Operations Research und Simulation in SmartFrame

zur Erlangung des Grades

Doktor der Wirtschaftswissenschaften
(Dr. rer. pol.)

am

Department Wirtschaftswissenschaften der Universität Hamburg

eingereichte

kumulative Dissertation

von Torsten Reiners

(geboren in Cuxhaven)

Eingereicht am: 1. Juni 2005

Referent: Prof. Dr. Stefan Voß

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe nur unter Verwendung der angeführten Literatur angefertigt habe.

Torsten Reiners

Erklärung zum Promotionsvorhaben

Hiermit erkläre ich, dass ich zuvor noch keiner Doktorprüfung unterzogen wurde sowie ich mich noch um keine Zulassung an der Universität Hamburg bzw. einer anderen Universität beworben habe. Weiterhin habe ich noch keiner Universität oder ähnlichen Einrichtung eine Dissertation vorgelegt.

Torsten Reiners

meiner Liebe und Motivation

Operations Research und Simulation in SmartFrame

Die kumulative Dissertation beschreibt in 13 Fachartikeln die Entwicklung der innovativen, modularen virtuellen Lernumgebung SmartFrame vom (didaktischen) Konzept bis zur eigentlichen Umsetzung sowie Anwendung in unterschiedlichen Szenarien. Darüber hinaus wurde anhand der Fachgebiete Operations Research – hier insbesondere das Teilgebiet Meta-Heuristiken – und Simulation durch Schaffung spezieller Lerninhalte im Rahmen von Veranstaltungen die Anwendbarkeit in Hauptstudiumsveranstaltungen aufgezeigt und durch mehrere Evaluationen belegt.

Grundlage für die Fachartikel war die wissenschaftliche Mitarbeit am BMBF-Projekt *Virtuelles Studienfach Operations Research/Management Science* (VORMS; Projektnummer: 01NM094D.). Ursprünglich als reines Projekt zur Entwicklung von Lerninhalten ausgeschrieben, wurde ausgehend vom technologischen Stand Anfang 2001 im Bereich e-Learning und den anvisierten Zielen in Bezug auf die Umsetzung innovativer Konzepte und Technologien das Projektszenario um die Konzeption und exemplarische Umsetzung einer virtuellen Lernumgebung erweitert und entsprechend im Prototyp SmartFrame realisiert. Ein zweiter – in die kumulative Dissertation einfließender – Forschungsbereich ist die bereits in der Diplomarbeit begonnene Entwicklung und Anwendung von Algorithmen im Bereich *Maximum Likelihood Clustering*; hier mit dem Schwerpunkt auf die unterstützende Auswertung von (allgemeinen) Experimenten. Der Bezug zum e-Learning und die damit einhergehende Verwendung innerhalb meiner kumulativen Promotion ist durch die Ausrichtung der Lerninhalte im Bereich *Meta-Heuristiken* gegeben und als Konzept Bestandteil von Fachartikeln

Mit der kumulativen Dissertation eingereichte Fachartikel

Veröffentlichungen in Zeitschriften:

- Experiments with, and on, Algorithms for Maximum Likelihood Clustering (mit D.L.Woodruff). Computational Statistics & Data Analysis 47(2), 237–253 (2004)
- Teaching Meta-Heuristics within Virtual Learning Environments (mit S.Voß). International Transactions in Operational Research 11, 225–238 (2004)
- Die virtuelle Lernumgebung SMARTFRAME: Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte (mit D.Reiß, I.Sassen, S.Voß). i-com 3, 27–35 (2003)

Beiträge in Konferenzbänden:

- Synchronized Blended Learning in Virtual Learning Environments (mit C.Frank, I.Sassen, L.Suhl, S.Voß). Eingereicht bei der E-Learn 2005 (2005)
- Supporting the Authoring Process of Hierarchical Structured Learning Material (mit D.Reiß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004. 917–924 (2004)
- Instructional Design and Implementation of Interactive Learning Tools (mit I.Sassen, B.Paschilk, S.Voß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004, 1918–1921 (2004)
- Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung (mit D.Reiß, I.Sassen, S.Voß). In: L.Suhl und S.Voß (eds.) E-Learning in Wirtschaftsinformatik und Operations Research. DSOR Lab/BoD, Paderborn, 77–115 (2003)
- Instructional design of interactive learning modules by example of a special flow shop problem (mit R.Eisenberg, I.Sassen, S.Voß). In: A.Palma dos Reis und P.Isaias (eds.) e-Society 2003 Proceedings, Volume 1, Iadis, Lisabon, 520–527 (2003)
- SMARTFRAME: An integrated environment for XML-coded learning material (mit D.Reiß, H.Schulze, S.Voß). In: W.Uhr, W.Esswein und E.Schoop (eds.) Wirtschaftsinformatik 2003/Band I, Physica, Heidelberg, 613–632 (2003)
- Meta-Heuristiken in Virtuellen Lernumgebungen (mit I.Sassen, S.Voß). In: U.Leopold-Wildburger, F.Rendl und G.Wäscher (eds.) Operations Research Proceedings 2002, Springer, Berlin, 359-364 (2003)
- Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts (mit D.Reiß, S.Voß). Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002), ICSC-Naiso Academic Press, Millet Alberta (2002), #100029-03-TR-026, 1–7 (2002)
- Mining the Data from Experiments on Algorithms using Maximum Likelihood Clustering (mit D.L.Woodruff, R.Singh). In: E.Rolland, N.S.Umanath (eds.) Proceedings of the 4th INFORMS Conference on Information Systems and Technology, INFORMS, Linthicum, 235–254 (1999)

Arbeitspapier:

- Some thoughts on how to educate OR/MS (mit S.Voß). Arbeitspapier, Universität Hamburg (2003)

Ko-Autorenschaft bei den eingereichten Fachartikeln

Professoren

- Leena Suhl, Universität Paderborn
- Stefan Voß, Universität Hamburg
- David L. Woodruff, University of California, Davis, CA, USA

Wissenschaftliche Mitarbeiter

- René Eisenberg, Universität Hamburg
- Christine Frank, Universität Paderborn
- Imke Sassen, Universität Hamburg

Studierende

- Björn Paschilk, Universität Hamburg
- Ramanpreet Singh, University of California, Davis, CA, USA
- Dirk Reiß, Technische Universität Braunschweig
- Henrike Schulze, Technische Universität Braunschweig

Inhaltsverzeichnis

Teil I

Begründung des thematischen Zusammenhangs

Kapitel 1

Operations Research und Simulation in SmartFrame

1.1 Einleitung

Die Zeit meiner Promotion war durch das Mitwirken an mehreren Projekten geprägt, welche die thematische Ausrichtung meiner universitären Forschung maßgeblich beeinflusst haben. Insbesondere das in Kooperation mit insgesamt sechs Universitäten[1] durchgeführte BMBF-Projekt *Virtuelles Studienfach Operations Research/Management Science* (VORMS)[2] stellt den zentralen Fokus dieser kumulativen Promotion dar und ist demnach Hauptthema in meinen bisherigen Fachartikeln. Das VORMS-Projekt selbst war ursprünglich als reines Projekt zur Entwicklung von Lerninhalten ausgeschrieben. Ausgehend von dem technologischen Stand Anfang 2001 im Bereich e-Learning und den anvisierten Zielen wurde in Bezug auf die Umsetzung innovativer Konzepte und Technologien das Projektszenario um die Konzeption und exemplarische Umsetzung einer virtuellen Lernumgebung erweitert und im Prototyp SmartFrame realisiert. Ein zweiter – in die kumulative Promotion einfließender – Forschungsbereich ist die bereits in der Diplomarbeit begonnene Entwicklung und Anwendung von Algorithmen im Bereich *Maximum Likelihood Clustering*; hier mit dem Schwerpunkt auf die unterstützende Auswertung von (allgemeinen) Experimenten. Der Bezug zum e-Learning und die damit einhergehende Verwendung innerhalb meiner kumulativen Promotion ist durch die Ausrichtung der Lerninhalte im Bereich *Meta-Heuristiken* gegeben und als Konzept Bestandteil von Fachartikeln; u.a. [2,10,13].

Im Folgenden ordne ich die im Rahmen der kumulativen Promotion eingebrachten Fachartikel in eine thematische – sowie hinsichtlich der Entwicklung von SmartFrame chronologische – Reihenfolge. Hierbei werde ich grundlegende Meilensteine der Forschungsarbeit darlegen, diese jedoch nicht im Detail ausführen, sondern vielmehr auf die entsprechenden Fachartikel verweisen; siehe Teil III für eine vollständige Reproduktion. Das Kapitel ist entsprechend der drei Phasen Konzept (Abschnitt 1.2), Umsetzung (Abschnitt 1.3) sowie Anwendung (Abschnitt 1.4) unterteilt und wird mit einer Konklusion abgeschlossen.

1.2 Konzept einer virtuellen Lernumgebung

Ausgehend von bestehenden Technologien zu Beginn des Projekts wurde für die Tagung *Network Learning 2002* ein Konzept entworfen, welches erste grundlegende Modelle zur Gestaltung der Lernmaterialien sowie der Komponenten zur Umsetzung der virtuellen Lernumgebung enthielt; siehe [11]. Hierzu zählten insbesondere die modulare, hierarchische Strukturierung der Lernobjekte, wobei zu diesem Zeitpunkt die Wiederverwendbarkeit im Vordergrund stand. Weitere Inhalte bezogen sich auf die Visualisierung mit mehreren Freiheitsgraden bezüglich der Gestal-

[1] Es handelt sich hierbei um die Universitäten Paderborn, Bochum, Berlin, Magdeburg, Hohenheim sowie Braunschweig (bzw. Hamburg bedingt durch den Ruf von Prof. Dr. Stefan Voß).

[2] Projektnummer: 01NM094D.

tung und der Zusammensetzung sowie (übersichtlicheren) Darstellung von aggregierten Informationen unter Nutzung der seitlich integrierten SmartBars innerhalb der Lernmaterialien. Für die Zusammensetzung sowie die semantischen Beziehungen der Lernobjekte wird in diesem Fachartikel ein hyperbolischer Baum vorgeschlagen; auf Grund der Komplexität der Materialien und deren Abhängigkeiten untereinander wurde dieser im weiteren Verlauf der Entwicklung durch einen Hypergraph[3] substituiert sowie um Komponenten zur Darstellung und Editierung von Lernmaterialien erweitert; siehe [5].

Das Konzept für SmartFrame wird in dem Arbeitspapier zum Workshop *Virtual Environments for Advanced Modeling* verfeinert und weitergeführt; siehe [13], wobei darüber hinaus Vorgehensmodelle bezüglich der Entwicklung von virtuellen Lernumgebungen diskutiert werden und eine zentrale Hypothese für die Notwendigkeit von e-Learning aufgestellt wird, die durch langfristige Experimente und Evaluationen zu belegen bzw. zu widerlegen ist.[4] Die Semantik der Lernobjekte und damit die Anwendung in adaptiven Lernszenarien wird in diesem Fachartikel als weiteres Argument für die aufwändige Modularisierung der Lernmaterialien neben der Wiederverwendbarkeit in den Vordergrund gestellt.[5] Weiterhin definiert der Fachartikel Methoden und Technologien, die zum einen die Grundlage für die Umsetzung von SmartFrame darstellen und zum anderen in weiteren Fachartikeln vertieft wurden. Hierzu zählen u.a. der didaktische Ansatz *Synchronized Blended Learning* für die Verschmelzung von Präsenzlehre und virtueller Lehre [4], die Systemarchitektur sowie eine mögliche Ausgestaltung der im Rahmen des VORMS-Projekts umzusetzenden Kurse in *Simulation* und *Meta-Heuristiken*.[6] Anzumerken ist weiterhin, dass das Arbeitspapier eine wesentliche Grundlage für den Fachartikel auf der Sitzung der Arbeitsgruppe Wirtschaftsinformatik der GOR, Paderborn, 2002 [3] war, wobei die ursprünglichen Konzepte des Fachartikels im Rahmen ihrer Umsetzung im Prototypen SmartFrame erweitert wurden.

Die Fachartikel bezüglich *Maximum Likelihood Clustering* sind zum aktuellen Zeitpunkt als Konzept zu sehen und stellen die unterstützende Funktion hinsichtlich der Auswertung von Experimenten dar. Eine Integration in den Kontext e-Learning ist bezüglich der Inhalte *Meta-Heuristiken* vorgesehen, indem Lernende bei der Durchführung und Auswertung von einführenden und realitätsnahen Experimenten angeleitet sowie durch unterschiedliche Methoden bei der Auswertung unterstützt werden. Anzumerken ist hierbei, dass die Fachartikel zunächst inhaltlich keinen direkten Bezug zum Themengebiet e-Learning aufweisen, im Gesamtrahmen jedoch eine Komponente für die Ausbildung von Studierenden im Bereich Operations Research darstellen und in der weiteren Entwicklung von SmartFrame Einfluss finden werden; siehe [1,12].

1.3 Umsetzung von SmartFrame

Eine technisch-orientierte Beschreibung von SmartFrame ist im Konferenzbeitrag zur Wirtschaftsinformatik 2003 gegeben; siehe [9] sowie darauf basierend [3]. Die Artikel belegen eine Realisierbarkeit des aufgestellten Konzepts und zeigen einen ersten Ansatz hinsichtlich der Nutzung der frei verfügbaren Software Apache Cocoon und der modularen, unmittelbar adaptiven Lernmaterialien bezüglich der Konfiguration und Charakteristik des Lernenden.

Eine Problematik, die in Bezug auf technologische Aspekte nur unzureichend in der Literatur vertieft wird, ist die Erstellung und Komposition von strukturierten Lernmaterialien unter Verwendung von Lernobjekten und deren Beziehungen zueinander. Insbesondere durch die vielfache Verwendung von (einfachen) HTML-Inhalten innerhalb von existierenden Lernumgebungen beschränken sich die Autorenwerkzeuge auf die Erstellung von HTML-Seiten und die rudimentäre Zuordnung von Meta-Daten zu den Inhalten. In SmartFrame wird ein allgemeiner Ansatz verfolgt, bei dem eine intensive Unterstützung bezüglich der Eingabe der Meta-Daten erfolgt und die Beziehungen der Lernobjekte graphisch in Form eines Hypergraphen definiert werden

[3]Siehe Touchgraph; `http://www.touchgraph.com`.

[4]Dieses ist u.a. ein Untersuchungsgegenstand des Promotionsvorhabens von Frau Imke Sassen.

[5]Ein direkter Bezug zwischen dem Aufbau der Lernmaterialien und der Adaptivität der virtuellen Lernumgebung an die Charakteristik des Benutzers ist u.a. in [3,7] zu finden.

[6]Das Konzept für den Kurs *Meta-Heuristiken* wurde zunächst auf der Konferenz GOR 02 in Klagenfurth, Österreich, vorgestellt; siehe [10].

können. Die Lernobjekte können in einem eigens entworfenen und umgesetzten Autorenwerkzeug editiert werden; siehe [5].

1.4 Anwendung von SmartFrame

Die virtuelle Lernumgebung SmartFrame ist als Prototyp umgesetzt und in verschiedenen Anwendungsszenarien eingesetzt worden; u.a. im Rahmen einer universitätsübergreifenden[7] Veranstaltung *Simulation* sowie begleitend zum *Rechnerpraktikum* von Frau Dr. Birgit Schwartz-Reinken an der Universität Hamburg. Darüber hinaus wurde die Gestaltung von virtuellen Kursen unter Einsatz der technologischen Möglichkeiten von SmartFrame in mehreren Fachartikeln publiziert. Hierbei wurden Komponenten für Veranstaltungen im Fachgebiet *Meta-Heuristiken* [2,7], *Simulation* [6,7] sowie *Produktion* [8] thematisiert.

