist die Kodierung des globalen Namens im lokalen Dateinamen. Aufgrund der vor allem auf der Windowsplattform nur eingeschränkten Möglichkeiten der Namensvergabe und der Unhandlichkeit bei der Verwendung eines dann in der Regel recht langen Namens, scheint dieses Vorgehen aber unpraktikabel. Vorteile sind die einfache Weitergabe und die Unabhängigkeit vom Dateiformat.

- **Separater Stream**

Eine zweite Möglichkeit besteht in der Speicherung in einem separaten Stream. Dieses Verfahren ist bei MacOS gang und gäbe (es gibt neben dem Inhaltsstream noch einen Resourcefork) und auch NTFS von Windows unterstützt mehrere Streams. Es ist aber immer noch ein so exotisches und weitgehend unbekanntes sowie nur bedingt portables Feature, dass von der Verwendung abzusehen ist.

- **Containerdatei**

Eine weitere Möglichkeit bietet das Verpacken des eigentlichen Dokuments in eine Containerdatei, in der dann weitergehende Informationen gespeichert sind. Da Kompressionsformate wie das beinahe allgegenwärtige Zip-Format in der Regel Kommentare unterstützen, die, wieder per Konvention, zur Speicherung von Metainformationen wie eines global gültigen Namens benutzt werden können, bieten sie sich geradezu an. Das Problem ist aber die fehlende Unterstützung durch die Hostapplikation (in diesem Fall ProDuctivE), die Containerformate als solche erkennen und verarbeiten können muss. Für Modifikationen an einzelnen Dateien, sind Kompressionsformate außerdem nicht vorgesehen und lassen sich nur mit Effizienzeinbußen hinsichtlich der Geschwindigkeit und/oder des Platzverbrauchs dafür nutzen. Vorteile sind wieder einfache Weitergabe und Robustheit. Ein Nachteil ist die etwas komplizierte Erstellung.

- **Mappingdatei**

Die eigentlich nächstliegende Möglichkeit besteht in der Speicherung des globalen Namens in einer externen Datei. Diese bildet dann den lokalen, relativen Dateinamen auf den globalen Namen ab. Dabei kann nicht nur der Namen für ein Dokument, sondern eine Abbildung für mehrere Namen in einer einzigen Mappingdatei erfolgen. Es ist auch das Mapping ganzer Verzeichnisse oder verallgemeinert der Einsatz von Platzhaltern bis hin zu regulären Ausdrücken vorstellbar.

Ein großer Nachteil liegt in der aufwendigen Weitergabe, da die Zusatzdatei immer mitwandern muss. Dies kann zum Beispiel durch Verwenden eines Kompressionsformats erreicht werden. Da außerdem

im Lehreinsatz sowieso mehrere Dateien (mehrere Aufgaben, Zusatzdateien wie Erläuterungen, etc.) weitergegeben werden, ist der Einsatz einer Containerdatei ohnehin sinnvoll. Anhand einer Konvention für die Dateinamensvergabe der Mappingdatei kann diese auch ohne explizite Anwenderintervention automatisch erkannt und verwendet werden.

Das Mapping von Verzeichnissen (auch des aktuellen) ermöglicht das einfache Erstellen von neuen Dateien, da diese automatisch, unter Verwendung des globalen Namens für das Verzeichnis und dem Anhängen des Dateinamens, einen globalen Namen erhalten. Ist dieses nicht gewünscht (Studentennamensraum ungleich Dozentennamensraum) kann auch ein explizites Dateimapping verwendet werden. Der große Vorteil gegenüber dem Containerdateiansatz liegt im Verzicht auf die Anpassung der Applikation zur Verwendung eines Containerformats. Dies hat einen weiteren Vorteil: Während die eigene Hostapplikation angepasst werden kann, ist dies bei fremden Anwendungen häufig nicht oder nur mit großem Aufwand möglich. Da aber ein Wunsch für Annotation der Verweis auf weitere externe Dateien (Webseiten, Audiodateien, etc.) ist, die auch online verfügbar sein sollten und von fremden Anwendungen dargestellt werden, können diese zusammen mit der Protokolldatei verteilt und so auch automatisch offline verwendet werden. Da diese Studienarbeit auch in Verbindung mit dem NUSS-Projekt steht, ist die Benutzung des Notebooks an der Universität an einer „Datentankstelle“ bei der neben der Protokolldatei beispielsweise auch Audiokommentare übertragen werden könnten denkbar. Auf diese kann dann, unabhängig davon ob sie lokal verfügbar sind, durch Online-URLs verwiesen werden. Das Annotationssystem kann dann bei einem Verweis durch eine URL automatisch eine lokal verfügbare Kopie verwenden.

Uniform Resource Identifier (URI)

Neben dem Verfahren zur Abbildung eines lokalen Namens auf einen globalen ist auch über das Format des globalen Namens zu entscheiden. Zwar sind letztendlich beliebige Zeichenketten möglich, aber es bietet sich der Einsatz von URIs [URI] an.

URIs werden vom W3C als kurze Strings definiert, die Namen oder Adressen von Ressourcen beschreiben. Die allseits bekannten URLs sind eine Spezialform davon, die sich auf einige populäre Namensräumen der URIs (http, ftp, ...) beschränken und deren Fokus klar auf der Adressierung (im Ggs. zur Benennung) von Ressourcen liegt. Das wichtigste Konzept ist, dass URIs keine Adressangaben für den Rechner darstellen. Mit URIs beschriebene Ressourcen können, müssen aber nicht über das Internet verfügbar sein. Und URIs können, müssen aber nicht dem Rechner helfen ein Objekt zu finden. Ein gängiges Beispiel für den Einsatz von URIs ist die Definition von Namensräumen in XML-Dokumenten.

