

Projektgruppenantrag

Visuelle Verifikation und Validierung via Animation - Entwicklung eines Animator-Generators zur (online-)Animation von Simulationstraces

1. Thema

2. Zeitraum

SoSe 2014 und WiSe 2014/15

3. Veranstalter

Karl Napp, Informatik XXI, OH 16, Raum 301

Tel.: 4711, E-Mail: Karl.Napp@tu-dortmund.de

Willi Wacker, Informatik IX, OH 14, Raum 007

Tel.: 9922, E-Mail: Willi.Wacker@tu-dortmund.de

Prof. Dr. Wilhelm Wichtig, Informatik IX, OH 14, Raum 008

E-Mail: wichtig@cs.tu-dortmund.de

4. Aufgabe

- Motivation:** In der Arbeitsgruppe Modellierung und Simulation werden am Lehrstuhl Informatik IV die Modellierungswerkzeuge HIT [MCDD⁺06] und ProC/B [BBF⁺02, Tep04] zur Leistungsanalyse von Rechen- bzw. logistischen Systemen eingesetzt. Dabei kommt u.a. als Lösungsverfahren ein ereignisdiskreter stochastischer Simulator zum Einsatz. Eine Simulation ist eine Vorgehensweise zur Analyse dynamischer Systeme [LK00]. Hierbei werden Experimente an einem Modell durchgeführt, um Erkenntnisse über ein reales System zu gewinnen. Im Zusammenhang mit Simulation spricht man von einem zu simulierenden System und von einem Simulator als Implementation oder Realisierung eines Simulationsmodells. Letzteres stellt eine Abstraktion des zu simulierenden Systems dar (Struktur, Funktion, Verhalten). Der Ablauf eines Simulators mit konkreten Werten (Parametrisation) wird als Simulationsexperiment bezeichnet. Dessen Ergebnisse können dann interpretiert und auf das zu simulierende System übertragen werden.

Unsere jüngsten Erfahrungen haben gezeigt, dass der Einsatz von Cut- & Paste-Operationen in grafikgestützten Modellierungswerkzeugen oft zu umfangreichen Modellen führt, die zwar syntaktisch korrekt sind aber dennoch semantisch mit der Zielsetzung wenig zu tun haben. Ein kleines Beispiel soll das veranschaulichen. Abbildung 1 zeigt die Objektstruktur eines Leistungsmodells, wobei Details hier keine Rolle spielen sollen. Auf oberster Ebene, im Modelltyp, sei eine probabilistische Aufteilung der Aufträge, gemäß Abbildung 2, gewünscht.

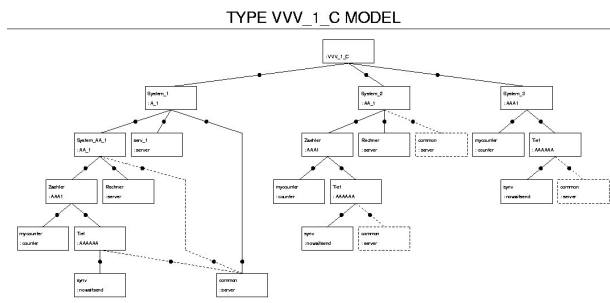


Abbildung 1: Konzeptuelle Struktur

BRANCH

PROB 0.7: Auftrag_1(negexp(0.7));

PROB 0.15: Auftrag_2(negexp(0.5));

ELSE : Auftrag_3(negexp(0.5));

END BRANCH;

Abbildung 2: Richtiger Code

Durch Cut- & Paste hat der Modellierer jedoch den in Abbildung 3 gezeigten Code erzeugt, was zur Folge hat, dass das Modell syntaktisch korrekt ist aber dynamisch nur der rechte Strang, dargestellt in Abbildung 4, durchlaufen wird.

```

BRANCH
  PROB 0.7: Auftrag_3(negexp(0.7));
  PROB 0.15: Auftrag_3(negexp(0.5));
  ELSE : Auftrag_3(negexp(0.5));
END BRANCH;