1.5 Konklusion

Im Rahmen der Promotion wurde ein alternativer Ansatz zu der allgemeinen technologischen Ausgestaltung von virtuellen Lernumgebungen entwickelt, umgesetzt und anhand verschiedener Veranstaltungen in der universitären Lehre erprobt. Mit Abschluss der Promotion kann der entstandene Prototyp SmartFrame in vielfältigen Anwendungsszenarien zukünftig eingesetzt werden. Innerhalb der nachfolgenden Lebenszyklen können somit die Konzepte von SmartFrame – entsprechend der Vorgehensweise während der Promotion im Sinne von formativen Evaluationsdurchläufen – angewandt, evaluiert und hinsichtlich der Erschließung weiterer Nutzerkreise ausgebaut werden.

1.6 Literatur

[1] Experiments with, and on, Algorithms for Maximum Likelihood Clustering (mit D.L.Woodruff). Computational Statistics & Data Analysis 47(2), 237–253 (2004)

[2] Teaching Meta-Heuristics within Virtual Learning Environments (mit S.Voß). International Transactions in Operational Research 11, 225–238 (2004)

[3] Die virtuelle Lernumgebung SmartFrame: Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte (mit D.Reiß, I.Sassen, S.Voß). i-com 3, 27–35 (2003)

[4] Synchronized Blended Learning in Virtual Learning Environments (mit C.Frank, I.Sassen, L.Suhl, S.Voß). Eingereicht bei der E-Learn 2005 (2005)

[5] Supporting the Authoring Process of Hierarchical Structured Learning Material (mit D.Reiß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004. 917–924 (2004)

[6] Instructional Design and Implementation of Interactive Learning Tools (mit I.Sassen, B.Paschilk, S.Voß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004, 1918–1921 (2004)

[7] Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung (mit D.Reiß, I.Sassen, S.Voß). In: L.Suhl und S.Voß (eds.) E-Learning in Wirtschaftsinformatik und Operations Research. DSOR Lab/BoD, Paderborn, 77–115 (2003)

[7] Hierbei wurden die Ergebnisse im Rahmen einer virtuellen Veranstaltung an der Universität Hamburg – hier parallel angeboten zu einer Präsenzveranstaltung von Herrn Dr. Kai Gutenschwager – und der Universität Paderborn genutzt.

[8] Instructional design of interactive learning modules by example of a special flow shop problem (mit R.Eisenberg, I.Sassen, S.Voß). In: A.Palma dos Reis und P.Isaias (eds.) e-Society 2003 Proceedings, Volume 1, Iadis, Lisabon, 520–527 (2003)

[9] SmartFrame: An integrated environment for XML-coded learning material (mit D.Reiß, H.Schulze, S.Voß). In: W.Uhr, W.Esswein und E.Schoop (eds.) Wirtschaftsinformatik 2003/Band I, Physica, Heidelberg, 613–632 (2003)

[10] Meta-Heuristiken in Virtuellen Lernumgebungen (mit I.Sassen, S.Voß). In: U.Leopold-Wildburger, F.Rendl und G.Wäscher (eds.) Operations Research Proceedings 2002, Springer, Berlin, 359-364 (2003)

[11] Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts (mit D.Reiß, S.Voß). Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002), ICSC-Naiso Academic Press, Millet Alberta (2002), #100029-03-TR-026, 1–7 (2002)

[12] Mining the Data from Experiments on Algorithms using Maximum Likelihood Clustering (mit D.L.Woodruff, R.Singh). In: E.Rolland, N.S.Umanath (eds.) Proceedings of the 4th INFORMS Conference on Information Systems and Technology, INFORMS, Linthicum, 235–254 (1999)

[13] Some thoughts on how to educate OR/MS (mit S.Voß). Arbeitspapier, Universität Hamburg (2003)

Kapitel 2

Kumulative Promotion

2.1 Drei thematisch zusammenhängende Fachartikel

Die in Kapitel 1 gegebene Darstellung und Einordnung positionieren die im Rahmen dieser kumulativen Promotion eingereichten Fachartikel in einem gemeinsamen Themenzusammenhang. Der zentrale Fokus des Promotionsvorhabens liegt auf den drei Phasen Konzept, Umsetzung und Anwendung in Bezug auf die virtuelle Lernumgebung SmartFrame, welche insbesondere durch die folgenden drei thematisch zusammenhängenden Arbeiten repräsentiert werden.[1]

1. Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts (mit D.Reiß, S.Voß). Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002), ICSC-Naiso Academic Press, Millet Alberta (2002), #100029-03-TR-026, 1–7 (2002)
2. Die virtuelle Lernumgebung SmartFrame: Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte (mit D.Reiß, I.Sassen, S.Voß). i-com 3, 27–35 (2003)
3. Teaching Meta-Heuristics within Virtual Learning Environments (mit S.Voß). International Transactions in Operational Research 11, 225–238 (2004)

2.2 Veröffentlichung von Fachartikeln

Die Veröffentlichung und damit Bereitstellung von Forschungsergebnissen ist eine elementare Notwendigkeit in der wissenschaftlichen Forschung und Lehre. Im Rahmen meiner wissenschaftlichen Arbeit wird eine heterogene Verteilung bezüglich Zeitschriften und Konferenzen angestrebt[2] und neben nationalen insbesondere auch internationale Konferenzen und Zeitschriften als Präsentationsplattform ausgewählt.[3] Die der Arbeit beigefügten Artikel sind wie folgt veröffentlicht:

Veröffentlichungen in Zeitschriften:

- Experiments with, and on, Algorithms for Maximum Likelihood Clustering (mit D.L.Woodruff). Computational Statistics & Data Analysis 47(2), 237–253 (2004)
- Teaching Meta-Heuristics within Virtual Learning Environments (mit S.Voß). International Transactions in Operational Research 11, 225–238 (2004)

[1] Siehe auch Abschnitt 2.2 für eine vollständige Übersicht aller eingereichten Fachartikel.

[2] Konferenzen erlauben im Gegensatz zu Zeitschriften eine zeitnahe Präsentation der Ergebnisse mit der Option, im Rahmen eines Vortrages sowohl eine Diskussion anzuregen als auch ein Feedback von Wissenschaftlern verschiedener Fachrichtungen unmittelbar zu erhalten.

[3] Eine Übersicht der gehaltenen Vorträge auf nationalen und internationalen Konferenzen ist im Lebenslauf angegeben.

- Die virtuelle Lernumgebung SmartFrame: Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte (mit D.Reiß, I.Sassen, S.Voß). i-com 3, 27–35 (2003)

Beiträge in Konferenzbänden:[4]

- Synchronized Blended Learning in Virtual Learning Environments (mit C.Frank, I.Sassen, L.Suhl, S.Voß). Eingereicht bei der E-Learn 2005 (2005)
- Supporting the Authoring Process of Hierarchical Structured Learning Material (mit D.Reiß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004. 917–924 (2004)
- Instructional Design and Implementation of Interactive Learning Tools (mit I.Sassen, B.Paschilk, S.Voß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004, 1918–1921 (2004)
- Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung (mit D.Reiß, I.Sassen, S.Voß). In: L.Suhl und S.Voß (eds.) E-Learning in Wirtschaftsinformatik und Operations Research. DSOR Lab/BoD, Paderborn, 77–115 (2003)
- Instructional design of interactive learning modules by example of a special flow shop problem (mit R.Eisenberg, I.Sassen, S.Voß). In: A.Palma dos Reis und P.Isaias (eds.) e-Society 2003 Proceedings, Volume 1, Iadis, Lisabon, 520–527 (2003)
- SmartFrame: An integrated environment for XML-coded learning material (mit D.Reiß, H.Schulze, S.Voß). In: W.Uhr, W.Esswein und E.Schoop (eds.) Wirtschaftsinformatik 2003/Band I, Physica, Heidelberg, 613–632 (2003)
- Meta-Heuristiken in Virtuellen Lernumgebungen (mit I.Sassen, S.Voß). In: U.Leopold-Wildburger, F.Rendl und G.Wäscher (eds.) Operations Research Proceedings 2002, Springer, Berlin, 359-364 (2003)
- Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts (mit D.Reiß, S.Voß). Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002), ICSC-Naiso Academic Press, Millet Alberta (2002), #100029-03-TR-026, 1–7 (2002)
- Mining the Data from Experiments on Algorithms using Maximum Likelihood Clustering (mit D.L.Woodruff, R.Singh). In: E.Rolland, N.S.Umanath (eds.) Proceedings of the 4th INFORMS Conference on Information Systems and Technology, INFORMS, Linthicum, 235–254 (1999)

Arbeitspapier:[5]

- Some thoughts on how to educate OR/MS (mit S.Voß). Arbeitspapier, Universität Hamburg (2003)

Ein Ranking der Zeitschriften variiert in Abhängigkeit der Quellen, Kriterien der Untersuchung sowie z.T. subjektiver Nuancen. Von den hier betroffenen Zeitschriften ist nur die *International Transactions in Operational Research* in der Ranking-Tabelle von A.-W. Harzing[6] – gemäß der Empfehlungen von Ranking-Tabellen innerhalb der Promotionsempfehlungen – angeführt, wobei die Bewertung von **A**[7], **B**[8], **6.6**[9] bis **1**[10] variiert. Die Zeitschrift *Computational Statistics*

[4]Der Vortrag auf der Konferenz Wirtschaftsinformatik 2003 wurde von Prof. Dr. Stefan Voß gehalten. Der letzte Beitrag für die E-Learn 2005 ist zur Veröffentlichung angenommen worden.

[5]Das Arbeitspapier wurde im Rahmen eines Workshops in Japan (Virtual Environments for Advanced Modeling, Ishikawa, Japan, 2002) erstellt.

[6]Siehe `http://www.harzing.com/download/jql.zip`.

[7]WI01; WU Wien Journal Rating 2001; `http://www.wuwien.ac.at/fides~/rating-definition_en.html`.

[8]VHB03; Association of Professors of Management in German speaking countries.

[9]BJM04; British Journal of Management 2001 Business & Management RAE rankings.

[10]Cra04; Cranfield University School of Management June 2004. Hierbei sei angemerkt, dass die Einordnung *lower national quality* für diese international verfügbare und mit internationalen Beiträgen versehene Zeitschrift von den übrigen Bewertungen stark abweicht und daher hinterfragt werden sollte.

& Data Analysis wird in anderen Quellen mit dem Ranking A[11] versehen bzw. im Vergleich mit 83 Zeitschriften aus dem Fachgebiet *Econometrics Journals* auf Position 18[12] eingeordnet. Ein Ranking für die deutschsprachige nationale Zeitschrift *i-com* des Oldenbourg-Verlags[13] ist mir nicht bekannt.

2.3 Ko-Autorenschaft

Alle beigefügten Fachartikel repräsentieren Ergebnisse von Forschungsprojekten und sind auf Grund dessen mit den Namen aller beteiligten Personen unabhängig des Status (extern, Student, wissenschaftlicher Mitarbeiter, Professor) veröffentlicht worden. Gemäß der angegebenen Berechnungsvorschrift – $2/n + 1$ mit n gleich der Anzahl an Autoren – ergibt sich für die *Promotionspunkte* ein Wert von 6,51. Dieser setzt sich wie folgt zusammen:[14]

Nr.	Titel	AZ	PP
1	Experiments with, and on, Algorithms for Maximum Likelihood Clustering	2	0,67
2	Teaching Meta-Heuristics within Virtual Learning Environments	2	0,67
3	Die virtuelle Lernumgebung SmartFrame: Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte	4	0,4
4	Synchronized Blended Learning in Virtual Learning Environments	5	0,33
5	Supporting the Authoring Process of Hierarchical Structured Learning Material	2	0,67
6	Instructional Design and Implementation of Interactive Learning Tools	4	0,4
7	Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung	4	0,4
8	Instructional design of interactive learning modules by example of a special flow shop problem	4	0,4
9	SmartFrame: An integrated environment for XML-coded learning material	4	0,4
10	Meta-Heuristiken in Virtuellen Lernumgebungen	3	0,5
11	Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts	3	0,5
12	Mining the Data from Experiments on Algorithms using Maximum Likelihood Clustering	3	0,5
13	Some thoughts on how to educate OR/MS	2	0,67
			$\sum 6,51$

Die hier angeführten Fachartikel sind nicht Teil einer weiteren Promotion und werden einzig für die eingereichte kumulative Promotion genutzt.

2.4 Substantieller Beitrag des Doktoranden

Die hier eingereichten Fachartikel stellen einen wesentlichen Bestandteil meiner wissenschaftlichen Forschung dar und beinhalten somit alle einen substantiellen Beitrag meiner Person. Die Fachartikel sind in Projekten und damit in Gruppen unterschiedlicher Zusammensetzung entstanden und beinhalten somit auch einen entsprechenden Anteil *externer* Komponenten. Im

[11] Siehe Fußnote 7.
[12] Ranking Courtesy of The Faculty of Economics and Business Administration, Vrije Universiteit Amsterdam; `http://bear.cba.ufl.edu/centers/MKS/marketing/science/econometric.pdf`.
[13] Siehe `http://www.oldenbourg.de/frame0.htm?http://www.oldenbourg.de/cgi-bin/romedia?Z=3630`.
[14] Für eine bessere Lesbarkeit werden nur der Titel des Artikels, die Autorenanzahl sowie die daraus resultierenden Promotionspunkte – auf zwei Nachkommastellen mathematisch gerundet – angeführt, wobei die Reihenfolge entsprechend der Auflistung in Abschnitt 2.2 beibehalten wird.

Folgenden sind drei Arbeiten genannt, bei denen mein Anteil sich deutlich[15] abzeichnet:[16]

- Experiments with, and on, Algorithms for Maximum Likelihood Clustering (mit D.L.Woodruff). Computational Statistics & Data Analysis 47(2), 237–253 (2004)
- Teaching Meta-Heuristics within Virtual Learning Environments (mit S.Voß). International Transactions in Operational Research 11, 225–238 (2004)
- SmartFrame: An integrated environment for XML-coded learning material (mit D.Reiß, H.Schulze, S.Voß). In: W.Uhr, W.Esswein und E.Schoop (eds.) Wirtschaftsinformatik 2003/Band I, Physica, Heidelberg, 613–632 (2003)

Keiner der für diesen Abschnitt relevanten Fachartikel ist zum aktuellen Zeitpunkt Bestandteil eines laufenden oder abgeschlossenen Promotionsvorhabens.

[15] Wobei dieses sich u.a auf Initiative, Entwicklung und Programmierung von notwendigen Applikationen sowie prozentualen Anteil an Worten beziehen und in keinster Weise die Leistung und Qualifikation der Ko-Autoren in Frage stellen soll.

[16] Der erste Fachartikel basiert auf meiner Forschungsarbeit an der *University of California, Davis* sowie der daraus entstandenen – mit dem Michehl-Preis ausgezeichneten – Diplomarbeit.