Zur Bildung eines globalen Namens für ein Dokument bietet sich die Verwendung einer URL an, unter der die Datei online verfügbar ist bzw. der Vorgabe eines Namens evt. Unter zu Hilfenahme einer Teambezeichnung für studentische Arbeiten. Das Annotationssystem selbst unterliegt keinen Einschränkungen.

Fazit

Die Wichtigkeit der Robustheit ist wie in den Anforderungen erwähnt nicht überzubewerten. Von großer Bedeutung ist eine einfache Weitergabe, da diese auf jeden Fall für die Lehre notwendig ist. Eine einfache Erstellung ist sicherlich wünschenswert. Kleine Einschränkungen sind aber beim Ungleichgewicht zwischen der Anzahl der Autoren und der Anzahl der Benutzer zu verschmerzen. Selbstverständlich dürfen diese Nachteile nur minimal sein, um die Einsatzbereitschaft nicht zu beeinträchtigen. Die Möglichkeit für die Einbeziehung von Dokumenten für Fremdapplikationen ist ein gewichtiger Vorteil.

Die folgende Tabelle fasst die Vor- und Nachteile der einzelnen Verfahren nochmals zusammen:

Bewertung: ++, +, 0, -, --	Weitergabe	Robustheit	Komfort Autor	Flexibilität Autor	Handhabung Benutzer	Unterstützung Fremdapplikationen	Realisierung
Zusatzinformation im Dokument	++	+	0	+	+	--	+
Inhaltsbezogen	++	-	++	--	++	--	++
Bezug auf Metainformation	++	--	++	--	++	--	+
Globaler Name kodiert als lokaler	+	+	--	+	--	--	-
Separater Stream	--	+	0	+	+	--	--
Containerdatei	++	+	+	+	++	--	-
Mappingdatei + Containerdatei	++	+	+	++	+	++	+

Tabelle 5-1: Bewertung von Ansätzen für den Dokumentbezug

Die Vorteile der Kombination von Mapping- und Containerdatei geben den Ausschlag für die Wahl dieses Ansatzes.

5.2.4 Verankerungsrichtung

Wie in den Anforderungen beschrieben ist ein „Rückkanal“, der die Simulation an eine gewünschten Stelle bringt wünschenswert. Bei der Nutzung von Trail-Files und der Ablaufspeicherung stellt das kein Problem dar, da ProDuctivE ein vor- und zurückgehen in der Sequenz unterstützt. Bei dem Bezug auf einen Teilzustand ist dies jedoch nicht möglich, da keine Eindeutigkeit bezogen auf den Zustand des Simulators gegeben ist. Die Möglichkeiten des Rückkanals sind aber als so erstrebenswert (s.o.) anzusehen, dass eine inkonsistente Bedienung, bei der nicht von jeder Annotation ein Rückkanal möglich ist, das bedeutend geringere Übel ist.

5.2.5 Navigationsanker

Es besteht die Möglichkeit des Navigierens per Verankerungs-URL. Dies ist allerdings nur mit dem Bezug auf Zustände während der Verarbeitung von Trailfiles oder ProDuctivE Aufzeichnungen möglich.

5.2.6 Zusammenfassung

- Die Verankerung wird in einem URL-ähnlichen Format gespeichert
- Es kann auf Entitäten und mit Einschränkungen auch auf Zustände Bezug genommen werden.
- Eine kombinierte Referenz auf eine Entität, während eines Simulationszustands, ist möglich.
- Ein global gültiger Name für ein Dokument (Protokollspezifikation, Trailfile, usw.) wird durch eine Mappingdatei erreicht. Dateien und Verzeichnissen können globale Namen zugeordnet werden.
- Das Mapping wird bei Verweisen auf externe Dokumente beachtet.
- Ein „Rückkanal“ wird unter Verwendung der entsprechenden Betriebsmodi von ProDuctivE, bei denen ein Vor- und Zurückspringen möglich ist, bereitgestellt.

5.3 Datenformat der Annotationen

Das Datenformat des Annotationssystems spielt eine zentrale Rolle. Es bildet die Grundlage für die Speicherung und die Weitergabe von Annotationen. Andererseits kann es auch die Repräsentation im Speicher zur Laufzeit beeinflussen.

Der Flexibilität des Datenformats kommt eine besondere Bedeutung zu. Einerseits müssen vor allem für die Speicherung unterschiedlicher Annotationsarten beliebige Daten (d.h. auch binäre) unterstützt werden andererseits muss es möglich sein beliebige Metainformationen zu einer Annotation unterzubringen. Diese sollen auf eine möglichst „natürliche“ Art in das Dateiformat integrierbar sein.

Auch unterschiedliche Anwendungsversionen müssen problemlos koexistieren können. Besonders für den Server des Annotationssystems sind nicht alle Änderungen des Annotationsformats relevant. Deshalb darf eine Modifikation des Annotationsformats nicht zum Zwang der Anpassung aller Komponenten führen.

5.3.1 Java Serialisierung

Die Nutzung der von Java gebotenen Serialisierung hat vor allem den Vorteil des einfachen Einsatzes, da nur eine Klasse für die Repräsentation einer Annotation im Speicher erstellt werden muss und diese automatisch persistent gemacht werden kann. Das Problem liegt in der Inkompatibilität zwischen unterschiedlichen Versionen. Das verbietet den Einsatz dieser Methode.

5.3.2 XML

Ein großer Vorteil von XML liegt in der Standardisierung und großen Verbreitung. Dies führt zu einer großen Anzahl verfügbarer Werkzeuge und Bibliotheken.

Hinsichtlich der Flexibilität gibt es zweierlei zu beobachten. Einerseits ist besonders durch die Unterstützung von baumartigen Strukturen das XML-Datenformat sehr mächtig, andererseits ist eine Erweiterung eines bereits definierten Schemas nicht einfach. Besonders Wünsche wie beispielsweise das Erweitern einer Annotation um weitere Metainformationen, die direkt durch XML-Tags repräsentiert werden sollen, sind nicht möglich.