```

Abbildung 3: Cut- & Paste-Fehler

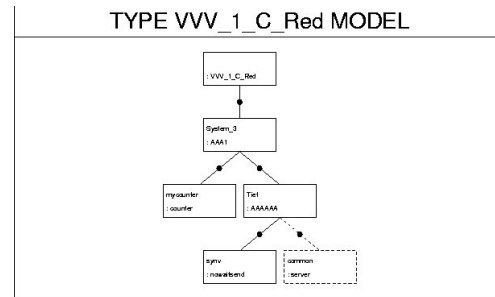


Abbildung 4: Ergebnisstruktur

Es muss geprüft werden, ob das realisierte Simulationsmodell dem zu simulierenden System entspricht: Unter einer Verifikation wird eine Überprüfung der Transformation eines konzeptionellen Simulationsmodelles in ein „korrekt“ arbeitendes (Computer-)Programm verstanden, während bei einer Validierung untersucht wird, ob das konzeptionelle Simulationsmodell das zu simulierende System genau genug nachbildet.

Gerade von ungeübten Modellierern wird das oft unterschätzt da sie den Schritt der Modellvalidierung, wie er nach [LK00] gefordert ist, gern aus dem Weg gehen. In [Sar04] werden verschiedene Validierungstechniken, die während eines Modellierungsprozesses, siehe Abbildung 5, zum Einsatz kommen, beschrieben. Modellvalidierung steht somit im Mittelpunkt von Modellierungsprozessen. Den Modellierern fehlen hierzu jedoch geeignete praktische Hilfsmittel. Hier soll die Arbeit der Projektgruppe ansetzen.

Um die Arbeitsweise - im Sinne einer operationalen Validierung - eines Simulators nachvollziehen zu können, bedient man sich der Animation sogenannter Traces, das sind zeitliche Aufzeichnungen der Änderungen ausgewählter Zustandsvariablen des Simulationsmodells. Da die Traces abhängig vom Modellierungstool sind, also proprietär festgelegte Formate haben, wäre es erforderlich für jedes Trace-Format einen eigenen Animator zu entwickeln. Eine Animation ist eine computerunterstützte Technik für dynamisch dargestellte und bewegte Grafiken, Texte oder Bewegtbild-darstellungen. Eine Animation fördert durch das kontinuierliche, sporadische oder sprunghafte Hervorheben und Verändern von Grafiken, Grafikdetails oder Textstellen die Erkennung, Verständlichkeit und die Lernfähigkeit. Mit Hilfe von Animationen können funktionale Abläufe ebenso wie Präsentationen dynamisch gestaltet werden.

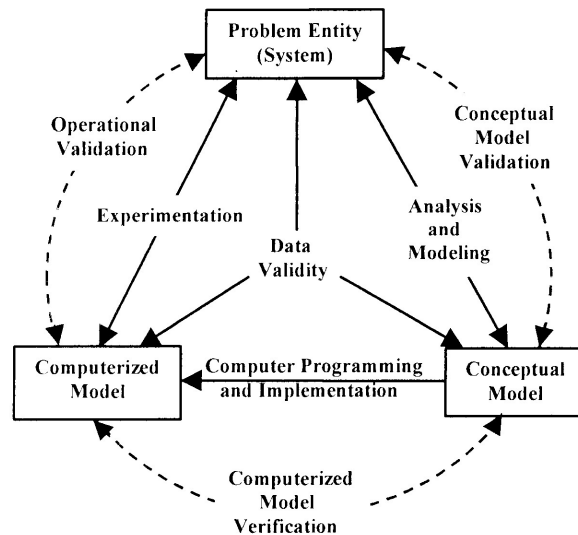


Abbildung 5: Einfacher Modellierungsprozess

Durch den Einsatz, aus dem Complerbau bekannter, generierender Techniken [Sch98] können aus toolspezifischen Trace-Beschreibungen angepasste Animatoren erzeugt werden, die einerseits

Durch den Einsatz, aus dem Complerbau bekannter, generierender Techniken [Sch98] können aus toolspezifischen Trace-Beschreibungen angepasste Animatoren erzeugt werden, die einerseits

dynamisch die Modellstruktur rekonstruieren und den Prozeßfluss innerhalb des Modelles visualisieren.

- **Aufgabenstellung:**

Den folgenden Arbeitspunkten liegt das Strukturdiagramm aus Abbildung 6 zugrunde. Ein zu generierender Animator (A) soll einen von einem Simulator erzeugten Trace (T) in eine graphische Darstellung transformieren. Dabei wird gefordert, dass aus den Trace-Daten die Struktur des simulierten Systems dynamisch rekonstruiert werden kann, d.h., dass der Animator wie ein Compiler die Trace-Daten als Eingabesprache, die in der Trace-Beschreibung (TD) definiert ist, betrachtet und übersetzt. Ausserdem ist der Prozeßfluss geeignet darzustellen.