Teil II

Lebenslauf, Veröffentlichungen und Vorträge

Persönliche Daten

Name	Torsten Reiners
Titel	Dipl. Wirtsch.–Inform.
Wohnort	Quellenweg 7, 20535 Hamburg
Geburtstag	7. März 1972
Geburtsort	Cuxhaven
Nationalität	Deutsch
Familienstand	ledig
Eltern	Jürgen Reiners, 01.06.1942, Seelotse Gisela Reiners, geborene Schepler, 09.02.1947, Justizangestellte
Geschwister	Antje Reiners, 17.07.1976, Grundschullehrerin

Schulausbildung

1978 – 1984	Grundschule *Döse* in Cuxhaven
1984 – 1986	Orientierungsstufe *Döse* in Cuxhaven
1986 – 1991	Lichtenberg Gymnasium in Cuxhaven, Abitur mit der Note *gut* (2.3)

Militärdienst

1991 – 1992	3. Fernmeldebataillion in Buxtehude; Fernschreiber in der Fernmeldezentrale

Hochschulausbildung

10.1992 – 10.1998	Wirtschaftsinformatikstudium an der *Technischen Universität Braunschweig*. Diplom am 30.10.1998 mit der Note *sehr gut*
08.1994 – 01.1995	Auslandsstudium an der *University of Texas in Austin, TX, USA*
10.1995 – 03.2003	Zweitstudium in Informatik an der *Technischen Universität Braunschweig*. Vordiplom im September 1996
09.1997 – 07.1998	Forschungsaufenthalt an der *University of California, Davis, CA, USA* in der *Graduate School of Management*

Auszeichnung

10.1998	*Michehl–Förderpreis für Wirtschaftswissenschaften* für die Diplomarbeit mit dem Titel *Maximum Likelihood Clustering Data Sets Using a Multilevel, Parallel Heuristic*

Sprachkenntnisse

Englisch	9 Jahre Schulenglisch, weiterführende Kenntnisse durch Schüleraustausch, 6 Monate Rundreise in den USA, Studium an der *University of Texas, Austin, TX, USA* sowie Forschungsaufenthalt an der *University of California, Davis, CA, USA*
Französisch	5 Jahre am Gymnasium
Latein	2 Jahre am Gymnasium
Chinesisch	1 Semester an der *Technischen Universität Braunschweig*

Besondere Tätigkeiten

1989	8. Informatikwettbewerb, Platz 83
08.1990	Teilnahme an der *4. Schülerakademie* an der *Jugenddorf–Christopherusschule Braunschweig*. Veranstalter: Bildung und Begabung e.V., Bonn. Thema: Experimentalphysik
1992	10. Informatikwettbewerb, Platz 85
08.1994 – 12.1994	Ehrenmitglied der ACM (Association for Computing Machinery) in Austin, TX; Mitarbeit in der Kommission für die Administration der Kursseiten
SS 1995	Betreuung der Übung *Programmieren von Datenstrukturen in Modula 2*
WS 1995/96	Betreuung der Übung *Programmieren von Algorithmen in Scheme*
01.1996 – 10.1996	Installation, Organisation und Administration der Datenbank *Oracle* für das *Symposium on Operations Research 1996* in Braunschweig
10.1996 – 09.1997	Netzwerk und Software Administration in der Abteilung Allgemeine BWL, Wirtschaftsinformatik und Informationsmanagement
SS 1996	Betreuung der Übung *Programmieren in Pascal für Nicht-Informatiker*

Berufstätigkeit

11.1998 – 12.2003	Wissenschaftlicher Mitarbeiter an der *Technischen Universität Braunschweig* in der Abteilung Allgemeine BWL, Wirtschaftsinformatik und Informationsmanagement (Prof. Dr. Stefan Voß)
01.2004 – 04.2004	Wissenschaftlicher Mitarbeiter an der *Universität Hamburg* im Institut für Wirtschaftsinformatik (Prof. Dr. Stefan Voß)
05.2004 – 06.2004	Wissenschaftlicher Mitarbeiter an der *Otto-von-Guericke-Universität Magdeburg* am Lehrstuhl für Betriebswirtschaftslehre, insbesondere Management Science (Prof. Dr. Gerhard Wäscher)
07.2004 – 04.2005	Promotion
seit 05.2005	Wissenschaftlicher Mitarbeiter an der *Universität Hamburg* im Institut für Wirtschaftsinformatik (Prof. Dr. Stefan Voß)

Besondere Kentnisse

Sprachen	C/C++, Java, JSP, Lisp/Scheme, Prolog, Pascal, Modula, SQL, Perl, Skript-Sprachen
Weitere Kenntnisse	XML, XSLT, XSP, XHTML
Betriebssysteme	Unix/Linux, Windows NT (Nutzung/Administration)
Software	Apache Cocoon, Jakarta Tomcat
Themengebiete	e-Learning, Entwicklung und Nutzung von virtuellen Lernumgebungen, Simulation, Meta-Heuristiken

Projekte

05.1999 – 05.2000	Evaluation von bioanalogen Algorithmen in der Disposition auf einem Seehafen-Containerterminal (Hamburger Hafen- und Lagerhaus-AG)
06.2000 – 11.2000	Intelligente Flottenlogistik – Fahrzeugdisposition bei einer Autovermietung (AVIS-Deutschland)
07.2001 – 06.2004	Virtuelles Studienfach Operations Research/Management Science (Bundesministerium für Bildung und Forschung)

Lehre (Zeitraum 1999-2005)

1. Dozent der Veranstaltung *Wirtschaftsinformatik* (4+0 SWS) an der Universität Bremen
2. Durchführung der Übungen in der Wirtschaftsinformatik
3. Vertretung von Vorlesungssitzungen
4. Durchführung von Seminaren und Praktika in der Informatik und Wirtschaftsinformatik
5. Konzeption und Betreuung von Klausuren und Hausaufgaben
6. Betreuung von Diplom- und Studienarbeiten

Veröffentlichungen

1. Synchronized Blended Learning in Virtual Learning Environments (mit C.Frank, I.Sassen, L.Suhl, S.Voß). Angenommen zur Veröffentlichung bei der E-Learn 2005 (2005)
2. Ontology-based Retrieval, Authoring, and Networking for Generalized e-content (mit D.Reiß, I.Sassen). Angenommen zur Veröffentlichung bei der E-Learn 2005 (2005)
3. Modeling and Solving the Short-term Car Rental Logistics Problem (mit A.Fink). Transportation Research Part E: Logistics and Transportation Review (2005), im Druck
4. Experiments with, and on, Algorithms for Maximum Likelihood Clustering (mit D.L.Woodruff). Computational Statistics & Data Analysis 47(2), 237–253 (2004)
5. Supporting the Authoring Process of Hierarchical Structured Learning Material (mit D.Reiß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004. 917–924 (2004)
6. Instructional Design and Implementation of Interactive Learning Tools (mit I.Sassen, B.Paschilk, S.Voß). In: Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004, 1918–1921 (2004)
7. Teaching Meta-Heuristics within Virtual Learning Environments (mit S.Voß). International Transactions in Operational Research 11, 225–238 (2004)
8. Die virtuelle Lernumgebung SmartFrame: Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte (mit D.Reiß, I.Sassen, S.Voß). i-com 3, 27–35 (2003)
9. Implementing an Online Version of the Case Method: A Qualitative Evaluation (mit C.Frank, L.Suhl). Angenommen zur Veröffentlichung im Proceedings zur Konferenz ICDE World Conference on Open Learning and Distance Education, Hong Kong, China, Oktober (2003)
10. Lernerspezifische Komposition und Visualisierung von Lernmaterialien (mit D.Reiß, S.Voß). Arbeitspapier, Universität Hamburg (2003)
11. Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung (mit D.Reiß, I.Sassen, S.Voß). In: L.Suhl und S.Voß (eds.) E-Learning in Wirtschaftsinformatik und Operations Research. DSOR Lab/BoD, Paderborn, 77–115 (2003)
12. Kodierung und didaktische Aufbereitung von Lernmaterialien in XML (mit I.Sassen, S.Voß). Arbeitspapier, Universität Hamburg (2003)
13. Instructional design of interactive learning modules by example of a special flow shop problem (mit R.Eisenberg, I.Sassen, S.Voß). In: A.Palma dos Reis und P.Isaias (eds.) e-Society 2003 Proceedings, Volume 1, Iadis, Lisabon, 520–527 (2003)

14. SmartFrame: An integrated environment for XML-coded learning material (mit D.Reiß, H.Schulze, S.Voß). In: W.Uhr, W.Esswein und E.Schoop (eds.) Wirtschaftsinformatik 2003/Band I, Physica, Heidelberg, 613–632 (2003)

15. Some thoughts on how to educate OR/MS (mit S.Voß). Arbeitspapier, Universität Hamburg (2003)

16. Meta-Heuristiken in Virtuellen Lernumgebungen (mit I.Sassen, S.Voß). In: U.Leopold-Wildburger, F.Rendl und G.Wäscher (eds.) Operations Research Proceedings 2002, Springer, Berlin, 359-364 (2003)

17. Application Service Providing (mit A.Mies). wisu 1, 58–62 (2003)

18. XML-basierte Kodierung von Lernobjekten (mit D.Reiß, S.Voß). Arbeitspapier. Technische Universität Braunschweig (2002)

19. A Conceptual Framework for Synchronized Blended Learning on the Web (mit C.Frank, L.Suhl, S.Voß). Arbeitspapier, Technische Universität Braunschweig (2002)

20. Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts (mit D.Reiß, S.Voß). Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002), ICSC-Naiso Academic Press, Millet Alberta (2002), #100029-03-TR-026, 1–7 (2002)

21. NETRALOG: ein Forschungsprojekt zum wasserseitigen Containerumschlag mit Hilfe neuer Abfertigungsstrategien und unter Verwendung von Methoden der Bioinformatik (mit D.Steenken, D.Martinssen, F.Wölfer, H.Lührs, P.Ziehl, K.König, S.Voß). Hansa 138, 1, 72–78 (2001)

22. NETRALOG: Machbarkeitsprojekt für die Entwicklung und den Einsatz bioanaloger Algorithmen zur Steuerung des Horizontaltransports bei der Schiffsabfertigung auf dem Container Terminal Burchardkai; Schlußbericht (verschiedene Autoren). `http://edok01.tib.uni-hannover.de/edoks/e001/32197610X.pdf` (2001)

23. Konfiguration von Distributionslogistiknetzwerken unter Berücksichtigung kundenorientierter Lieferserviceanforderungen (mit J.Böse, A.Fink, K.Gutenschwager, G.Schneidereit). In: H.-J.Sebastian und T.Grünert (eds.), Logistik Management - Supply Chain Management und e-Business, Teubner, Stuttgart, 337–350 (2001)

24. Costs and Benefits of EDI Integration in Supply Chains and Networks (mit S.Voß). Arbeitspapier, Technische Universität Braunschweig (2001)

25. Einsatz bioanaloger Verfahren bei der Optimierung des wasserseitigen Containerumschlag (mit D.Martinssen, D.Steenken, S.Voß, F.Wölfer). In: H.-J.Sebastian und T.Grünert (eds.) Logistik Management - Supply Chain Management und e-Business, Teubner, Stuttgart, 377–388 (2001)

26. Vehicle Dispatching at Seaport Container Terminals Using Evolutionary Algorithms (mit J.Böse, D.Steenken, S.Voß.). In: R.H.Sprague (ed.) Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE, Piscataway (2000), DTM-IT: 1–10. Nominated for best paper award (2000)

27. Mining the Data from Experiments on Algorithms using Maximum Likelihood Clustering (mit D.L.Woodruff, R.Singh). In: E.Rolland, N.S.Umanath (eds.) Proceedings of the 4th INFORMS Conference on Information Systems and Technology, INFORMS, Linthicum, 235–254 (1999)

28. Maximum Likelihood Clustering of Large Data Sets Using a Multilevel, Parallel Heuristic. Diplomarbeit (1998)

Konferenzen mit eigenen Vorträgen

1. Ed-Media 2004 in Lugano (2004). Supporting the Authoring Process of Hierarchical Structured Learning Material (mit D.Reiß)
2. Logistikmanagement 2003 in Braunschweig (2003). Instructional design of interactive learning modules by example of a special flow shop problem (mit R.Eisenberg, I.Sassen, S.Voß)
3. Symposium on Operations Research – International Conference on Operations Research in Heidelberg (2003). SimTool: eine Plattform zum Design interaktiver Kurse im Bereich Simulation (mit I.Sassen, S.Voß)
4. Symposium on Operations Research – International Conference on Operations Research in Heidelberg (2003). Die adaptive virtuelle Lernumgebung SmartFrame (mit D.Reiß, I.Sassen, S.Voß)
5. VEAM IFIP Working Group 7.6 Workshop on Virtual Environments for Advanced Modeling in Ishikawa, Japan (2003). Teaching Meta-Heuristics and Simulation within VORMS (mit S.Voß)
6. Network Learning 2002 in Berlin (2002). Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts (mit D.Reiß, S.Voß)
7. Symposium on Operations Research 2002 – International Conference on Operations Research in Klagenfurth, Österreich (2002). Einsatz eines Simulationswerkzeuges im Zusammenhang mit Blended Learning (mit C.Frank, I.Sassen, L.Suhl, S.Voß)
8. Symposium on Operations Research 2002 – International Conference on Operations Research in Klagenfurth, Österreich (2002). Meta-Heuristiken in virtuellen Lernumgebungen (mit I.Sassen, S.Voß)
9. Workshop E-Learning in Wirtschaftsinformatik und Operations Research in Paderborn (2002). Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung (mit I.Sassen, S.Voß)
10. INFORMS Annual Meeting in Miami (2001). Customizing Interactive Learning Methods for the Internet (mit L.Suhl, H.G.Brunn, C.Frank, L.Tan)
11. INFORMS Annual Meeting in Miami (2001). A Blackboard Architecture Applied to Maximum Likelihood Clustering for Data Mining Computational Experiments (mit D.L.Woodruff)
12. INFORMS Annual Meeting in Miami (2001). Teaching Meta-Heuristics through Interactive Learning over the Internet (mit S.Voß)
13. Symposium on Operations Research 2001 in Duisburg (2001). Data Mining Applied to Computational Experiments of Algorithms (mit D.L.Woodruff)
14. 7^{th} Conference of the International Federation of Classification Societies in Namur, Belgien (2000). A Blackboard Architecture to Combine Meta-Heuristics with Heuristic Search for Maximum Likelihood Clustering (mit D.L.Woodruff)
15. Third Metaheuristics International Conference in Angra dos Reis, Brasilien (1999). A Blackboard Architecture to Combine Meta-Heuristics for Maximum Likelihood Clustering (mit D.L.Woodruff)

Anlage

Vorderseite Diplomzeugnis Wirtschaftsinformatik
Rückseite Diplomzeugnis Wirtschaftsinformatik
Michehl-Förderpreis-Urkunde
Bescheinigung Forschungsaufenthalt
Vorderseite Vordiplomzeugnis Informatik
Rückseite Vordiplomzeugnis Informatik

Aktuelles Photo

TECHNISCHE UNIVERSITÄT CAROLO-WILHELMINA ZU BRAUNSCHWEIG

FACHBEREICH PHILOSOPHIE, WIRTSCHAFTS- UND SOZIALWISSENSCHAFTEN

ZEUGNIS

ÜBER DIE

DIPLOMPRÜFUNG

HERR TORSTEN R E I N E R S

GEBOREN AM 07.03.1972 IN CUXHAVEN

HAT DIE DIPLOMPRÜFUNG IM STUDIENGANG

WIRTSCHAFTSINFORMATIK

MIT DER GESAMTNOTE

"SEHR GUT"

BESTANDEN

Das Zeugnis wurde auf Grund eines Studiums mit einer Regelstudienzeit von zehn Semestern erteilt und kann international dem Grad eines Master Degree gleichgesetzt werden.

ERGEBNISSE DER DIPLOMPRÜFUNG	BEURTEILUNGEN
1. Pflichtfächer	
Betriebswirtschaftliches Fach I: Operations Research	befriedigend
Betriebswirtschaftliches Fach II: Informationsmanagement	sehr gut
Informatik-Fach: Praktische Informatik	sehr gut
2. Wahlpflichtfach I: Medizinische Informatik	sehr gut
3. Wahlpflichtfach II: Genetik	sehr gut
4. Zusatzprüfungen: *	*
5. Diplomarbeit aus dem Gebiet:	"sehr gut"

Thema: "Maximum Likelihood Clustering of Data Sets Using a Multilevel, Parallel Heuristic"

Braunschweig, den 30. Oktober 1998

TECHNISCHE UNIVERSITÄT CAROLO-WILHELMINA BRAUNSCHWEIG 45

Dekan

Vorsitzender des Prüfungsausschusses

Notenstufen: sehr gut, gut, befriedigend, ausreichend.