5.3.3 Resource Description Framework (RDF)

RDF

RDF wurde mit der Zielsetzung eines einfachen Datenformats zur Speicherung von Metainformationen für Ressourcen entwickelt.



Abbildung 5-2: Einfaches RDF-Tripel

Die Grundlage des Datenmodells von RDF sind Tripel und URIs (s. 5.2.3). Ein Subjekt ist über ein Prädikat mit einem Objekt verbunden. Alle drei Elemente sind durch URIs beschriebene Ressourcen. Nur das Objekt kann auch ein Literal sein, etwa ein Datum oder, für diese Arbeit, der Inhalt einer

Annotation. Abbildung 5-2 zeigt ein einfaches Beispiel, welches den Autor einer Webseite durch eine Mailadresse beschreibt. „<http://purl.org/DC/Creator>“ ist dabei ein vom Dublin Core [DC] definiertes Prädikat, welches den Verfasser einer Ressource definiert. Von den Subjekten und Objekten können weitere Beziehungen ausgehen. Alle Tripels zusammen bilden einen (gerichteten) Graphen (s. Abbildung 5-3). In diesem Beispiel sieht man auch, dass ein Prädikat durchaus mehrfach ein Subjekt mit Objekten verbinden kann.

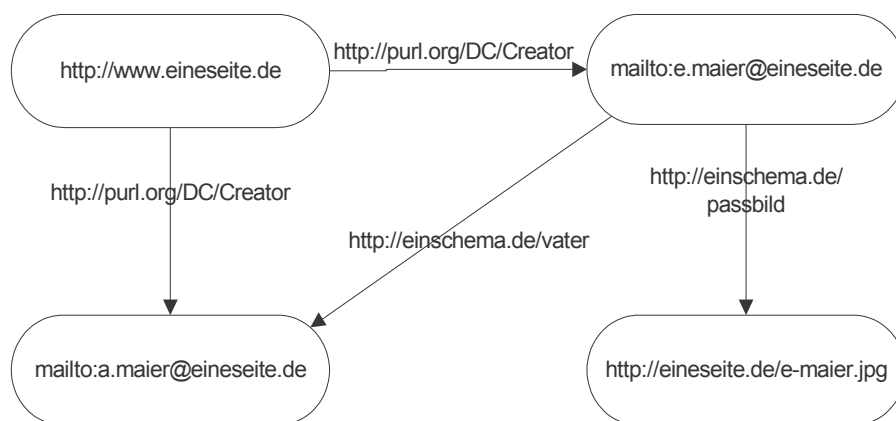


Abbildung 5-3: RDF-Graph

Die Repräsentation eines (oder mehrerer) RDF-Graphen in einer XML-Datei ist standardisiert.

Dieses Datenmodell zeichnet sich also vor allem durch seine große Flexibilität aus. Einerseits ist der Graph ein bekanntermaßen mächtiges Konzept, andererseits ermöglicht die Verwendung von URIs die Koexistenz verschiedener Definitionen ohne auf eine übergeordnete Schiedsstelle angewiesen zu sein. Ein Graph zu einem Subjekt, welches im Rahmen eines Annotationssystems dem annotierten Dokument entspricht, kann von verschiedenen Anwendungen gelesen und geschrieben werden. Dass dabei jede Anwendung nur einen Teil des Graphen interpretieren kann ist grundsätzlich unerheblich.

Es gilt dabei aber nicht zu vergessen, dass nur eine Standardisierung der Prädikate dazu führt, dass eine Anwendung die Daten einer anderen Anwendung auch versteht. In diesem Bereich ist vor allem der schon erwähnte Dublin Core [DC] relevant, da dieses Projekt viele Prädikate, die besonders die Domäne von Bibliotheken berühren, definiert hat. RDF selbst ist ein Standard des W3C und hat schon eine recht ansehnliche Verbreitung erreicht, was sich auch die Verfügbarkeit von vielen Werkzeugen und Programmierbibliotheken ausdrückt.

Da RDF XML als Dateiformat benutzt erbt es seine Vorteile. Auch die Standardisierung durch das W3C und seine Verbreitung sprechen für dieses Modell. Vor allem ist es aber die Flexibilität, die für den Einsatz in einem Annotationssystem sprechen.

5.3.4 Fazit

Die Entscheidung für das Datenmodell ist klar auf RDF gefallen.

RDF überzeugt besonders durch seine Flexibilität. Ein weiterer Vorteil ist, dass auch Annotea dieses Datenformat einsetzt und deshalb viele benötigte Prädikate bereits definiert sind. Außerdem ermöglicht es die Nutzung derselben Infrastruktur. Auch die Standards des Dublin Core sind für ein Annotationssystem nutzbar.

5.4 Promeladoc

Das Annotationssystem unterstützt die Eingabe von einfachen Textnotizen direkt im Promela-Quelltext. Es wird dafür eine Javadoc ähnliche Konvention benutzt. Der Parser von ProDuctivE ist bereits für das Senden von Ereignissen entworfen, so dass eine ConnectionProvider Komponente diese Empfangen und verarbeiten kann. Es ist nur eine kleine Modifikation des Scanners und des Parsers nötig.

5.5 Annotationsart

Bei der Annotationsart ist eine möglichst weitreichende Erweiterbarkeit zu erreichen, da nicht von vorneherein alle möglichen Inhalte absehbar und vor allem in einen Prototyp realisierbar sind. Folgende Formen sind vorgesehen:

5.5.1 Text

Der erste Gedanke zu Annotationsinhalten dreht sich beinahe unweigerlich um Text. Das entspricht auch dem Analogon in der materiellen Welt, wo Kommentare in Büchern eine der häufigsten Annotationsarten sind.