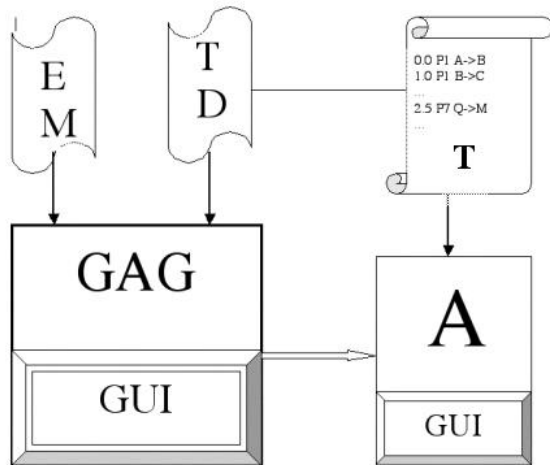


Abbildung 6: Strukturdiagramm

Der hierfür erforderliche grafikgestützte Animator-Generator (GAG) soll mittels einer graphischen Benutzungsoberfläche aus einer den Trace (T) beschreibenden Trace-Beschreibung (TD) und einer Beschreibung von Element-Abbildungen (EM) einen Animator (A) erzeugen, d.h., dass je nach Wahl der EMs eine andere Darstellung generiert werden kann.

Der hierfür erforderliche grafikgestützte Animator-Generator (GAG) soll mittels einer graphischen Benutzungsoberfläche aus einer den Trace (T) beschreibenden Trace-Beschreibung (TD) und einer Beschreibung von Element-Abbildungen (EM) einen Animator (A) erzeugen, d.h., dass je nach Wahl der EMs eine andere Darstellung generiert werden kann.

Im einzelnen soll die Projektgruppe

- sich in einem Modellierungspraktikum mit Methoden der Leistungsbewertung vertraut machen;
- mittels compilergenerierender Techniken einen grafikgestützten Animator Generator (GAG) zur Erzeugung eines Animators (A) konzipieren und wahlweise in C++ oder Java realisieren;
- einfache Anwendungsbeispiele mit dem Modellierungswerkzeug HIT entwerfen und damit den GAG und den Animator testen;
- eine Bibliothek von Element-Abbildungen anlegen, die es erlaubt, problembezogene Animationsbausteine einzubinden;
- an einem vorgegebenen umfangreichen HIT-Simulationsmodell die Funktion des GAGs und des Animators demonstrieren;
- einen Zwischenbericht zur Dokumentation der Arbeiten im ersten PG-Semester sowie einen Endbericht zur Dokumentation der PG-Ergebnisse erstellen;

5. Teilnahmevoraussetzungen

- Vorlesung „Mensch-Maschine-Interaktion“ oder dazu äquivalente Kenntnisse (V)
- Kenntnisse über Algorithm Engineering, z.B. durch Teilname an der gleichnamigen Vorlesung (W)
- Kenntnisse über Übersetzerbau, z.B. durch Teilname an der Vorlesung (W)
- UNIX-/LINUX-Kenntnisse (W)
- „Modellgestützte Analyse und Optimierung“ oder dazu äquivalente Kenntnisse (W)
- „Modellierung und Simulation diskreter und kontinuierlicher Systeme“ oder dazu äquivalente Kenntnisse (W)
- Kenntnisse in objektorientierter Programmierung, z.B. in Java (V)

Legende: (M) Mindestens eine; (V) Voraussetzung; (W) wünschenswert

6. Minimalziel(e)

- Konzeption eines Animator-Generators für das Modellierungstool HIT
- Konzeption einer online Trace-Animation für das Modellierungstool HIT
- prototypische Realisierung der Konzepte
- Exemplarischer Einsatz anhand von Anwendungsbeispielen und eines vorgegebenen HIT- Simulationsmodells
- Erstellung eines Endberichts
- Fachgespräch