Technische Universität
Carolo-Wilhelmina
zu Braunschweig

Die Technische Universität Carolo-Wilhelmina zu Braunschweig

vergibt den

Michehl-Förderpreis für Wirtschaftswissenschaften 1998

an

Herrn Dipl.-Wirtsch.-Inform. Torsten Reiners

für seine Diplomarbeit mit dem Titel

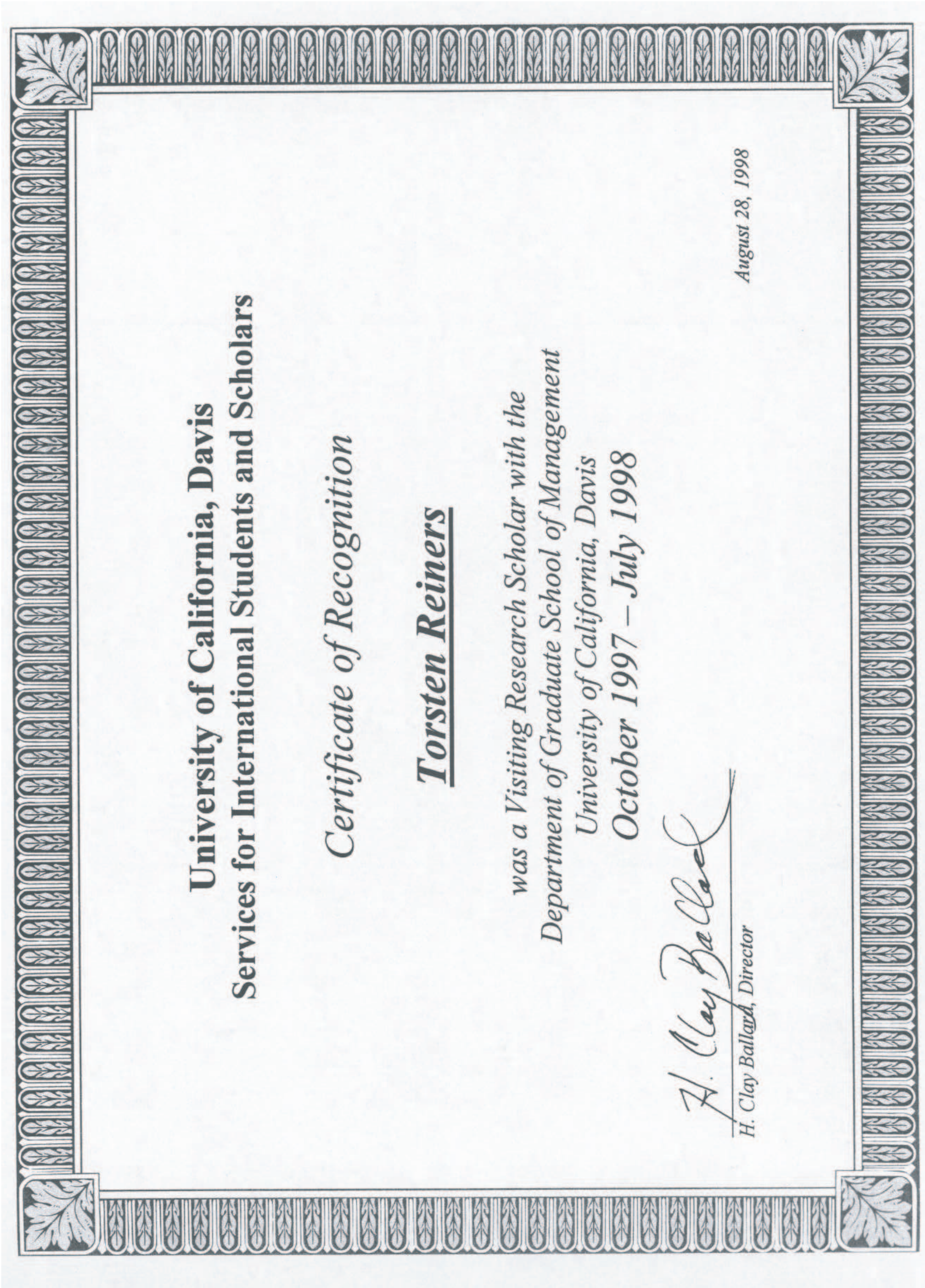
„Maximum Likelihood Clustering of Data Sets Using a Multilevel, Parallel Heuristic"

Der Preis wurde gestiftet von Herrn Peter Michehl, Michehls Atelier GmbH, Braunschweig, für hervorragende wirtschaftswissenschaftliche Arbeiten.

Braunschweig, den 10. November 1998

Prof. Dr. Bernd Rebe
Präsident
der Technischen Universität Braunschweig

Prof. Dr. Jens Jokisch
Geschäftsführender Leiter
des Instituts für Wirtschaftswissenschaften



University of California, Davis
Services for International Students and Scholars

Certificate of Recognition

Torsten Reiners

was a Visiting Research Scholar with the
Department of Graduate School of Management
University of California, Davis
October 1997 – July 1998

H. Clay Ballard, Director

August 28, 1998

TECHNISCHE UNIVERSITÄT CAROLO-WILHELMINA
ZU BRAUNSCHWEIG

FACHBEREICH MATHEMATIK UND INFORMATIK

ZEUGNIS

ÜBER DIE

DIPLOMVORPRÜFUNG

HERR TORSTEN R E I N E R S

GEBOREN AM 07.03.1972 IN CUXHAVEN

HAT DIE DIPLOMVORPRÜFUNG IM STUDIENGANG

INFORMATIK

MIT DER GESAMTNOTE

„BEFRIEDIGEND"

BESTANDEN

FACHPRÜFUNGEN	BEURTEILUNGEN
INFORMATIK I	GUT
INFORMATIK II	BEFRIEDIGEND
MATHEMATIK I	GUT
MATHEMATIK II	GUT
ELEKTROTECHNISCHE GRUNDLAGEN	AUSREICHEND
*	*

BRAUNSCHWEIG, DEN 21. AUGUST 1997

TECHNISCHE UNIVERSITÄT CAROLO-WILHELMINA 36 BRAUNSCHWEIG

D. Wätjen

VORSITZENDER DES PRÜFUNGSAUSSCHUSSES

NOTENSTUFEN: SEHR GUT, GUT, BEFRIEDIGEND, AUSREICHEND

Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computational Statistics & Data Analysis 47 (2004) 237–253
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Experiments with, and on, algorithms for maximum likelihood clustering

David L. Woodruff[a,*], Torsten Reiners[b]

[a] *Graduate School of Management, University of California Davis, Davis CA 95616, USA*
[b] *University of Hamburg, Institute of Information Systems, Von-Melle-Park 5, D20146, Hamburg, Germany*

Accepted 9 November 2003

Abstract

Elements of statistics, computer science, and operations research are connected with optimization heuristics as the catalyst. Heuristic search is used as a basis for a maximum likelihood clustering algorithm and it is demonstrated that clustering can be used to improve heuristic search algorithm performance. An important problem is described, a neighborhood structure for the problem is provided, and its value for heuristic algorithm development is demonstrated.

Keywords: Clustering; Data mining; Heuristic search; Tabu search; Simulated annealing

1. Introduction

This paper addresses heuristic optimization and model based statistical clustering from two perspectives. One of our goals is to examine the use of heuristic search as a basis for maximum likelihood clustering algorithms and we also demonstrate that clustering can be used to improve heuristic search algorithm performance. Statistical methods of finding clusters in data are described and applied to data generated by the parameters and performance of algorithms applied to a hard problem: the problem of finding clusters in data via maximum likelihood.

The analysis of algorithms applied to hard problems gives rise to a new set of hard problems, namely: How can the results of experiments on algorithms be exploited? In the sequel we describe some statistical methods of exploratory algorithm performance analysis. Often, one has a hypothesis concerning algorithm performance and experiments are conducted to test it. Certainly, this is an important activity. However, in this

* Corresponding author. Tel.: +1-530-752-0515; fax: +1-530-752-2924.
E-mail address: dlwoodruff@ucdavis.edu (D.L. Woodruff).

0167-9473/$ - see front matter
doi:10.1016/j.csda.2003.11.002

paper we are interested in using algorithm performance data to see if new relationships can be discovered rather than in testing hypothesized relationships.

The self-referential nature of our work requires modification of standard notation. A simple example is that the optimization literature typically uses x as a variable while the statistics literature typically employs x to represent data; since we span the fields, we eschew the use of x altogether to avoid confusion. We define the generic hard problem to which algorithms are applied as

$$\min_{\tau} \quad f(\tau) \quad \text{(P)}$$
$$\text{subject to:} \quad \tau \in \Xi,$$

where the set Ξ is intended to summarize the constraints placed on the decision vector τ. We refer to all data for the problem—the data that specifies the *objective function* $f(\cdot)$ and Ξ—as (P). In some cases it may be convenient to use a vector $\boldsymbol{P}$ that implies, rather than specifies, the problem instance. For the problem instances of interest to us here, there are no known algorithms that can find provably optimal solutions in a reasonable amount of time, so heuristic algorithms are employed that find reasonably good solutions fairly quickly.

We will refer to the heuristic algorithm parameters as ρ. We denote by $\boldsymbol{h}(\rho;\boldsymbol{P})$ the vector giving the results of applying a heuristic algorithm parameterized by ρ to the problem instance specified by (P). The results are a vector because at the very least they will include the best value of $f(\cdot)$ found and some measure of the time or effort required to find it and in many situations we will be interested in additional statistics.

Experiments consist of the execution of an algorithm and the result is a tuple ($\boldsymbol{P}$, ρ, $\boldsymbol{h}(\rho;\boldsymbol{P})$). It will be convenient to refer to one such 'record' as a vector $\boldsymbol{z}$ and the (arbitrarily ordered) collection of records from multiple runs as $\boldsymbol{Z}$. We might also refer to a set of records, $\boldsymbol{Z}$, as a dataset (or a *sample* or a *population* depending on the context). To be consistent with statistics literature on clustering, we will refer to the dimension of each vector as p and the number of vectors as n.

We cluster groups in a dataset, $\boldsymbol{Z}$, of points in $\Re^p$ using a maximum likelihood estimator (McLachlan and Basford, 1988). We describe a version of this type of estimator that explicitly allows for outliers. This methodology is applied to vectors generated by experiments on algorithms. In the next section, we develop the methodology while simultaneously providing connections with other research literature. In Section 3, we report on application to data generated by a simulated annealing (SA) algorithm as parameterized by Johnson et al. (1989) applied to the problem of finding clusters. Since the performance of this algorithm with respect to its parameters has been studied extensively it provides a useful demonstration of the methods. This data mining is continued when we do the same thing using reactive tabu search (RTS) in Section 4. The final section offers conclusions and directions for further research.

2. Maximum likelihood clustering

2.1. Estimators

A collection of points in $\Re^p$ (i.e., vectors) are presented as data and we ask if there are naturally occurring clusters. We begin by assuming that we are looking for some number of clusters g, which can be varied by the analyst (or 'data miner'). But we do not assume that we know a priori an appropriate distance metric. This is in contrast to many popular clustering objectives, such as minimum sum-of-squares that assumes the validity of a Euclidean metric. A recent paper by Pacheco and Valencia (2003) explores of the use of heuristic optimization techniques for this problem.

There is a growing literature devoted to clustering when no metric is assumed, but instead a statistical model is assumed (see, e.g., Hennig and Christlieb, 2002; McLachlan, 1982; McLachlan and Peel, 2000). In this model the data are generated in two stages: (1) a cluster distribution is selected from among g possibilities and (2) an observation is drawn from the selected cluster. The notation π_i is used to denote the probability of selecting cluster i. Cluster i is assumed to be modeled by a multivariate normal distribution with density f with mean and covariance matrix $\boldsymbol{\theta}_i = (\mu_i, \Sigma_i)$. This results in the mixture likelihood

$$L(\boldsymbol{Z}; \boldsymbol{\theta}) = \prod_{j=1}^{n} \left[\sum_{i=1}^{g} \pi_i f(z_j, \boldsymbol{\theta}_i) \right], \tag{1}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_g, \pi_1, \ldots, \pi_g)$.

A thorough examination of criteria based on the likelihood is given by Banfield and Raftery (1993). Their paper proposes a number of criteria that maximize the likelihood conditional on a clustering, under a number of assumptions about the relative sizes and shapes of the clusters. A popular method is to solve problem (MINW) (Friedman and Rubin, 1967), which finds the clustering that minimizes the determinant of a multiple of the pooled covariance matrix det($\boldsymbol{W}$), where

$$\boldsymbol{W} = \sum_{i=1}^{g} \boldsymbol{W}_i,$$

$$\boldsymbol{W}_i = \sum_{j=1}^{n_i} (z_{ij} - \bar{z}_i)(z_{ij} - \bar{z}_i)^{\mathrm{T}},$$

where z_{ij} is the jth vector assigned to the ith cluster, $\bar{z}_i$ is the mean of the vectors assigned to the ith cluster and n_i is the number of points assigned to cluster i. This corresponds to maximum likelihood under the assumption that the vectors z are multivariate normal with homogeneous but otherwise unrestricted cluster covariances. Algorithms proposed for this minimization include hierarchical agglomeration (Murtagh and Raftery, 1984; Ward, 1963) and local search (Späth, 1985). We cast this minimization into the framework given in the introduction by giving the vector τ two indexes. We let τ_{ij} be one if the jth vector in the population or sample has been assigned to cluster i and zero otherwise. The objective function is then det($\boldsymbol{W}$).

We have no particular reason to expect homogenous cluster covariance structures, so we make use of an objective that is similar from a computational standpoint, which is

$$\sum_{i=1}^{g} n_i \log \det\left(\frac{\boldsymbol{W}_i}{n_i}\right).$$

The minimum corresponds to a maximum likelihood under the assumption of heterogenous covariance matrices. It was first given by Scott and Symons (1971) and adjusted by Banfield and Raftery (1993). Call the problem with this objective function (MIND).

For either objective function, we need to constrain the τ_{ij} to be either zero or one. To ensure that every point is in some cluster we also require that $\sum_{i=1}^{g} \tau_{ij} = 1$ for all $j = 1, \ldots, n$. Refer to these as the *placement* constraints. In addition, we need to constrain each cluster to contain some minimum number of points, H. Clearly, $H > p$ or else the determinant will be undefined. This can be restated as $\sum_{j=1}^{n} \tau_{ij} \geqslant H$ for all $i = 1, \ldots, g$. We refer to these as *size* constraints.

When used in a data mining application such as the one that we have proposed, it is extremely unrealistic to assume that all data comes from a population that has contributed enough points to form a cluster that can be identified. Banfield and Raftery (1993) have proposed maximum likelihood clustering with a cluster numbered $g + 1$ that is assumed to come from a multi-variate Poisson. However, we have no reason to expect a Poisson process to contribute, so we need a more general model. Hence, we make use of a pure *outlier* model that first appeared in Reiners (1998).

We notice that (MINW) is a multiple cluster generalization of the minimum covariance determinant (MCD) estimator of Rousseeuw (Hampel et al., 1986; Rousseeuw, 1985; Rousseeuw and Leroy, 1987; Woodruff and Rocke, 1994) for location (mean) and shape (covariance) of one cluster in the presence of outliers. This causes us to add a cluster numbered $g + 1$ to protect against up to some number of outliers T. Starting with problem (MIND) we change only the placement constraint so that the summation is to $g + 1$ rather than g and we add a constraint $\sum_{j=1}^{n} \tau_{ij} \leqslant T$ for $i = g + 1$. We refer to this as the *outlier* constraint. In particular, we retain the objective function that sums over the first g groups. The effect is that the points labeled as *outliers* (i.e., assigned to cluster $g + 1$) "do not count." It is easy to prove that the $g + 1$ group will always have T members for optimal solutions so the outlier constraint can be rewritten to be an equality constraint. This has some computational advantages.

Refer to the (MIND) problem modified to allow for T outliers as problem (MINO). So to summarize: the data for problem (MINO) consists of the tuple (p, n, Z, g, H, T) and a solution to an instance of the problem is implied by giving a vector τ that satisfies the constraints and minimizes the objective function. The apparent redundancy between p, n and Z is useful for visual inspection. A full mathematical programming formulation of (MINO) is given in the Appendix. This formulation provides an immediate proof of Remark 1, which leads directly to the NP-Hardness of the (MINO) when combined with the proof that the MCD is NP-Hard provided by Bernholt and Fischer to appear.

Remark 1. The objective function for (MINO) with $g = 1$ and $T = n - \lfloor (n + p + 1)/2 \rfloor$ imposes the same order on feasible solutions as the objective function for the (MCD). Furthermore, all (MCD) instances can be given as (MINO) instances in this way.

2.2. Neighborhood structures

There are no exact algorithms for (MCD) and (MINW) problems that can be executed on moderate sized data sets. Hence, these NP-hard problems are of the sort that must be addressed using heuristics that find good solutions in a reasonable amount of time. For computational reasons, neighborhood based search algorithms are attractive for the problems in the (MCD) or (MINW) family.