Die Möglichkeiten der Textdarstellung sind mannigfaltig. HTML hat sich allerdings als so etwas wie die Lingua Franca des Computerbereichs etabliert. Einige der Vorteile für den Einsatz in diesem Projekt sind:

- Weit verbreitet (viele Anwender, viele Tools)
- Quellcodekenntnisse weit verbreitet (einfach)
- Reduzierter Tag-Satz (vgl. Javadoc) sehr einfach für den Verfasser handhabbar
- Verweise auf externe Dokumente möglich
- Reiner Text (nicht binär)
- Technisch: Vorhandene Komponenten

Eine weitere Alternative ist RTF (Rich Text Format), welches aber weniger weit verbreitet und praktisch nicht im Quellcode verfasst werden kann.

Es gibt außerdem eine Vielzahl einfacher Formatierungssprachen, die zum Beispiel in Content Management Systemen verwendet werden. Eine recht verbreitete ist die Sprache der Wikis, bei der sich aber auch schon unterschiedliche Dialekte herausgebildet haben. Die meisten haben den Vorteil etwas weniger „Spitze-Klammer“-zentriert und damit weniger kryptisch und dafür schreibfreundlicher zu sein. Sie leiden aber vor allem an der mangelnden Verbreitung (im Vergleich mit HTML) und bieten in der Regel nicht die Flexibilität von HTML. Da aber die Flexibilität von HTML in diesem besonderen Einsatzgebiet (in der Regel kurze einfache Sätze) nicht erforderlich ist, ist der Einbau einer Komponente in das Annotationssystem für eine (oder mehrere) solcher Sprachen ein lohnendes Betätigungsfeld für etwaige Erweiterungen. Da die Darstellungsfähigkeiten von HTML in der Regel weit darüber hinausgehen, ist auch keine Visualisierungskomponente, sondern nur ein Interpretations- und Konvertierungsmodul nötig.

Es wird folglich die Verwendung eines vereinfachten HTMLs vergleichbar mit Javadoc, bei dem auch auf die Nutzung von HTML-Tags völlig verzichtet werden kann, als Textformat verwendet.

5.5.2 Hervorhebung

Neben Textnotizen nehmen Hervorhebungen den größten Stellenwert für Annotation ein. In diesem Projekt stehen zwar sicherlich eher Texte, sei es für Erläuterungen, Fragen oder Antworten im Mittelpunkt. Markierungen und Hervorhebungen sind sicherlich ein ebenso nützliches Annotationskonstrukt. Überdies können Hervorhebungen auch Texte in besonderem Maß unterstreichen.

Als Basis soll dazu ein „Highlight“-Kommando dienen, das schlicht eine Hervorhebung herbeiführen soll. Wie diese Hervorhebung aussieht, wird in der Visualisierungskomponente entschieden.

5.5.3 Verweise

Nicht nur Texte oder Markierungen sind als Annotationsinhalt vorstellbar. Auch größere Dokumente unterschiedlichster Formate (PDF, DOC, HTML) sollen an Simulationsankern „aufgehängt“ werden können.

Schon bei kurzen Überlegungen kristallisiert sich eine ungeheure Menge unterschiedlicher Dateiformate heraus. Da es nicht möglich und auch technisch nicht sinnvoll ist eine Vielzahl von Formaten direkt mit dem Annotationssystem anzeigen zu können, ist eine Verweisfunktion die nahe liegende Lösung. Das Annotationssystem ruft dazu fremde Applikationen auf, um die Dokumente darzustellen. Um den Komfort zu erhöhen wird dabei das Mapping aus Abschnitt 5.2.3 auch auf diese Verweise (auf externe Dateien) eingesetzt um (wie auch dort schon erwähnt) Dokumente optional zusammen mit den Protokolldateien weiterzugeben und so das Konzept einer „Datentankstelle“ zu unterstützen.

Verweise haben aber natürlich schon eo ipso eine Daseinsberechtigung vor allem als Möglichkeit auf weitergehende umfangreiche Materialien eben nur zu verweisen (vgl. Literaturverweise).

Durch die Nutzung verschiedener AnnotationViewer Komponenten (s. 5.1 Architektur) ist ein zweistufiges System möglich: Einige Dokumenttypen wie zum Beispiel Audiodateien können direkt durch einen dafür entwickelten AnnotationViewer dargestellt bzw. abgespielt werden. Ist für einen Dokumenttypen keine Komponente zuständig, benutzt ein Fallback-Mechanismus (ebenfalls ein AnnotationViewer) die Mechanismen des Betriebssystems um eine passende Fremdapplikation aufzurufen. Die Verweise selbst sollen URLs sein. Von der Funktionsweise her entspricht das also dem Verhalten eines gängigen Internetbrowser.

5.5.4 Benennungen und Stichwortvorschläge durch den Verfasser

Durch den Einsatz von RDF stellt das aus Sicht der Datenspeicherung kein Problem dar. Es können einfach eigene Prädikate für Benennungen und Stichwörter benutzt werden. Allerdings werden im Prototyp nur vom Verfasser vorgegebene Stichwörter unterstützt. Diese stellen einen eigenen Annotationstyp dar und werden durch eine AnnotationCreator Komponente angeboten.

Die Nutzung durch den Benutzer wird durch eine AnnotationExtender Komponente ermöglicht, welche die für einen Zustand oder eine Entität gültigen Stichwörter als Metainformationen an eine Annotation anzuhängen erlaubt.

Eine AnnotationViewer Komponente ermöglicht darüber hinaus eine dem ITS-Konzept (s. 1) entsprechende Ansicht. Dabei wird die Anzahl der Annotationen aufgeschlüsselt nach Stichworten angezeigt.

5.6 Kollaboration

Wie in den Anforderungen dargestellt sind zwei verschiedene Kommunikationstypen zu unterstützen: asynchron und synchron.