7. Literatur

- [BBF⁺02] F. Bause, H. Beilner, M. Fischer, P. Kemper, and M. Völker. The proc/btoolset for the modelling and analysis of process chains. In *TOOLS 2002*, pages 51–70, 2002.
- [BDF⁺06] M. Büttner, M. Deniz, B. Fricke, O. Klaaßen, J. Mäter, S. Nolte, A. Schulze, M. Sczittnick, H. Stahl, N. Weißenberg, and C. Wysocki. Hi-slang reference manual, version 3.8.01. Interner Bericht; Universität Dortmund, Informatik IV, 2006.
- [Bur02] M. Burghardt. *Projektmanagement, Leitfaden für die Planung, Überwachung und Steuerung von Entwicklungsprojekten*. Publicis MCD Verlag, 2002. ISBN 3-89578-120-7.
- [KT06] P. Kemper and C. Tepper. Trace analysis - gain insight through modelchecking and cycle reduction. SFB Report 06007, Universität Dortmund; Informatik IV, 2006. SSN 1612-1376.
- [LK00] Averill M. Law and David W. Kelton. *Simulation Modelling and Analysis*. McGraw-Hill Education - Europe, April 2000.
- [MCDD⁺06] B. Müller-Clostermann, M. Deniz, W. Dittrich, J. Mäter, S. Shaban, and N. Weißenberg. Hit and hi-slang - an introduction, version 3.8.01. Interner Bericht; Universität Dortmund, Informatik IV, 2006.
- [Sar04] Robert G. Sargent. Validation and verification of simulation models. In *Winter Simulation Conference*, pages 17–28, 2004.
- [Sch98] A.T. Schreiner. Compilerbau mit java, 1998. Vorlesungsskript.
- [Ste00] P.A. Steinbuch. *Projektorganisation und Projektmanagement*. Kiehl, 2000.
- [Tep04] C. Tepper. Prozessablauf-visualisierung von proc/b-modellen. SFB Report 04003, Universität Dortmund; Informatik IV, 2004. SSN 1612-1376.

8. Rechtlicher Hinweis

Die Ergebnisse der Projektarbeit und die dabei erstellten Software sollen der Fakultät für Informatik uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.

9. PG-Realisierung

Im folgenden ist ein Vorschlag für eine Zeitplanung der beiden Projektsemester (Tabellen 1 und 2) aufgeführt. Der Projektgruppe wird Gelegenheit gegeben, selbst einen Projektzeitplan zu erstellen und nach diesem, in Abstimmung mit dem Veranstalter, die Projektarbeit durchzuführen.

Aufgabe	Anfang	Ende	Wochen
<i>Einarbeitungsphase</i>			
- Vorbereitung u. Seminar	KW 14	KW 15	2
- Projektplan erstellen	KW 15	KW 15	1
- Modellierungspraktikum	KW 16	KW 20	5
<i>Analysephase</i>			
- Einsatz generierender Techniken	KW 18	KW 22	5
- Trace Beschreibungen	KW 18	KW 22	5
- Animationstechniken	KW 18	KW 22	5
- Online-Analyse-Techniken	KW 18	KW 22	5
<i>Konzeptions- & Entwurfsphase</i>			
- Animator-Generator	KW 23	KW 28	6
- GUI	KW 23	KW 28	6
- Online-Analyse	KW 23	KW 28	6
- Anwendungsbeispiele	KW 23	KW 28	6
<i>Validierungsphase</i>			
- Anwendungsbeispiel (von Hand)	KW 26	KW 28	3
<i>Dokumentation</i>			
- Zwischenbericht	KW 16	KW 28	13

Tabelle 1: Projektplan SoSe 2007

Legende: KW = Kalenderwoche

Die PG wird mit einem Blockseminar beginnen. Hierzu erhalten die Teilnehmer in der letzten Woche der Vorlesungszeit des WiSe 2006 Themen für die Seminarvorträge und Hinweise auf relevante Literaturstellen. Das Seminar soll einerseits zur Einarbeitung in die Themengebiete der quantitativen Analysetechniken, insbesondere in ereignisdikreter Simulation dienen und andererseits Methoden des Projektmanagements [Bur02], [Ste00] behandeln.

Die Einarbeitungsphase ist so angelegt, dass sich die Teilnehmer Modellierungskennntnisse im Rahmen eines Seminars erarbeiten und in dem anschließenden Modellierungspraktikum vertiefen. In diesem Praktikum werden die PG-Teilnehmer in vier Teilgruppen vier unterschiedliche Mini-Modellierungsprojekte bearbeiten, wobei sie die Modellierungswerkzeuge HIT [BDF⁺06] und ProC/B [BBF⁺02] kennenlernen. Die Mini-Modellierungsprojekte werden so ausgelegt sein, dass insbesondere Probleme der Verifikation von Simulationsprogrammen aufgezeigt werden.