Neighborhoods are based on *moves* from one solution to another. All of the solutions that can be reached from a given solution in one move are said to be in the neighborhood of the solution. We use the notation $\mathcal{N}(\tau)$ to indicate the set of solutions that are neighbors of a solution τ.

For (MINW) and (MIND) an obvious neighborhood is one where a point is moved from one group to another. For solutions where the size constraints are not binding, the neighborhood has $(g-1)n$ solutions. There are fewer neighbors of solutions for which one or more of the size constraints are binding. This neighborhood is used by Späth (1985) as well as Coleman and Woodruff (2000) in first-improving local search algorithms.

The neighborhood structures are more complicated for (MINO) due to the presence of the outlier group. Moves involving the outlier group always involve two points: one coming into the outlier group and one leaving. So when none of the size constraints are binding, the neighborhood has $(n-T)(g-1)+(n-T)T$ solutions. For a solution that causes s size constraints to bind, the neighborhood size is $(n-T-sH)+(n-T)T$. We cannot use first improving neighborhoods that have been developed for the older (MINW) and (MIND) because after a move involving the outlier group, continuation through the neighborhood is undefined. We now describe in detail the computations needed to support local search for the estimator defined by (MINO).

Once the objective function has been computed for a solution, one can anticipate the effect on the objective function of all moves with far less computational effort than computing the objective function given an entirely new solution vector. Update formulas for the covariance determinant are used as described in Hawkins (1994). Also, for steepest descent algorithms, the best swaps involving outliers can be precomputed.

The feasible moves in a neighborhood can be classified into two types:

(1) Move not involving outlier group: This move investigates the possibility that the jth data vector z_j has been misclassified to cluster i instead of i', $i' \neq i$ and $i' = 1,2,\ldots,g$. That is move z_j from cluster i to a different cluster i', provided such a move satisfies the size constraints. This involves setting $\tau_{ij}=0$ and $\tau_{i'j}=1$ and decrement n_i by 1 and increment $n_{i'}$ by 1.

(2) Move involving outlier group: This move investigates the possibility that the jth data vector z_j, currently assigned to cluster i, should be an outlier and a data vector z_k, currently assigned to the outlier group ($\tau_{ik}=0$, $\forall i$), should be a member of the cluster i'. Here there are two possibilities:

   (a) $i'=i$. That is move z_j from cluster i to the outlier group and move a member of the outlier group z_k back to cluster i. This involves setting $\tau_{ij}=0$ and $\tau_{ik}=1$.

   (b) $i' \neq i$. That is move z_j from cluster i to the outlier group and move a member of the outlier group z_k to a different cluster i'. This involves setting $\tau_{ij}=0$ and $\tau_{i'k}=1$.

Implementation of local search heuristics can be broken broadly into two parts, first computing the objective function for the current solution and second evaluating all feasible moves in the neighborhood of the current solution and making the best move if there exist improving moves.

2.2.1. Evaluation of the objective function

We introduced

$$\sum_{i=1}^{g} n_i \log \det\left(\frac{W_i}{n_i}\right)$$

as the objective function to be evaluated, but it turns out to be useful to compute this by augmenting the data vectors z_k with a 1 (to adjust the matrix W_i for the mean) defining $y_k = (1; z_k^{\mathrm{T}})^{\mathrm{T}}$, $k = j_1, \ldots, j_{n_i}$ and writing the partitioned matrices

$$Y_i = (y_{j_1}, y_{j_2}, \ldots, y_{j_{n_i}})$$

and

$$Z_i = (z_{j_1}, z_{j_2}, \ldots, z_{j_{n_i}}),$$

where we let $J_i = (j_1, j_2, \ldots, j_{n_i})$ be the set of indices of the n_i elements assigned to cluster i in the current solution (see, e.g., Hawkins, 1994). Then,

$$Y_i Y_i^{\mathrm{T}} = \begin{pmatrix} n_i & n_i \overline{z_i}^{\mathrm{T}} \\ n_i \overline{z_i} & Z_i Z_i^{\mathrm{T}} \end{pmatrix}.$$

Now, defining $C_i = Y_i Y_i^{\mathrm{T}}$,

$$\begin{aligned} \det(C_i) &= n_i \det(Z_i Z_i^{\mathrm{T}} - n_i \overline{z}_i \overline{z}_i^{\mathrm{T}}) \\ &= n_i \det\left(n_i \sum_{k=j_1}^{j_{n_i}} (z_k - \overline{z_i})(z_k - \overline{z_i})^{\mathrm{T}} \right) \\ &= n_i \det(W_i) \\ &= n_i^{p+1} \det\left(\frac{W_i}{n_i}\right). \end{aligned}$$

Hence, the objective function is equal to

$$\sum_{i=1}^{g} n_i \log \det\left(\frac{C_i}{n_i^{p+1}}\right). \tag{2}$$

Now, since

$$C_i = \sum_{k=j_1}^{j_{n_i}} y_k y_k^{\mathrm{T}}, \tag{3}$$

this gives an opportunity for efficiently computing the covariance matrix C_i and hence the objective function. This is done by taking each data vector z_j (of dimension p), creating a $(p+1)$-dimensional vector $y_j = (1 : z_j^{\mathrm{T}})^{\mathrm{T}}$, and calculating a $(p+1) \times (p+1)$ dimensional matrix $y_j y_j^{\mathrm{T}}$. The covariance matrix C_i is then constructed by incrementally adding $y_j y_j^{\mathrm{T}}$ for all data points of the ith cluster. Once the matrices C_i are constructed, their determinants are calculated. With the membership n_i of each cluster and the dimension p of the data, calculation of the objective function using Eq. (2) is straightforward.

2.2.2. *Evaluating objective function for a move and making the best move*

Every move involves adding and/or removing a data point from a cluster. Anticipating the changed objective function for such a move involves updating/downdating the determinants of the covariance matrices of the affected clusters. Eq. (3) can be used directly for this purpose, however, this requires the determinants of the matrices to be recomputed for each move being considered. Since calculation of the determinants is an expensive operation the following update formulas (see, e.g., Späth, 1985) are used:

$$\det(C_i + \beta y_k y_k^{\mathrm{T}}) = \det(C_i)(1 + \beta y_k^{\mathrm{T}} C_i^{-1} y_k),$$

where $\beta = +1$ when y_k is being added, and $\beta = -1$ when y_k is being removed from the cluster i.

For the case of a swap (case 2a above) where an element y_j is removed from cluster i and a different element y_k is added to the same cluster, the update formula is

$$\det(C_i + y_k y_k^{\mathrm{T}} - y_j y_j^{\mathrm{T}})$$
$$= \det(C_i)[(1 - y_j^{\mathrm{T}} C_i^{-1} y_j)(1 + y_k^{\mathrm{T}} C_i^{-1} y_k) + (y_j^{\mathrm{T}} C_i^{-1} y_k)^2].$$

These formulas provide the updated determinants for the affected clusters that are then used to calculate the objective function for the move. The objective function is evaluated for each feasible move in the neighborhood, and the best move determined. Making the selected move involves setting the τ_{ij} appropriately. In addition, the inverses of the covariance matrices need to be maintained and updated for the evaluation of the next move. The following standard formulas are used to update or downdate these inverses:

$$(C_i + \beta y_k y_k^{\mathrm{T}})^{-1} = C_i^{-1} - \frac{\beta C_i^{-1} y_k y_k^{\mathrm{T}} C_i^{-1}}{1 + \beta y_k^{\mathrm{T}} C_i^{-1} y_k}.$$

This inverse of $(C_i + \beta y_k y_k^{\mathrm{T}})$ exists whenever $1 + \beta y_k^{\mathrm{T}} C_i^{-1} y_k \neq 0$.

It is well known that computational errors accumulate when using these formulas repeatedly, particularly the downdating formula. Consequently, the algorithms recompute the covariance structures after every 10 moves.

2.2.3. *Precomputation of a change in the objective function for moves involving outliers*

In moves that involve outliers, we need to determine the best outlier to move back into one of the clusters and the cluster to which to move it. For this we need to evaluate the change in objective function for each of the T outliers moving to each of the g clusters, and then determine the outlier and cluster corresponding to the minimum change. This involves making Tg computations of determinants. Since for a given solution this computation remains the same for all moves involving outliers, this computation is done only once. An exception to this is when the outlier is to move to the group from which a data point is moving to the outlier group. In this case there is some small inaccuracy which could result in a sub-optimal choice of outlier to move, but for the sake of efficiency this is acceptable (with data of any significant size the chance of this would be negligible). When a move involving outliers is made, recomputation of the determinants is done only for the affected clusters and the new outlier. When a move that does not involve outliers is made, recomputation is done only for the two affected clusters.

3. Application of, and to, Johnson et al. SA

Armed with a neighborhood structure we can employ local search heuristics. To illustrate the use of the clustering methods, we first employ and study a version of simulated annealing, which is a general purpose heuristic defined relative to a neighborhood structure and a *cooling schedule*. At each iteration k, a solution, τ', is selected at random from the neighborhood of the incumbent solution, $\tau^{(k)}$. Let Δ be $f(\tau') - f(\tau^{(k)})$. The move is said to be *accepted* and $\tau^{(k+1)} \leftarrow \tau'$ if $\Delta < 0$ or if

$$e^{-\Delta/c} > R,$$

where c is the "temperature" that is controlled by the cooling schedule and R is a pseudo-random deviate drawn from a distribution that is uniform on (0,1). Otherwise, the move is said to be *rejected* and $\tau^{(k+1)} \leftarrow \tau^{(k)}$. The solution vector τ that satisfies the constraints and for which $f(\tau)$ is the lowest seen to that point in the search is retained as the "current champion." For our applications, the initial solution $\tau^{(0)}$ is selected at random.

The cooling schedule developed and parameterized by Johnson et al. (1989) is popular and has a number of properties that make it worthy of study using clustering algorithms. The basic idea of the cooling schedule is a geometric progression of temperatures, with multiple random trials at each temperature. After some number, L, of moves have been generated and tested the temperature is reduced.

The Johnson et al. algorithm has four parameters: $\rho = (\mathrm{TF}, \mathrm{IP}, \mathrm{MP}, \mathrm{SF})$ with the following meaning as outlined in their paper:

TF—TEMPFACT: When the temperature is to be reduced, the current temperature is multiplied by TF to give the new temperature.

IP—INITPROB: Based on an abbreviated trial annealing run, a temperature is found at which the fraction of accepted moves is approximately IP, and this is used as the starting temperature.

MP—MINPERCENT: This is used to test whether the annealing process is *frozen* and should be terminated. A counter is incremented by 1 each time a temperature is completed for which the percentage of accepted moves is MP or lower. The counter is reset every time a new champion is found. If the counter ever reaches 5, the algorithm is terminated. The value of 5 is actually a fifth parameter in disguise with a fixed value. Due to strong interactions with MP, leaving this parameter fixed at 5 seems reasonable.

SF—SIZEFACTOR: We set $L=\lambda^*$SF, where λ is approximately the average neighborhood size (e.g., $\lambda=(n-T)(g-1)+(n-T)T$).

The parameter values recommended by Johnson et al. are (0.95, 0.4, 2, 16). Extensive experiments across a broad range of problems indicates that the CPU time and best objective function value seen vary in fairly consistent ways as a function of parameters. This consistent performance makes us suspect that the algorithm represents a good opportunity to exploit clustering.

3.1. Experiments

Our experiments are done with a fixed data set and fixed clustering parameters, so that only the four SA algorithm parameters vary. The data set that we make use of is the famous Fisher Iris Data (FID) due to Fisher (1936) which has 150 points with four values each. We know that it has three clusters, but for these simulations we pretend that we do not know what they are. The problem instance is characterized by the vector ($p=4$, $n=150$, FID, $g=3$, $H=35$, $T=2$)

We execute the Johnson et al. SA algorithm with randomly selected parameter vectors and observe the CPU time in seconds, CPU, and the best objective function value seen for (MINO), f^*. I.e., $h(\cdot)=(\text{CPU}, f^*)$. The pseudo-random number stream is also varied for each run. For the experiments of interest to us the SA parameter values were IP uniform on [0.1, 0.7], TF uniform over a log scale on [0.9, 0.99], SF uniform over a log scale on [8, 128], and MP uniform on [3, 8], which were set somewhat arbitrarily with an objective of having a reasonable running time for the experiments with reasonable coverage of the parameters. The experiment consisted of 200 runs. Once the runs of SA on the Fisher Iris Data are complete there is a new data set that consists of the four SA parameters and the results given as CPU and f^*. Refer to this data set as SA1, which is characterized by $p=6$, $n=200$; each record in the data set gives an instance of (TF, IP, MP, SF, CPU, f^*) .

The data set has a mean of (0.948, 0.389, 5.475, 40.945, 325.315, −1647.390) with standard deviations (0.026, 0.169, 1.665, 31.390, 306.452, 12.761). The correlation matrix (lower half without the diagonal) is

	TF	IP	MP	SF	CPU
IP	0.00				
MP	−0.12	−0.06			
SF	0.05	−0.01	0.10		
CPU	0.63	0.49	−0.12	0.01	
f^*	−0.16	0.10	0.37	0.11	−0.13

where the *correlation* between i and j is defined as

$$\frac{W_{ij}}{\sqrt{W_{ii}W_{jj}}}.$$

Those values that are statistically significant at 95% (one tail) are shown in bold typeface. Given the large range over which the parameters are varied, the correlations are not very surprising. CPU time has a significant positive correlation with the values of IP and TF (the correlations are 0.63 and 0.49). It is a little bit surprising that time is essentially uncorrelated with quality (-0.13) but this is no doubt due to such a broad set of parameter values. Given this, one might expect that there are clusters within the data with correlations that are more consistent with a priori expectations about the way the algorithm should work (and therefore, also clusters that are even less consistent). In the next subsection we explore these clusters.

3.2. Data mining methods

Data mining is a somewhat tedious and not a very mathematically satisfying activity. Our goal is to discover interesting clusters. Such a qualitative objective cannot give rise to a fully quantitative and straightforward process. Rather than proceeding in Gauss-like fashion from assumptions to results, one tries a lot of parameters for the clustering problem and examines the results to see if they might be "interesting." As noted in Cabena et al. (1994):

> The analysis of results is inseparable from the data mining step in that the two are typically linked in an interactive process. The exploratory nature of data mining ensures that this is always the case.

Hence, in this context, the word "interesting" clusters are those that ultimately lead to potentially useful hypotheses or ideas. In order to find such clusters, we often looked for those with good separation, but not always.

During the process of mining the results of SA on the iris data, we vary the minimum cluster size, H, the number of clusters, g, and the number of outliers, T. We use SA to find good clusters in the SA performance data but resist the temptation to recurse again and cluster the data mining runs themselves as part of our study of SA. Dozens of parameter combinations are tried and each time a clustering of the SA results is found by SA, we examine it to see if it is "interesting."

To improve the clarity of exposition, we can make use of prior knowledge about the iris data. We know what a good clustering of the iris data is because we know in advance which types of flowers generated which data (except, of course, for the chance that some of the flowers were originally misclassified). There seem to be a handful of local minima for (MINO) applied to the iris data with our neighborhood structures and a wide range of parameters. We can characterize those local minima with less than eight classification errors as "good" results and the rest of the local minima as "bad." It turns out that good classifications have objective function values at and below about -1650. Hence, clusters of SA performance data can be qualitatively characterized as "good" or "bad" if they happen to have almost all, or all objective function values above or below this value respectively. As a result of intuitively appealing parameterization of SA developed by Johnson et al. and the extensive results reported for this algorithm we also know what sort of correlations should be seen when the algorithm is working well.

3.3. Clusters found

To mine the data from the experiments, we must sift through far too many clusterings to report. Ultimately, the clustering that turned out to be most interesting was found with the parameters $H=30$, $g=2$, and $T=10$. Since $p=6$, there are enough points designated as outliers to treat them as a small cluster for the purpose of reporting results. These parameters result in a clustering with 87 points in the first cluster and 103 in the second.