5.6.1 Asynchron

Für die asynchrone Kommunikation werden zwei verschiedene Konzepte realisiert. Durch das Speichern von Annotationen in einer lokalen Datei und deren Weitergabe zum Beispiel per E-Mail kann eine asynchrone Kommunikation erreicht werden. Die Sichtbarkeit wird bei diesem Ansatz implizit durch eine auf die gewünschten Personen beschränkte Weitergabe erreicht.

Das zweite Konzept besteht in der Bereitstellung eines öffentlichen Servers. Dazu wird das Annoteprotokoll (s. 4.7) verwendet. Durch die Nutzung des HTTP-Protokolls ergeben sich folgende Vorteile:

- Integration in bereits installierte Webserver möglich

- Firewalltauglich
- Einfache Implementierung

Für den Prototyp wird auf eine Steuerung der Sichtbarkeit und ein Rechtemanagement verzichtet. Diese Anforderungen sind für den synchronen Betrieb von größerer Bedeutung und werden nur dort unterstützt. Im asynchronen Betrieb steht die öffentliche Diskussion im Vordergrund.

5.6.2 Synchron

Es gibt eine Vielzahl von Kommunikationsverfahren die sich für die synchrone Kommunikation eignen. Dazu gehören beispielsweise Entfernte Prozeduraufrufe (RPC: Remote Procedure Calls), Tupelräume und Ereignisdienste [BURGER97]. Verschiedene Implementierungen wurden in [PAPESCH03] und [OBER01] analysiert und verglichen. Da diese Verfahren aber nicht im Zentrum dieser Arbeit stehen, wird für den Prototyp die einfachste Lösung RMI (Remote Method Invocation) verwendet. Alternative Kommunikationsverfahren, die beispielsweise auch die Skalierbarkeit, etwa mithilfe von Multicast, verbessern, können dank der modularen Architektur jederzeit leicht in das Annotationssystem integriert werden.

Rechtemanagement und Sichtbarkeit

Für das Rechtemanagement über das auch die Sichtbarkeit gesteuert wird, ist eine Kombination aus einer AnnotationExtender Komponente und der Unterstützung durch den Server zuständig. Als grundlegende Technologie wird das NUSS-Framework verwendet. Der Server verarbeitet pro Annotation drei RDF Prädikate:

- NussReadRight
- NussChangeRight
- NussDeleteRight

Als RDF Objekte sind dabei stets Literale zu verwenden, die den verwendeten NUSS-Rechten entsprechen. Die Rechte selbst werden durch ein zum NUSS-Framework gehörendes Administrationstool und durch die ProDuctivE Anwendung vorgegeben. Die AnnotationExtender Komponente ermöglicht es für eine Annotation die NUSS-Rechte auszuwählen.

Ein Beispiel soll den Ansatz verdeutlichen: ProDuctivE benutzt für den Act-On-Behalf Modus (s. 2.8) unter Anderem das NUSS-Recht „BeAlice“. Nur Anwender, die dieses Recht besitzen können die Steuerung des Prozesses „Alice“ übernehmen. Falls nun eine Annotation über das Prädikat „NussReadRight“ mit dem Literal „BeAlice“ verbunden ist, können nur noch Benutzer die Annotation lesen, die auch den Prozess „Alice“ steuern dürfen.

Es ist zu beachten, dass aufgrund der bisherigen Einschränkungen des NUSS-Frameworks, kein rein personenbezogenes Rechtemanagement möglich ist vergleichbar etwa mit den ACLs (Access Control List) von Dateisystemen. Dies ist für den Einsatz in den entworfenen Szenarien allerdings auch nicht unbedingt nötig.

Neben der annotationsabhängigen Rechteverwaltung definiert der Server noch einige globale Rechte, die den grundsätzlichen (Lese-/Schreib-/usw.) Zugriff erlauben.

Registry

Neben der Rechteverwaltung des NUSS-Frameworks wird auch dessen Registry unterstützt. Alle in einer laufenden Sitzung gestarteten Server werden für den Anwender aufgelistet.

Der synchrone Server ist in jeden Client eingebaut, d.h. jeder Teilnehmer einer Sitzung kann, falls er das Recht dazu hat, einen Server betreiben. Dies ist vor allem für Lerngruppen wichtig. So kann jeder Studierende seine persönlichen Anmerkungen den Anderen anbieten.

5.6.3 Offlinebetrieb

Aufgrund der durchgängigen Nutzung desselben Datenmodells (RDF), ist es leicht das Herunterladen von Annotationen von einem Server zu ermöglichen, um so einen Betrieb ohne Internetverbindung zu unterstützen.

5.7 Metainformationen

Zur Nutzung von Metainformationen werden wie im Abschnitt Architektur beschrieben AnnotationExtender Komponenten verwendet. Für den Prototyp wird dabei nur ein einfaches System benutzt, das die Definition des Annotationstyps (Frage, Anmerkung, ...) und die freie Vergabe von Stichwörtern erlaubt.

5.8 Visualisierung

Die Visualisierung kann in drei Kategorien unterteilt werden:

Darstellung im Simulationsfenster

Die Darstellung direkt im Simulationsfenster wird vor allem für Symbole benutzt. Diese können etwa das Vorhandensein einer Annotation anzeigen. Auch der Typ der Annotation (Frage, Anmerkung, usw.) wird durch unterschiedliche Symbole (Fragezeichen, Ausrufezeichen, usw.) repräsentiert. Durch Anzeige eines Zählers bei den Entitäten, kann die ITS-Ansicht den Dozenten über die Anzahl der Annotationen, die sich auf eine Entität beziehen, informieren.

Hervorhebungen werden ebenfalls direkt im Simulationsfenster dargestellt. Entweder durch ein Symbol oder durch Farbänderungen.