Mit Beginn des Praktikums, soll ein Projektmanagementmodell ausgewählt und ein Projektzeitplan erstellt werden.

In der fünften Woche, also noch während des Modellierungspraktikums, beginnt die Analysephase. In vier Untergruppen werden

- Einsatz generierender Techniken,
- Trace Beschreibungen sowie
- Animationstechniken und

- Online-Analyse-Techniken

zu untersuchen sein.

In der nach neun Wochen beginnenden Konzeptions- und Entwurfsphase soll

- ein Animator-Generator,
- eine Grafische Benutzungsschnittstelle,
- Möglichkeiten von Online-Analysen,
- eine Reihe von Anwendungsbeispielen, die teilweise im zweiten PG-Semester exemplarisch getestet werden sollen,

konzipiert werden.

In der Validierungsphase soll das entwickelte Konzept anhand eines der entworfenen Anwendungsbeispiele von Hand geprüft werden.

Parallel zu allen o.a. Arbeiten werden die Ergebnisse aller Teilaktivitäten in einem Zwischenbericht festgehalten, der das Ergebnis des ersten Projektsemesters repräsentieren wird.

Während der vorlesungsfreien Zeit im SoSe 2007 (KW 29 - KW 41) sind keine Aktivitäten geplant.

Aufgabe	Anfang	Ende	Wochen
<i>Konsolidierungsphase</i>	KW 42	KW 43	2
<i>Implementierungsphase</i>	KW 44	KW 02	10
<i>Testphase</i>	KW 46	KW 01	7
<i>Exemplarischer Testeinsatz</i>	KW 03	KW 05	3
<i>Dokumentation</i> - Endbericht	KW 42	KW 06	16

Tabelle 2: Projektplan WiSe 2007/08

Legende: KW = Kalenderwoche

Das zweite PG-Semester wird nur grob skizziert und wird im wesentlichen damit beginnen das im ersten PG-Semester entwickelte Konzept zu konsolidieren und Implementierungs- und Testarbeiten sowie einem exemplarischen Testeinsatz, anhand der im ersten PG-Semester entwickelten Anwendungsbeispiele, zum Inhalt haben. Der eigentliche Projektzeitplan soll von der Gruppe erstellt werden.

Die Projektgruppe wird abgeschlossen mit einem Endbericht, in dem die zweite Projektphase dokumentiert wird sowie mit einem Fachgespräch, in dem die Projektergebnisse dem Fachbereich öffentlich bekannt gemacht werden.

Während der gesamten Projektarbeit wird großer Wert auf selbständige Projektorganisation und projektbegleitende Dokumentation gelegt.

10. Erweiterungsmöglichkeiten

Die o.a. Aufgabenstellung dieser Projektgruppe bildet nur einen kleinen Ausschnitt aus dem Bereich der Verifikation und Validierung von Simulationsmodellen. Eine denkbare Erweiterungsmöglichkeit wäre eine Transformation der Trace-Daten in Message Sequence Charts und einem eingeschränktem Modelchecking wie in [KT06] für das ProC/B-Tool beschrieben.

11. Beantragung von Ressourcen

Die im folgenden beantragten Ressourcen gehen nicht über den PG-Ressourcen-Standard hinaus, bzw. werden, falls nicht über die IRB verfügbar, vom Lehrstuhl Informatik IV gestellt.

- **Rechensysteme:**
Die unten aufgeführte Hard- und Software bildet den augenblicklichen Planungsstatus. Eventuell später notwendig werdende Ressourcen werden nach Absprache mit der IRB und dem Projektgruppenveranstalter installiert.
- **Hardware:**
4 UNIX/LINUX-Graphikarbeitsplätze
- **Software:**
HIT, ProC/B-Tool, Traviando (werden vom LS IV bereitgestellt)
- **Räume und Raumbelegungen:**
Einen PG-Arbeitsraum und einen Seminarraum nach Absprache mit dem Raumadministrator
- **Haushaltsmittel:**
Übliche Mittel für Druck- und Vervielfältigungskosten sowie Exkursionsmittel

12. Lehrdeputat

- Karl Napp: 4 SWS
- Willi Wacker: 4 SWS
- Prof. Dr. Wilhelm Wichtig: 0 SWS