Bear in mind that each data record being clustered in SAI describes a run of SA to cluster the iris data using (TF, IP, MP, SF, CPU, f^*). The clusters of interest are

Cluster 1:

Mean: (0.9707, 0.359, 5.2, 41.8, 406, −1649.8)

Standard deviation: (0.01, 0.16, 1.70, 27.62, 305.22, 12.83)

Correlation matrix:

	TF	IP	MP	SF	CPU
IP	**0.24**				
MP	0.02	**−0.20**			
SF	0.02	−0.04	**0.21**		
CPU	**0.60**	**0.75**	**−0.21**	−0.04	
f^*	−0.08	0.00	**0.48**	0.06	0.11

Cluster 2:

Mean: (0.9264, 0.408, 5.7, 36.7, 199, −1644.8)

Standard deviation: (0.01, 0.17, 1.63, 29.33, 105.70, 12.58)

Correlation matrix:

	TF	IP	MP	SF	CPU
IP	0.10				
MP	0.02	0.03			
SF	0.01	0.02	0.00		
CPU	**0.42**	**0.83**	0.07	0.08	
f^*	0.09	**0.15**	**0.29**	**0.26**	0.07

Outliers:

Mean: (0.9692, 0.446, 5.9, 76.5, 924, −1652.9)

Standard deviation: (0.03, 0.16, 1.45, 55.99, 644.75, 7.78)

Correlation matrix:

	TF	IP	MP	SF	CPU
IP	**0.41**				
MP	**−0.49**	**−0.47**			
SF	**−0.71**	**−0.21**	**0.25**		
CPU	**0.95**	**0.52**	**−0.31**	**−0.76**	
f^*	**0.38**	**0.28**	**−0.17**	**0.22**	**0.25**

The points designated as "outliers" are the most interesting. This designation is a little bit strange. They are outliers in the sense that they are not in either of the main groups. Note that the mean values of the four SA parameters are near the recommended values for Johnson et al. and that the correlations with time and objective function value are mostly pro-intuitive and statistically significant at 0.95 (one-sided). Based on the work of Johnson et al. and other experience with SA, these correlations represent exactly the sort of behavior we would expect to see when the algorithm is working properly. In fact, all of the points have low objective function values.

The first cluster also contains some points with low objective function values, but only a few. Many of the correlations are what we would expect from good SA performance, but many others are insignificant. For the second cluster, most of the correlations that we would expect to find are insignificant (e.g., the correlation between SF and CPU is nearly zero). It is no surprise, then, that this cluster is composed almost entirely of points that correspond to bad clusterings of the iris data.

For the outlier group shown above, the correlations are quite similar to those predicted by experience with successful annealing and furthermore, the mean parameter values for this small "cluster" happen to be very near the values recommended by Johnson et al. Regardless of the values of g and H that were used, we were never able to find large clusters with good objective function values and correlations that conform with the expectations of successful simulated annealing. This application of data mining causes us to believe that SA may not be the best algorithm for this problem so we now explore the use of RTS.

4. Application of, and to, reactive tabu search

reactive tabu search (Battiti and Tecchiolli, 1994) is a variant of Tabu search (Glover, 1989; Glover and Laguna, 1997). The particular implementation of RTS that we describe here is the one in the Woodruff (1997) class library and differs slightly from the version described by Battiti and Tecchiolli.

A tabu search proceeds from trial solution τ^k to a trial solution

$$\tau^{k+1} \in \mathscr{A} \subset \mathscr{N}(\tau^k),$$

where $\mathscr{A}$ is the set of admissible moves (for simplicity, we show it without parameter subscripts). The set $\mathscr{A}$ is formed by removing from $\mathscr{N}(\tau^k)$ any solutions that would reverse attributes of the most recent κ. The moves that are in $\mathscr{N}(\tau)$ but not in $\mathscr{A}$ are said to be tabu. The value of κ is changed dynamically during reactive tabu search (RTS). It is increased if a repeated solution is detected and decreased slowly over time and/or if all moves are tabu. The initial value of κ for our implementation is n/IK, where IK is a parameter. For our experiments, the factors controlling the amount of increase (TI) and decrease (TD) were not interesting (for example, a 10% change works well enough).

The distinguishing feature of RTS is its reliance on long-term memory that is concerned primarily with repeated solutions. If too many solutions (given by the parameter

RP) are repeated too often (given by the parameter CH) then the search is assumed to be trapped. The metaphor is of a chaotic attractor.

In addition to recording repeated solutions, the number of moves between repeated solutions is also stored (recorded in the variable Running_Average). This value is assumed to be proportional to the size of the domain of attraction of the chaotic attractor. A parameter that sets the maximum cycle length that will influence Running_Average as well as cause changes in the tabu list length is called CM. When RP solutions have been repeated CH times, the search executes a large number of random moves (with the number proportional to Running_Average) in order to try to escape the domain of attraction of the chaotic attractor. In our implementation, we simply create a new, random solution to escape rather than making random moves. In the implementation we employ, the tabu list length is reset to n/IK after an escape.

As a basis of comparison, note that other forms of tabu search likewise make use of frequency-based memory as a foundation for diversification strategies (to drive the search into new regions), but the frequencies are generally recorded for attributes of solutions, such as values of variables and functions, rather than for complete solutions. In addition, the trigger for diversification in other forms of TS typically consists of either an iteration count or a limit on the number of iterations without finding a new best solution.

4.1. Simulated data

Unless it is restricted to use so little time that it does not do random escape moves once or twice, RTS always finds the best known solution for the problems of clustering the iris data. Such short runs require orders of magnitudes less CPU time than SA runs and RTS without escapes is hardly RTS so we conclude that the iris data will not be good test bed. We turn to simulated data.

Based on arguments given in Coleman and Woodruff (2000) suggesting that clusters arranged on a line are the hardest for (MIND) algorithms to find, we create clusters with a standard multivariate normal distribution along a main coordinate diagonal. We measure the distance between clusters in terms of the unit of measurement $\sqrt{\chi^2_{p;0.001}}$, which is more or less the radius of the sphere around the mean that contains almost all the good points. This places the outliers at the correct distance out a diagonal. For the experiments reported here, the first cluster is centered at the origin, the second is created by placing the mean of the p components at $1.5\sqrt{\chi^2_{p;0.001}/p}$ and the mean of each of the components for the third cluster at $4\sqrt{\chi^2_{p;0.001}/p}$. The data set has 400 points in dimension 10 with the number of points in each cluster roughly equal. We refer to this simulated data set at SLD.

4.2. Experiments and clusters found

The simulated data results in a problem instance that is characterized by the vector ($p = 10$, $n = 400$, SLD, $g = 3$, $H = 30$, $T = 0$). The RTS algorithm was executed on this data many times, each time with a different randomly selected starting point. For some of the executions (259 of them) parameters were set by a person attempting to study the performance of the algorithm. For most of these runs CH was correlated with RP and/or TD was negatively correlated with TI by the experimenter. To augment this data, we executed RTS 200 times with parameters selected randomly from the uniform distributions: CH and RP on [1,6], IK on [3,40], TD on [0.60, 0.95], TI on [1.05, 1.35], and CM on [200,600]. Each execution is terminated after 1200 iterations because preliminary tests indicated that this could result in a correct clustering, but would not always do so.

The run results were gathered into a data set (or "mined") for which each record contains (CH, RP, IK, TD, TI, CM, CPU, f^*). The mean parameter values of these data is (CH, RP, IK, TD, TI, CM) = (3.6, 3.6, 22.1, 77.7, 119.5). Of the 459 runs, 142 resulted in the zero classification errors and the lowest f^* seen.

The data set immediately bifurcates into two large clusters: the 142 with the best f^* value and the other with all other objective function values. To be consistent with the last section, we can refer to these as the "good" cluster and the "bad," even though many of the results in the so-called bad cluster are not that bad. To perform exploratory data analysis (or data mining), we can then attempt to find interesting clusters from within these two clusters. Although hundreds of clusterings were examined for the bad cluster, no clusters were found that caused use to form a new hypothesis or draw any new conclusions so we turn our attention to the good cluster.

The good cluster is coplanar when the f^* value is included so we remove it for further processing. When clustering the resulting data, the interesting clusterings observed were well represented by the result when we looked for two clusters and allowed ten outliers. The interesting interactions were for CH, RP, IK; all other correlations were statistically insignificant. The portions of the correlation matrices of interest were:

		CH	*RP*
1 :	*RP*	**1.00**	
	IK	**0.36**	**0.36**
	CH	*RP*	
2 :	*RP*	−0.18	
	IK	0.07	−0.02

This leads to an interesting family of hypothesis that we would not have considered without the benefit of data mining: the three parameters CH, RP, and IK could perhaps be combined into one parameter. This is suggested (but by no means proven) by the fact that for good runs when CH and RP were perfectly correlated, the correlation with IK is also positive. We use the word *family* because the data mining results suggest an infinite variety of exact, well formed statistical hypothesis. This is the goal of data mining: to suggest hypotheses. So in that sense, we suggest that we have demonstrated a useful application of cluster analysis to the problem of exploratory algorithm analysis.

Although this paper is about clustering rather than hypothesis testing, we did conduct experiments on additional datasets to explore the hypothesis. We found no statistically significant difference between the algorithm with four parameters and the algorithm with six. In fact, the algorithm with four had slightly better average performance. It is beyond the scope of this paper to describe and defend new tabu search algorithms, but we have demonstrated that data mining can be applied to heuristic algorithm performance data and important hypothesis can be discovered.

5. Conclusions and directions for further research

In summation, we have described an important problem, demonstrated its value for heuristic algorithm development and provided a neighborhood structure for the problem. The self-referential nature of our work highlights the directions for further research. Research opportunities exist both in the use of data mining for algorithm development and algorithm development to support data mining and statistics.

Heuristic optimization methods gain popularity as more software becomes available (see, e.g., (Voß and Woodruff, 2002)) and as researchers discover the myriad of potential applications. Many of these applications are in statistics where estimators are often defined in terms of an optimization problem. Computationally difficult estimators can be addressed using heuristic methods.

As we have shown here, maximum likelihood based clustering tools can be useful during the exploratory phase of algorithm development. Sophisticated algorithms applied to complex problems are hard to analyze. Although some hypothesis are straightforward and suggest themselves to anyone working on the problem, others are surely waiting to be discovered by data mining tools. Unfortunately, there is no meta-algorithm for data mining. The analyst using clustering must examine many groupings in order to discover hypotheses worth testing. Our work here adds to the data mining literature suggesting that it is often worth the trouble.

We demonstrated the application of clustering/mining to a particular simulated annealing design where we found that clusters with correlations roughly corresponding to those predicted by the algorithm designers corresponded to good algorithm performance. The fact that these clusters were small, suggested that other algorithms might perform better. We also applied clustering to reactive tabu search. This led to an interesting family of hypothesis that we would not have considered without the benefit of data mining, namely that the three parameters CH, RP, and IK could perhaps be combined into one parameter. This finding suggests that perhaps additional experiments on simulated annealing would result in the discovery of a useful hypothesis. We studied the algorithm applied to a single problem multiple times, perhaps mining the data generated by application to multiple problems would be productive.

In the current excitement over data mining, the fruit is still on low branches. Practitioners are able to extract useful hypothesis from data using straightforward algorithms. Here again, there are surely interesting facts that can be culled from data using more sophisticated algorithms in support of more sophisticated methods such as maximum likelihood clustering. The advantage of model based statistical methods is that probabilistic statements can be made about the results. The perceived drawback is that they often result in difficult optimization problems. This presents an important research opportunity at the interface between statistics, computer science, and operations research.

Acknowledgements

An earlier version of this work appeared in the proceedings distributed at the INFORMS Conference on Information Systems and Technology, May, 1999; we appreciate the comments of the participants and discussants. This work was supported by a grants from the National Science Foundation, the Du Pont Corporation and NIH Grant P42 ES04699.

Appendix A. Mathematical programming formulation of (MINO)

The description of the estimator and the corresponding minimization problem as given in Section 2.1 makes use of notation common in the statistics literature. The notation also happens to be useful for describing the neighborhood structures. In the interest of rigor and to provide connections with other literature, a mathematical programming version is given here.

We have as data the number of groups, g, a minimum cluster size, H, number of outliers to guard against T, and n points in R^p with each point given as Z_j for $j=1,\dots,n$. Our objective is to select $\tau_{ij}, i=1,\dots,g+1,\ j=1,\dots,n$ so as to

$$\text{minimize} \sum_{i=1}^{g} n_i \log\det\left(\frac{W_i}{n_i}\right) \quad \text{(MINO)}$$

subject to

$$W_i = \sum_{j=1}^{n} (\tau_{ij}Z_j - \bar{z}_i)(\tau_{ij}Z_j - \bar{z}_i)^{\mathrm{T}}, \quad i=1,\dots,g,$$

$$\bar{z}_i = \left(\sum_{j=1}^{n} \tau_{ij}Z_j\right) / n_i, \qquad i=1,\dots,g,$$

$$n_i = \sum_{j=1}^{n} \tau_{ij}, \qquad i=1,\dots,g,$$

$$\sum_{i=1}^{g+1} \tau_{ij} = 1, \qquad j=1,\dots,n \text{ (placement)},$$

$$\sum_{j=1}^{n} \tau_{ij} \geqslant H, \qquad i=1,\dots,g \text{ (size)},$$

$$\sum_{j=1}^{n} \tau_{ij} \leqslant T, \qquad i=g+1 \text{ (outlier)},$$

$$\tau_{ij} \in \{0,1\}, \qquad i=1,\dots,g+1,$$
$$j=1,\dots,n.$$

References

Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian Clustering. Biometrics 49, 803–821.

Battiti, R., Tecchiolli, G., 1994. The reactive tabu search. ORSA J. Comput. 6, 126–140.

Bernholt, T., Fischer, P.. The complexity of computing the MCD-estimator. Theoret. Comput. Sci. (to appear).

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., Zanasi, A., 1994. Discovering Data Mining. Prentice-Hall, Upper Saddle Rivier, NJ.

Coleman, D.A., Woodruff, D.L., 2000. Cluster analyse for large data sets: efficient algorithms for maximizing the mixture likelihood. J. Comput. Graphic. Statist. 9, 672–688.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugen. 7, 179–188.

Friedman, H.P., Rubin, J., 1967. On some invariant criteria for grouping data. J. Amer. Statist. Assoc. 62, 1159–1178.

Glover, F., 1989. Tabu search (Part I). ORSA J. Comput. 1, 190–206.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publisher, Boston, Dordrecht, London.

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A., 1986. Robust Statistic—The Approach on Influence Functions. Wiley, New York, Chichester, Brisbane, Toronto, Singapore.

Hawkins, D.M., 1994. The feasible solution algorithm for the minimum covariance determinant estimator in multivariate data. Comput. Statist. Data Anal. 17, 197–210.

Hennig, C., Christlieb, N., 2002. Validating visual clusters in larger datasets: fixed point clusters of spectral features. Comput. Statist. Data Anal. 40, 723–739.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C., 1989. Optimization by simulated annealing: an experimental evaluation (Part I). Graph partitioning. Oper. Res. 37, 865–892.

McLachlan, G.J., 1982. The classification and mixture maximum likelihood approaches to cluster analysis. In: Krishnaiah, P.R., Kanal, L. (Eds.), Handbook of Statistics, Vol. 2. North-Holland, Amsterdam, (pp. 199–208).

McLachlan, G.J., Basford, K.E., 1988. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York.

McLachlan, G.J., Peel, D., 2000. Finite Mixture Models. Wiley, New York.

Murtagh, F., Raftery, A.E., 1984. Fitting straight lines to point patterns. Pattern Recognition 17, 479–483.

Pacheco, J., Valencia, O., 2003. Design of hybrids for the minimum sum-of-squares clustering problem. Comput. Statist. Data Anal. 43, 235–248.

Reiners, T., 1998. Maximum likelihood clustering of data sets using a multilevel, parallel heuristic. Master Thesis. Technical Report, Department of Economics, Business Computer Science and Information Management, Technische Universität Braunschweig, D-38106 Germany.