Aber auch eine kurze Textannotation selbst kann direkt im Simulationsfenster angezeigt werden.

Zur Darstellung wird HiSAP benutzt bzw. erweitert.

Tooltips

Die Entitäten sind bereits von ProDuctive mit Tooltips versehen, so dass sich deren Erweiterung um die Annotation anbietet.

Externes Fenster

Ein externes Fenster wird vor allem für die Darstellung von Übersichten benutzt. Aber auch die zustandsabhängigen Annotationen können in einem solchen Fenster fortlaufend angezeigt werden, vergleichbar mit der Fehlerliste in einer integrierten Entwicklungsumgebung wie Eclipse.

5.8.1 Export

Auch die Möglichkeit eine Übersicht über die Annotationen zu exportieren wird durch eine AnnotationViewer Komponente ermöglicht.

5.8.2 Konfiguration

Die Darstellungskomponenten sind, etwa durch entsprechende Einträge im Hauptmenü, selbst dafür verantwortlich eine Konfigurationsmöglichkeit zu schaffen. Dazu gehört auch die Einstellung etwaiger Filterkriterien.

5.9 Filterung

Wie im Abschnitt Architektur beschrieben werden zur Filterung FilterProvider Komponenten verwendet. Hier bietet sich vor allem die Nutzung von Metainformationen und an.

5.10 Fazit

Der Entwurf adressiert die in Kapitel 2 definierten Anforderungen. Das besondere Augenmerk richtete sich auf die Schaffung einer erweiterbaren Architektur.

5.11 Quellen

BURGER97 Burger, Cora
 Groupware
 dpunkt-Verlag, Heidelberg, 1997, 2001

- DC** Dublin Core
Dublin Core Homepage
<http://www.purl.org>
- OBERO1** Oberparleiter, Peter
*Gemeinsame Nutzung von Smartboards durch Dozent/in und Studierende
über mobile Endgeräte*
Universität Stuttgart, IPVS, Studienarbeit, 2003
- PAPESCHO3** Papesch, Matthias
A Collaborative Environment for Learning Security Protocols
Universität Stuttgart, IPVS, Diplomarbeit, 2003
- RDF** W3C
Resource Description Framework (RDF)
<http://www.w3c.org/RDF/>
- RDFTUT** Champin , Pierre-Antoine
RDF Tutorial
<http://www710.univ-lyon1.fr/~champin/rdf-tutorial/index.html>
- SOMMERO1** Sommerville, Ian
Software Engineering, 6th Edition
Addison-Wesley, 2001
- URI** W3C
Naming and Addressing: URIs, URLs, ...
<http://www.w3c.org/Addressing/>

6 Bewertung

Dieses Kapitel spannt den Bogen vom Entwurf zurück zur Anforderungsanalyse und überprüft, ob das Annotationssystem den Erfordernisse gerecht wird.

6.1 Analyse hinsichtlich der Anforderungen

Dokumenttyp

Es werden Simulationen für Protokollspezifikationen in Form von Promela-Quelltexten unterstützt. Darüber hinaus werden auch Trail-Files und ProDuctivE-Aufzeichnungen unterstützt.

Speicherung

Durch den Einsatz von Promeladoc wird die dokumentinterne Speicherung im Quelltext unterstützt. Der Fokus liegt aber auf der externen Speicherung.

Dokumentbezug

Mithilfe einer Mappingdatei findet eine Abbildung lokaler Dateinamen auf globale Namen.

Verankerung von Zusatzinformationen

Es ist der Bezug auf Entitäten und Zustände möglich.

Weitergabe

Es wird die synchrone Weitergabe unterstützt. Die Asynchrone wird sogar auf zwei Arten unterstützt: Per öffentlichem Server oder als einfache Datei, etwa per E-Mail.

Sichtbarkeit

Die Steuerung der Sichtbarkeit wird für den Fall der synchronen Weitergabe durch ein Rechtemanagement unterstützt. Im asynchronen Fall ist durch Wahl des Speicherorts (öffentlicher Server \leftrightarrow lokale Datei) eine Unterscheidung zwischen privaten und öffentlichen Annotationen möglich.

Rechte

Ein ausgefeiltes Rechtemanagement unterstützt die synchrone Kollaboration mithilfe des NUSS-Frameworks. Das spezielle Augenmerk richtete sich dabei auf den „Act-On-Behalf“ Modus von ProDuctivE.

Arten von Zusatzinformationen

Es sind verschiedene Arten von Zusatzinformationen möglich. Dazu gehören beispielsweise Textnotizen und Hervorhebungen. Die Architektur erlaubt die einfache Unterstützung weiterer Annotationsarten.

Metainformationen

Es wird eine Reihe von Metainformationen wie Kategorien und Stichwörter unterstützt. Die Architektur erleichtert die Integration weiterer Metainformationen.

Darstellung

Die Darstellung kann auf der Simulationsfläche, einem externen Fenster und in Tooltips erfolgen.

Übersicht und Statistiken

Eine (evt. gefilterte) Übersicht der Annotationen ist möglich. Auch eine Anzeige der Anzahl der Annotationen zu einem bestimmten Stichwort dem Konzept des ITS folgend ist möglich.

Verankerungsrichtung

Es wird soweit von ProDuctivE unterstützt ein Rückkanal angeboten.

Navigationsanker

Die Verankerung wird als URL repräsentiert und kann einfach weitergegeben werden. Falls der Rückkanal verfügbar ist, ist auch eine Navigation möglich.

Filterung

Eine Filterung nach Metainformationen ist vorgesehen. Erweiterte Möglichkeiten können einfach in das System integriert werden.