Rousseeuw, P.J., 1985. Multivariate estimation with high breakdown point In: Grossmann, W., Pflug, G., Vincze, I., Werz, W. (Eds.), Mathematical Statistics and Applications, Vol. B. Reidel, Dordrecht.

Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. Wiley, New York.

Scott, A.J., Symons, M.J., 1971. Clustering based on likelihood ratio criteria. Biometrics 27, 387–397.

Späth, H., 1985. Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples. Ellis Horwood Limited, Chichester.

Voß, S., Woodruff, D.L. (Eds.), 2002. Optimization Software Class Libraries. Kluwer, Boston.

Ward, J.H., 1963. Hierarchial groupings to optimize an objective function. J. Amer. Statist. Assoc. 58, 236–244.

Woodruff, D.L., 1997. A class library for heuristic search. INFORMS Comput. Sci. Tech. Sec. Newslett. 8 (2), 1–5.

Woodruff, D.L., Rocke, D.M., 1994. Computable robust estimation of multivariate location and shape in high dimension using compound estimators. J. Amer. Statist. Assoc. 89, 888–896.

Intl. Trans. in Op. Res. 11 (2004) 225–238

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Teaching meta-heuristics within virtual learning environments

T. Reiners[a] and S. Voß[b†]

[a]*University of Technology Braunschweig, Institut für Wirtschafswissenschaften, Spielmannstraße 8, D-38106 Braunschweig, Germany*
[b]*University of Hamburg, Institut für Wirstchaftsinformatik, Von Melle-Park 5, D-20146 Hamburg, Germany*
E-mails: t.reiners@tu-bs.de [Reiners]; stefan.voss@uni-hamburg.de [voß]
[†]*Corresponding author*

Received 26 September 2002; received in revised form 14 March 2003; accepted 23 April 2003

Abstract

Modern education of operations research and management science (OR/MS) can greatly benefit from interactive learning methods in order to build and develop modeling and problem-solving skills. In this paper we consider the teaching of meta-heuristics as an important part of OR/MS with significant recent interest. We discuss possibilities of supporting the teaching of meta-heuristics such as simulated annealing or tabu search through interactive learning. The paper also presents a survey of some relevant issues within VORMS (Virtual Operations Research/Management Science), a project currently undertaken at six universities within Germany, and provides a presentation of the advances regarding the teaching of meta-heuristics within this project. Further ideas refer to incorporating HOTFRAME, a heuristic optimization framework, into the virtual learning environment.

Keywords: Virtual learning; meta-heuristics; education.

Introduction

Modern information technology allows manifold variations of education, besides the traditional presence university. Especially in the context of growing numbers of students, fewer funds for education, and the desire to have a lifelong learning support, virtual universities may be an opportunity to support education and, therefore, are currently opened in many countries using different strategies for their educational programs (see, e.g., Hazemi, Hailes and Wilbur, 1998). Most of these programs are restricted to on-line access for all course materials (e.g. Massachusetts Institute of Technology, 2002), but can also lead to a full digital or virtual university where the learner is enrolled as a regular student (e.g. University of Phoenix, 2002; see Björck, 2002, for a list of virtual universities).

Regarding the field of operations research, it is important to keep track of the research developments in educational methods and virtual learning environments because of the high degree of required practical training. In this paper we demonstrate a concept of how to involve the training of practical skills in virtual learning environments, using meta-heuristics as an example. Meta-heuristics are modern and important tools for optimization, but so far, they lack an appropriate integration in lectures, as well as practical application. Most traditional presence courses either do not comprehensively teach meta-heuristics at all, or teach meta-heuristics in classes with limited discussion opportunities just using textbooks — where meta-heuristics currently are not included or only briefly mentioned. Furthermore, the learning material is in most cases only transferred into static learning content for virtual learning environments. As a result of this way of presenting the material, the main advantages of meta-heuristics compared to other optimization methods are neither sufficiently motivated in most cases nor sufficiently illustrated using good examples that are based on real-world scenarios. Therefore, meta-heuristics are still having a shadowy existence apart from some famous algorithms like the genetic algorithms.

Particularly for learners, it is a necessity to acquire a multitude of knowledge as well as practical skills in areas like modeling, algorithms, software engineering, or project management, and therefore, we have to ask for certain characteristics of the virtual learning environment. In addition to simple presentations of course material without any further media pedagogical revision, different virtual learning environments can be found on the Internet using a more sophisticated approach than hyperlinked documents. Examples for off-line software can be found at OR-World (2002); free web-based learning environments are, e.g. tutOR by Sniedovich and Byrne (2002) and the tutORial from the IFORS (2002), based on it. Sophisticated commercial approaches regarding the general support of teaching and managing course material are, e.g., WebCT (2002) and Blackboard (2002). In many countries, e.g. the United States of America or Australia, it is also common to use proprietary software like Microsoft Excel within the classroom instead of web-based learning environments. The tutor can use the given functionality to easily implement complex visualizations of problem formulations and their solutions, including a step-by-step development, whereas the learner already knows the software and, therefore, can focus on the learning material and given exercises; see, e.g. Bell (2000) or Evans (2000).

Even though the learner obtains an introduction or exemplification to operations research by having interactive components to visualize algorithms and their integration in the whole field, certain didactical concepts are missing in almost the same manner as specific subjects. For example, meta-heuristics are either not described at all, only given by static hypertext documents, or just as a collection of interesting and interactive applets without the textual background explaining the theory. Different forms of presenting general optimization results can be found, e.g. in Jones (1995).

As mentioned in the introduction above, the main disqualification of existing virtual learning environments for meta-heuristics (and also partially operations research in general) is the missing integration of reality-based problems and applications (which indeed is an old discussion in general as pointed out, e.g. by Wolsey (1979) or Reisman (1997) by saying: 'This profession currently has more algorithms than applications.'). Therefore, we will demonstrate below how the knowledge from a theoretical course as outlined in the section 'Teaching Basics in Meta-Heuristics' can be applied to real problems whereas the learner is extensively supported by the virtual learning environment. The architecture of a modern virtual learning environment is briefly described in the Section 'Concept of a Virtual Learning Environment'. Using this approach the learner can perform experiments using meta-heuristics without having the typical cleanroom conditions that are commonly used in most examples. In this paper we focus on the description of our concept using steepest descent with restart mechanisms and simulated annealing as examples without loss of generality. The final section provides some conclusions.

The research in this paper is part of the VORMS project (Virtual Operations Research/ Management Science; see VORMS (2002)) that started in 2001 at six universities in Germany. The main goal within this project is the design of new didactical methods to present learning material in the field of OR/MS. Here, the focus is the increase of motivation of the learners to actually participate in the courses but also to see the importance of operations research methods in combination with other fields.

Concept of a virtual learning environment

Before giving an explicit example of how to teach meta-heuristics, we are going to describe our concept for a virtual learning environment that supports the learner having a motivating learning experience by allowing (adaptive) configurations of the environment and the content itself. In addition to certain features that should be given in any learning environment (e.g. presenting the learning content in a structured form, using an intuitive interface for the presentation and navigation, or hyperlinked course units), we have to integrate additional components to embrace the difficulty of teaching and especially motivating learners to take courses in operations research without a tutor being available all the time. Currently, it is a major problem to obtain the interest as well as develop and support the motivation of the learner — whereas we do not distinguish between different kinds of learners, e.g. the distant learner, the enrolled learner, or the lifelong learner (Wedemeyer, 1981).

The technological implementation of a modern web-based virtual learning environment has to be based on up-to-date technologies that follow international standards and, therefore, guarantee a worldwide acceptance as well as (re)usability independent from certain software architectures. Furthermore, the system should not be based on proprietary components, but be compatible with different operating systems and should be free of copyrights. We focus on open-source software packages by using specifications that are approved by committees like IEEE or W3C. Using modern software and XML-based specifications like LOM (Learning Object Metadata), LMML (Learning Material Markup-Language), or DocBook to encode the learning material, allows us to build a flexible environment that adapts to the learners' preferences instead of pushing the learner into a given structure (Reiners, Reiß and Voß, 2002b).

Figure 1 shows an overview of the system architecture as well as the structure of the learning material. On the left-hand side, different learning objects (media files, applets, text, and formulas) are shown, which are part of the learning material and combined to the required learning unit or document. The right hand side of Fig. 1 shows a simplified system architecture with all relevant components, i.e., the JSP-processes to create the graphical user interface, Apache Cocoon for the transformation of XML-coded learning material to the required output format, the databases to store the learning material as well as other information. A more detailed technical description can be found in Reiners, Reiß and Voß (2002a, 2002b).

The learning material is given in so-called learning objects. The following overview represents the four realized types:

- media element, small and not further divided learning objects; e.g., text, animation, simulation, video, or audio sequence;

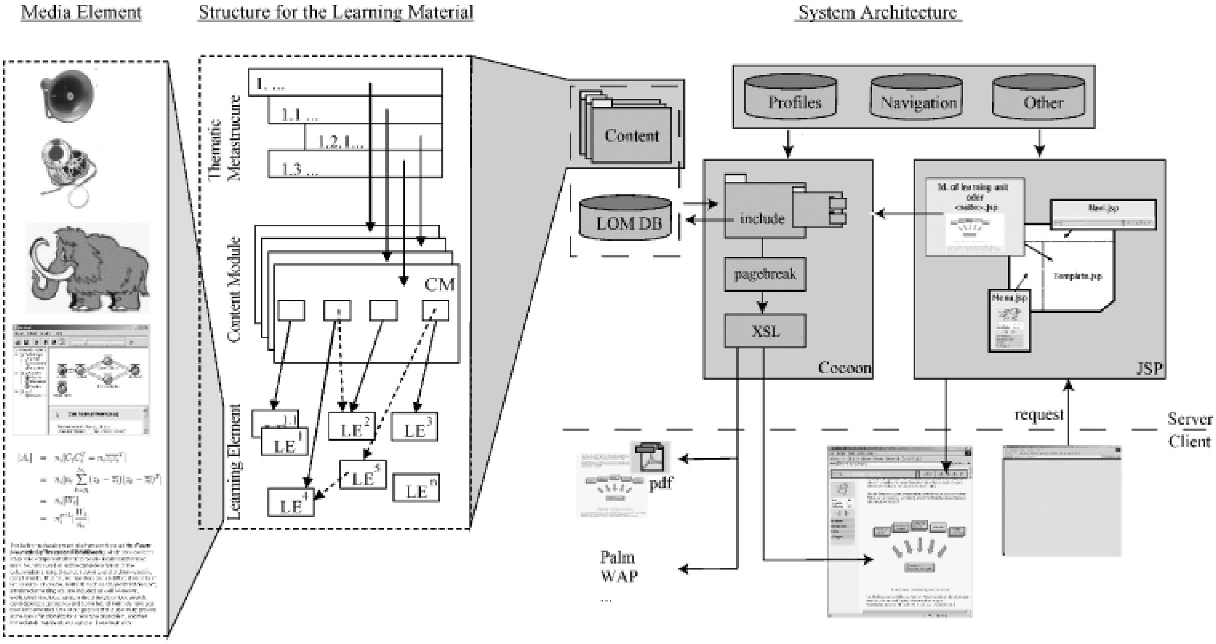


Fig. 1. Structure for a web-based virtual learning environment.

Notes: XSL: Extensible Stylesheet Language, JSP: JavaServer Page, PDF: Portable Document Format, WAP: Wireless Application Protocol, DB: Database

- learning element (LE) being a composition of one or more media elements or learning elements;
- content module (CM) consisting of one or more learning objects, and is understood as a node in the hypermedia network;
- thematic meta-structure defining guidelines how to use content modules to build thematic structures relevant for a specific study goal; such a structure can be put together in individual ways, thus adapting to different combinations and profiles.

The meta-data of the learning material is stored within a relational database. Learning objects are, e.g. media objects, text passages, or applications, whereas their sizes are determined by their semantic density. Learning objects are combined into larger learning units or complete documents. The learning objects are linked within the database allowing the learner a compilation of learning material on the fly according to the preferred learning experience like learning level, complexity of the units, degree of interactivity, or just the visual presentation.

Furthermore, authors can implement learning paths for certain groups of learners by combining the learning material and setting links between the learning objects. As mentioned above, interactivity, navigation, and communication are the main aspects for a framework. We use a browser-based approach to have an already available support for certain standards (e.g. Javaapplets, media elements, or plug-ins) so that we can concentrate on other factors, which are the development of new and innovative applets and software, as well as the integration of a communication interface for learner-learner and learner-tutor communication. For a more detailed description of a concept for a virtual learning environment and its architecture we refer to Reiners, Reiß and Voß, (2002a) where other technologies like hyperbolic trees and SmartBars are introduced in the context of learning environments.

Teaching basics in meta-heuristics

Our course on meta-heuristics within the virtual learning environment has to introduce the learner to the basics, assuming no specific prerequisites or previous knowledge about the subject. Even though we have to describe the basic concept of local search (as an essential part of most meta-heuristics), the main focus should be the transfer of knowledge about the meta-heuristics such that the learner understands the concept and also gains the capability to use it within real-world scenarios.

Meta-heuristics

The formal definition of meta-heuristics is based on a variety of definitions from different authors. Basically, a meta-heuristic is a top-level strategy that guides an underlying heuristic solving a given problem. That is, a meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate iteratively a complete (or incomplete) single solution or a collection of solutions. The subordinate heuristics are, e.g. high- (or low-) level procedures, simple local search, or just a construction method. Meta-heuristics may use learning strategies to structure information in order to find optimal or near-optimal solutions efficiently; see, e.g. Glover and Laguna (1997), Osman and Kelly (1996), or Voß, Martello, Osman and Roucairol (1999). This definition has to be explained stepwise by describing the components as well as their interactions.

- General description of a solution space and the idea of meta-heuristics

  Assuming a given problem, the goal is to find an optimal or at least a high-quality solution. Each problem is associated with a solution space containing all feasible solutions according to the restrictions of the problem. One way of finding an optimal solution can merely be the search through all solutions — also called complete enumeration — and selecting the best one. Unfortunately, the size of the solution space is too large to accomplish all comparisons within a realistic time span and, therefore, heuristic methods have to be applied that limit the search on interesting areas of the solution space (whereas the learner should have a reference to a further learning unit explaining complexity theory).

- Description of the main components of meta-heuristics
  - Problem formulation: For example, a traveling salesman problem (TSP) defining a minimization problem, where a number of cities has to be visited exactly once at minimum cost.
  - Objective function (value): An objective function for validating and evaluating solutions has to be given and an order has to be defined. For instance, it must be possible to determine which of two solutions is the better one.

  - Solution representation: Different data structures can be used to store a solution for a problem. Examples for a representation are binary or permutation vectors. For instance, the TSP can be coded as a vector of n elements, with n being the number of cities numbered from 0 to $n-1$.
  - Move: A move is the transition from one solution to another solution using certain rules; see the following section for examples.
  - Neighbor and neighborhood: The neighborhood of a solution is defined as the set of solutions (neighbors) that can be reached from that solution by one move.

- Relation between these components
- Interactive examples to experience and memorize various meta-heuristics

Well-known examples of meta-heuristics are, e.g. simulated annealing and tabu search. In this paper we focus on steepest descent and simulated annealing to illustrate the concept; see Ribeiro and Hansen (2002) or Voß (2001) for a more detailed description and references regarding additional meta-heuristics.

Both meta-heuristics are local search-based algorithms where, beginning with a given starting solution, the neighborhood is searched for the next move to be performed. The selection criterion for steepest descent is determined by the objective function value. That is, assuming a minimization problem a neighbor with a lowest objective function value is selected and becomes the starting solution for the next iteration. This process is repeated until the neighborhood does not contain solutions with — in case of a minimization problem — lower objective function values. This solution is the result of the algorithm and represents at least a local optimum.

A drawback of steepest descent is the fact that it starts from one solution and only has downhill moves. Therefore, only a small fraction of the whole search space is visited, leaving a large number of solutions untouched and uninvestigated. Two simple methods to avoid the lack of visited solutions are the restart of the steepest descent from various random solutions in a loop or the performance of several random steps to a new starting solution. The solutions of all descent runs will be compared and one with the best objective function value will be returned as the final result.