6.2 Fazit

Die wesentlichen Anforderungen werden durch den Entwurf erfüllt. Eine einfache Integration weiterer Funktionalität ist für wichtige Richtungen (Annotationsart, Filterung, Kommunikation, usw.) vorgesehen.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Ziel dieser Arbeit war es ein bestehendes System zur Simulation von Sicherheitsprotokollen um die Möglichkeit der Verankerung von Zusatzinformationen zu erweitern. Die Erstellung von Einsatzszenarien und die Analyse des Zielsystems ProDuctivE ermöglichte die Formulierung von Anforderungen, die an das Annotationssystem zu stellen sind. Die Schwerpunkte lagen dabei auf der Verankerung von Zusatzinformation in *Protokollsimulationen* und der kollaborativen Nutzung eines Annotationssystems.

Eine Untersuchung bereits existierender Annotationssysteme ergab mit Ausnahme von Annotea keine direkt verwendbaren Komponenten. Sie förderte allerdings eine Reihe von Konzepten, die zur Entwicklung dieses Annotationssystems herangezogen werden konnten, zu Tage.

Für den Entwurf des Annotationssystems wurden Lösungsansätze für die verschiedenen Problembereiche analysiert und diskutiert. Der besondere Fokus lag dabei auf einer flexiblen und erweiterbaren Architektur.

Die Bewertung des Entwurfs verglich das Ergebnis mit den zuvor definierten Anforderungen.

Das Annotationssystem unterstützt die speziellen Anforderungen für den Einsatz in einer Umgebung zur Protokollsimulation. Dazu gehören vor allem die zustandsabhängige Verankerung und die geeignete Visualisierung von Annotationen. Neben der Darstellung von einzelnen Annotationen bestehen auch besonders in der Verwendung von Übersichten und Export weitere Möglichkeiten.

Auch die Erfordernisse unterschiedlicher Einsatzszenarien im Hinblick auf die Kollaboration werden durch das Annotationssystem unter anderem mit dem Angebot von synchroner und asynchroner Kommunikation unterstützt. Ein Rechtemanagement mit besonderer Beachtung des Act-On-Behalf Modus von ProDuctivE wird für den Einsatz der synchronen Kommunikation unterstützt.

Die Architektur ermöglicht ein einfaches Erweitern des bestehenden Annotationssystems.

7.2 Ausblick

Es gibt eine ganze Reihe an Betätigungsfeldern für die Zukunft. Eine nahe liegende Möglichkeit ist die Unterstützung weiterer Annotationsinhalte. Da im Rahmen des NUSS-Projekts auch an der Unterstützung von Audioaufzeichnungen von Vorlesungen und Übungen gearbeitet wird, kann als Erläuterung auch der Verweis auf die entsprechende Aufnahme verwendet werden. Dafür ist eine Komponente zum direkten Abspielen von Tonaufnahmen innerhalb des Annotationssystems sinnvoll.

Auch die im Bereich der Darstellung und Analyse von Annotationen gibt es Erweiterungsmöglichkeiten. Besonders weitere Ansätze zur Unterstützung des Dozenten, neben den bereitgestellten Übersichten und der Verwendung des Donut-Konzepts, sind interessant. Die Frage, welche Auswertungen für einen Dozenten besonders hilfreich sind, um die Bedürfnisse der Studierenden besser zu verstehen und die Lernmaterialien dementsprechend anzupassen, verdient Aufmerksamkeit.

Eine erweiterte Nutzung von Annotationen zur Navigation ist ebenfalls denkbar. Dabei kann die Zahl der visualisierten Zustände für einen speziellen Ablauf reduziert werden, um so beispielsweise für das Verständnis eines Protokolls irrelevante Initialisierungsarbeiten auszublenden. Die Funktionen schrittweise vor und zurück zu gehen wird dann durch ein Springen von Annotation zu Annotation ersetzt.

Ein weiteres interessantes Betätigungsfeld ist die Nutzung eines noch mächtigeren Berechtigungsmanagements, das eine komplette Zugriffsteuerung für einzelne Annotationen ermöglicht (s. 5.6.2). In diesem Zusammenhang ist auch eine erweiterte Benutzerverwaltung sinnvoll. Allerdings ist das NUSS-Framework für den vorgesehenen Einsatzbereich ausreichend.

Der interessanteste Ansatz könnte in der Nutzung bzw. Anpassung des Annotationssystems für weitere Hostapplikationen bestehen. Eine Vielzahl der behandelten Konzepte lässt sich direkt auf andere Anwendungen übertragen und auch im Entwurf wurde auf eine möglichst weitgehende Generizität geachtet. Die wesentlichen Modifikationen sind dann selbstverständlich im Bereich der Verankerung vorzunehmen. Ein interessanter Bereich ist die Aufzeichnung des Fensterstroms von Applikationen (z.B. mittels VNC). Diese Anwendung für das Annotationssystem stellt ebenso wie Protokollsimulationen spezielle Anforderungen im Bereich der Verankerung und der Darstellung, die analysiert werden müssen. Annotationen können für diese Nutzung, ebenso wie etwa bei Audio- oder Videoaufzeichnungen zur Indizierung verwendet werden. Dies hängt eng mit der Nutzung als Navigationsmittel zusammen.

Für die genannten Beispiele ist es nicht nötig die grundlegende Architektur des Annotationssystems zu verändern.

In eine ganz andere Richtung geht die Fragestellung, inwieweit der Prozess der Protokollentwicklung, im Gegensatz zur Lehre, von Protokollen durch ein Annotationssystem unterstützt werden kann.

7.3 Fazit

Die Verankerung von Zusatzinformationen in Protokollsimulationen zur Unterstützung der Lehre stellt außergewöhnliche Anforderungen an ein Annotationssystem, die in dieser Arbeit diskutiert und adressiert wurden.

Anhang A: Projektplan

Zustandsabhängige Verankerung von Zusatzinformation bei Protokollsimulationen

Autor: Robert Sauter
Datum : 11.2.2003

A.1 Einleitung

Der Projektplan dient der/dem Bearbeiter(in) und der Betreuerin als Grundlage für die Projektüberwachung und -steuerung. Er legt den Projektablauf aber nicht ein für allemal fest, sondern soll im Laufe der Arbeit ergänzt und angepasst werden. So ist es z.B. möglich, dass anfänglich spezifizierte Arbeitspakete weiter verfeinert oder Arbeitspakete zugunsten anderer aufgegeben werden. Der Projektplan kann in Absprache mit der Betreuerin in beiderseitigem Einverständnis in den folgenden Fällen angepasst werden:

- bei Verzug,
- bei vorzeitigem Abschluss eines Arbeitspakets oder
- bei Gewinn neuer Erkenntnisse.

Der aktualisierte Projektplan soll in den Anhang der Ausarbeitung einfließen.

Projektbeschreibung:

Bei diesem Projekt handelt es sich um eine Studienarbeit an der Abteilung Verteilte Systeme des Instituts für Parallele und Verteilte Höchstleistungsrechner der Universität Stuttgart.

A.2 Beschreibung der Arbeitspakete

Dieser Abschnitt umfasst die Definition der wesentlichen Arbeitspakete, die Abhängigkeiten zwischen einzelnen Paketen und die Identifikation von kritischen Punkten, die die Durchführung des Projektes gefährden könnten.

Definition der Arbeitspakete

Die im Rahmen der Studienarbeit durchzuführenden Tätigkeiten lassen sich in folgende, zusammenhängende Arbeitspakete aufgliedern:

AP1, Projektplan: Erstellen eines initialen Projektplanes
Status: abgeschlossen

AP2, Einarbeitung: Genaue Einarbeitung in die Aufgabenstellung, die relevante Literatur und bestehenden Arbeiten auf der diese Studienarbeit aufbaut
Status: begonnen

AP3, Technische Einarbeitung: (Weiteres) Einarbeiten in die notwendigen Werkzeuge und Technologien zur Erfüllung der Aufgaben
Status: begonnen

AP4, Related Work: Suche und Bewertung von Arbeiten mit ähnlichen Zielen
Status: begonnen

AP5, Szenarien: Erstellung und Analyse von Szenarien

Status: -

AP6, Erläuterungssystem: Entwurf und (evt. prototypische) Implementierung des Erläuterungssystems für die ProductivE Umgebung

Status: -

AP7, Notizsystems: Entwurf und (evt. prototypische) Implementierung des Notizsystems für die ProductivE Umgebung

Status: -

AP8, Donut Integration: Entwurf und (evt. prototypische) Implementierung der Verbindung der ProductivE Umgebung an das Donut-System mit evt. Erweiterung von Donut

Status: -

AP9, Erweiterte Implementierung: Erweiterung der in AP6-8 erstellten Realisierungen

Status: -

AP10, Ausarbeitung: Bewertung der Arbeit und schriftlicher Ausarbeitung aller wesentlichen Ergebnisse

Status: -

AP11, Vortrag: Präsentation der Ergebnisse im Rahmen des VS-Kolloquiums

Status: -

Abhängigkeiten der Arbeitspakete

Zwischen den Arbeitspaketen der Studien bestehen die folgenden Abhängigkeiten:

Die Arbeitspakete können in folgende parallel ausführbare Gruppen aufgeteilt werden. Diese Gruppen müssen sequentiell bearbeitet werden.

AP1

AP2-5

AP6-7

AP8

AP9

Die Ausarbeitung (AP10) kann parallel dazu geschrieben werden. Der Vortrag (AP11) findet im Anschluss der Arbeit statt.

Kritische Punkte / Risiken

Ein kritischer Punkt ist nach der Begutachten des Related Work und Analyse der vorhandenen Basisarbeiten erreicht. Es ist dann zu entscheiden, welche Komponenten und Konzepte für das Annotationssystem übernommen werden.

A.3 Zeitplan

In diesem Abschnitt erfolgt die zeitliche Planung der Arbeitspakete und Meilensteine. Der Bearbeitungszeitraum ist 6 Monate oder 26 Wochen. Wenn sich der Zeitraum mehrerer Arbeitspakete überlappt, dann werden diese parallel bearbeitet. Am Ende des Bearbeitungszeitraums ist ein Puffer eingeplant worden, der für Verzögerungen oder andere unvorhergesehene Tätigkeiten genutzt werden soll.

Geplanter Zeitraum	Aufwand	Arbeitspakete und Meilensteine	Tatsächlicher Zeitraum
	3 PW	AP1 Projektplan und AP2 Einarbeitung Meilenstein 1	
	4 PW	AP3 Technische Einarbeitung, AP4 Related Work und AP5 Szenarien Meilenstein 2	
	8 PW-x	AP6 Erläuterungssystem und AP7 Notizsystem Meilenstein 3	
	2 PW	AP8 Donut Integration	
	2 PW+x	AP9 Erweiterte Implementierung Meilenstein 4	
	4 PW	AP10 Ausarbeitung Meilenstein 5	
	2 PW	Korrekturlesen und Verbessern Meilenstein 5	
	1 PW	Puffer	

A.4 Dokumente und Meilensteine

Als Ergebnis der jeweiligen Arbeitspakete sollen zu den Meilensteinen folgende Dokumente entstehen, die nach diesen einer Versionskontrolle unterliegen und nur in Absprache mit der Betreuerin geändert werden sollen.

Meilenstein 1: Projektplan, erste Einleitung

Meilenstein 2: Kapitel Szenarien und Related Work

Meilenstein 3: Lauffähige Realisierung

Meilenstein 4: Finale Realisierung

Meilenstein 5: Initiale schriftliche Ausarbeitung

Meilenstein 6: Finale schriftliche Ausarbeitung