Simulated annealing improves the neighborhood selection by allowing a (occasional) selection of neighbors with inferior objective function values resolving the problem of getting stuck in local optima. Based on a temperature that decreases over time and a given cooling schedule the probability of selecting an inferior move is calculated by

$$\Delta < 0 \quad \vee \quad e^{\frac{-\Delta}{t}} > R_p$$

with $\Delta = f(\tau') - f(\tau)$ being the delta between the objective function values f of the current solution τ and the randomly selected solution τ' from the neighborhood, R_p being a pseudo-random number drawn from a uniform distribution on (0, 1), and t being the temperature set by the cooling schedule. The search is terminated after a certain number of iterations returning the best found solution.

Interactive visualization of meta-heuristics

Based on the general description in the previous section, we will discuss how an interactive realization of a virtual course on meta-heuristics can be done. Figure 2 shows on the left-hand side a possible

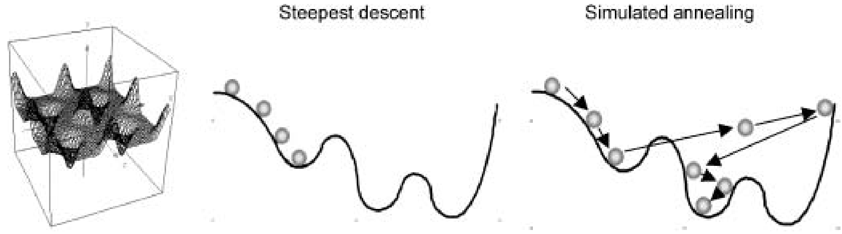


Fig. 2. Solution space and local.

visualization of a solution space in form of a mountain landscape. For a minimization problem the peaks are 'bad' solutions and the valleys are the 'good' ones. The search 'just' has to find the lowest point in the solution space and output its location and the solution belonging to it, respectively.

A common analogy of the search process uses a ball being placed on a peak. Using steepest descent the ball will roll downwards into a near valley, being at least a local optimum, and the method stops returning the current location or solution. Simulated annealing and other meta-heuristics on the other side provide special mechanisms to propel the ball over the next bump as shown on the right-hand side of Fig. 2. Therefore, the algorithms might find better solutions due to fewer restrictions; see also Woodruff (1994) for another nice analogy using a drawn line on a chalk board.

The following example shows a permutation vector representing a solution for the TSP as described in the previous section; note that a binary vector would be displayed in the same way. The solution shown in Fig. 3 on the left hand side describes the round-trip along the cities 0 to 4 and back to 0 with a trip length of 73 (objective function value).

Examples for moves are exchanges of two elements within the solution vector as shown on the right hand side of Fig. 3, insertion moves, where an element is moved from one position to another one, or add-and-drop moves, where elements are added or removed from the solution. The representation of the move as well as the solution should be interactively designed allowing the learner to see how solutions are changed. This can be done by having explanations for elements popping up while the mouse passes over one of the elements, letting the learner perform the move by dragging elements, or having an animation to demonstrate, e.g. the exchange of two elements.

The learning material for solution and move representations is followed by a course unit about neighborhoods. The neighborhood depends on several factors like the type of move or solution representation, but a general visualization can be given as shown in the following example. Within an interactive animation, moves from a given starting solution, which might be a randomly generated solution or the output of another (simpler) heuristic, are displayed forming the

Fig. 3. Solution and move representation.

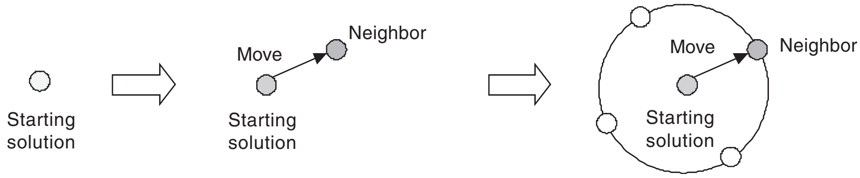


Fig. 4. Visualization of a neighborhood for a given solution.

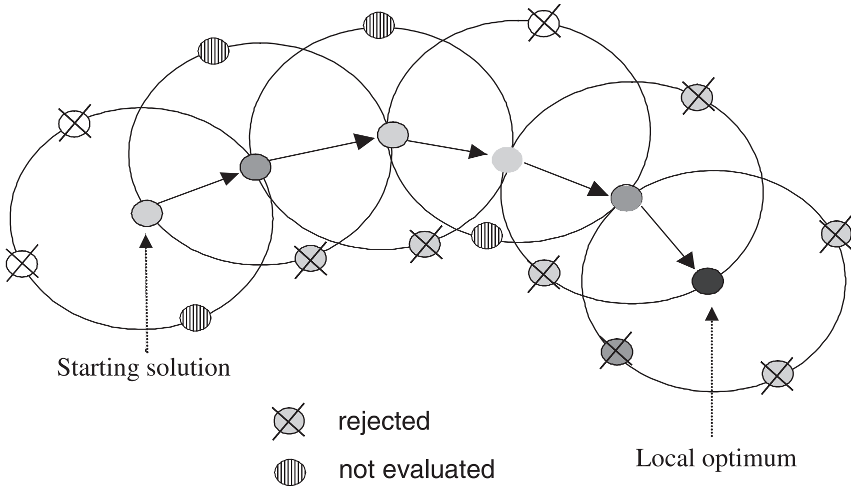


Fig. 5. Moving through the solution space.

complete neighborhood as shown in Fig. 4. The shade of gray corresponds with the objective function value, i.e. a darker gray symbolizes a better quality according to the optimality criterion.

These introduced components are used (in variations) within meta-heuristics and have to be internalized by the learners before continuing. Variations are, e.g., the number of calculated neighbors before initiating a move (simulated annealing does not require to know all neighbors whereas steepest descent has to select the next move between all neighbors) or allowed moves (tabu search might restrict possible moves).

After introducing the main components of meta-heuristics, specific algorithms like simulated annealing or tabu search can be explained. For simplicity, we set the focus on steepest descent with random restarts and simulated annealing. Other meta-heuristics could be presented in the same way using further visualization in terms of parameters or extra information like tabu lists. The traversal or trajectory through the solution space should be presented as an interactive animation where the learner can either observe the process or actively select the ‘correct’ neighbor according to the meta-heuristic strategy as part of a learning process. A realization of such an animation can be realized similar to Fig. 5, which shows the final result after a local optimum is reached using simulated annealing. Here, neighbors are indicated if they are either not considered as a neighbor at all or being rejected due to the selection criteria based on the temperature. Note that the rate of rejecting a solution increases over time due to the fact that the temperature is decreased.

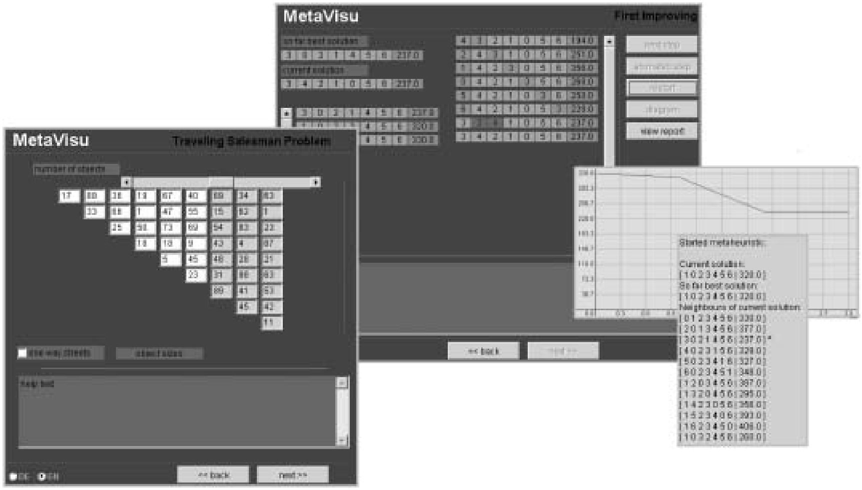


Fig. 6. Applet to demonstrate the internal processes of meta-heuristics.

We developed an applet allowing us to visualize meta-heuristics as well as their components; see Fig. 6. The example shows steepest descent for a traveling salesman problem where the solution is represented as a permutation vector s with s_i being the number of the city on the tour. On the left-hand side, the screenshot shows the definition of the problem instance allowing the adaptation of the problem size as well as the values of the edges. The middle part shows the visualization of the solution process; the current solution is shown including a history of previously visited solutions. The list of solutions in the middle column represents the neighborhood of the current solution. Cities that were exchanged according to the definition of a move to a neighbor are emphasized as colored elements. The learner has several options to use the applet. The learner can watch an animated walk through the solution space, walk step-by-step towards a local optimum reading the course material, or interactively perform the search by selecting the next step. Furthermore, the details of the animation can be chosen, either the solutions are directly copied to their new position, are moved slowly, or even the move itself is presented as shown in Fig. 3. Further windows can be opened to visualize the development of the objective function value or a log file of all previously executed steps.

A similar representation can be used for other meta-heuristics. The neighbors with deteriorating objective function values can be selected depending on a temperature-driven probability function within simulated annealing and, therefore, have to be specially presented. This can be done using colors for the solutions. Tabu search memorizes visited solutions that cannot be revisited as long as they are in a so-called tabu list. In this case, solutions could be crossed out within the presentation of the neighborhood.

Visualization of the learning material

A disadvantage of most learning environments is the adaptation to the learning level. For instance, the previously-described learning unit would be useful to beginners not having a detailed background in operations research. In case of advanced learners who want to learn about new algorithms or need to look up details, the illustrative presentation is both too long as well as not detailed enough. For this group, a learning unit needs a clear focus and a profound explanation using, e.g., mathematical formulas. A learning unit for advanced learners could be as described in the following paragraph.

Steepest descent starts from a given initial solution τ, which is, e.g., randomly generated and searches for the best (local) solution τ' in the neighborhood of τ, denoted as $N(\tau)$. In the case of a neighborhood containing a solution better than the best solution found so far, with respect to the objective function value, a descent move in this direction will be performed, i.e. the new solution will become the center of a neighborhood $N(\tau')$. Otherwise, the descent is terminated and a local optimum $\tau^* = \tau'$ is found. This local optimum cannot be understood as the optimal solution because the heuristic is missing a feature to force itself out of the found valley and eventually fall into a deeper one. Another possibility for a variation of a descent algorithm is, e.g., first improving descent where the first neighbor with a better objective function value is chosen. The following pseudo-code describes steepest descent where ic_{max} is a parameter indicating a maximum number of iterations.

$ic \leftarrow 0$ (*iteration counter*)
$\tau \leftarrow$ "Random Solution"
do
    $ic \leftarrow ic + 1$
    for all $\tau' \in N(\tau)$ **do**
        if $f(\tau') < f(\tau)$ **then** $\tau^* \leftarrow \tau'$ **end**
    end
    $\tau \leftarrow \tau^*$
while not ("stop criterion" $\vee$ $ic > ic_{max}$)

Interactive examples or visual representations can be referenced in the learning material, too, by using hyperlinks instead of direct inclusion. Therefore, the advanced learner gets a short but sufficient explanation, without unnecessary and maybe distracting visual extras. We should note, that even the mathematical representation allows a variety of interactivity like the description of variables if the mouse is moved over them, links from equations to their derivation, or animated execution of the pseudo-code in a debug-mode where the content of the variables is shown during execution time. On the other side, the beginning learner can also get the mathematical formulation using integrated links but should not be directly confronted with too many details.

Teaching meta-heuristics: From learning to applying

The learner has to work actively on non-trivial problems to achieve certain skills in operations research. Therefore, we have to extend the virtual learning environment by a further component with the main intention of supporting the application of the learned knowledge. The learner uses the existing learning material to study the fundamental principles of OR like algorithms, models, and solution procedures. This learning phase is mainly supported by simple interactive examples that visualize the underlying concepts and demonstrate the behavior of methods. Regarding an improved internalization process, the learner should apply the gained knowledge to larger and in particular realistic problems that might even be part of ongoing research projects or, e.g. instances of problems from an Internet library that might even not be solved to optimality so far; see, e.g., OR-library (2002).

Supported by the learning environment, the learner has to design a solution approach to solve given problems as good as possible using existing and configurable software packages. This software can either be a commercial tool having graphical user interfaces to enter the problem together with parameters for the solution algorithms, or software libraries (frameworks) that contain more or less reusable codes but also require some programming knowledge. To provide a strong integration of the software into the virtual learning environment without having large media or technology breaks, we use the framework HOTFRAME (Heuristic OpTimization FRAMEwork) by Fink and Voß (2002). The framework supports both adaptable components that incorporate local search based meta-heuristics and an architectural description of the collaboration among these components and problem-specific complements. Methods such as steepest/first descent, simulated annealing, and tabu search are included next to evolutionary methods, variable depth neighborhood search, candidate list approaches, and some hybrid methods.

Even though the learner has to provide some code fragments — in this case in C++— for certain problems, it is almost impossible to ask for a full understanding of the framework and its implementation. Therefore, we provide an interface to configure the main components of the meta-heuristics like the solution representation, the move, and the neighborhood according to the problem type and receive an automatically-generated source code with todo-parts that have to be replaced with special source code from the learner. The todo-parts can range from a simple method to read the problem formulation up to a complete new solution representation.

Figure 7 shows the complete sequence that the learner has to perform being continuously supported by step-by-step instructions within the learning environment. Using a software generator for the framework the learner needs to specify the problem as well as algorithmic details by selecting predefined meta-heuristic components to be used in the later program. The generator itself does not output a final program but a customized source code, which has to be completed by the learner. Finally, the virtual learning environment provides an environment to execute experiments where the compiled program is executed on several problem instances with a variation of specified parameter settings for the meta-heuristics. The results are collected and can be evaluated by the learner using several statistical methods and visualization forms. Note that we do not implement the all-in-one device suitable for every purpose for comparison of meta-heuristics being applicable within all scenarios. Even though our approach can be seen as a suggestion for a user interface to such a tool the underlying logic and mechanisms for an adequate scientific evaluation and especially comparison are designed to fulfill the requirements of learners. In this context, together with some students we have developed an interface to manage problems, algorithms, and solutions of experiments as well as an XML-based representation and storage of the corresponding data.

To fully grasp the rules and mechanisms to apply a framework one may have to manage a steep learning curve. Therefore, a framework might enable an incremental application process (adoption path); see Fink, Voß and Woodruff (1999). That is, the user may start with a simple scenario, which can be successively extended, if needed, after having learned about more complex application mechanisms.

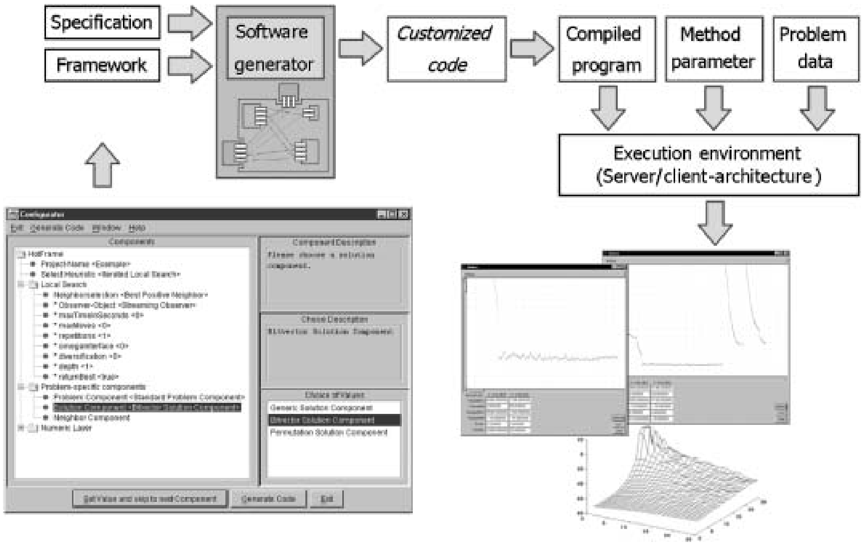


Fig. 7. Sequence for applying a meta-heuristic to a problem.

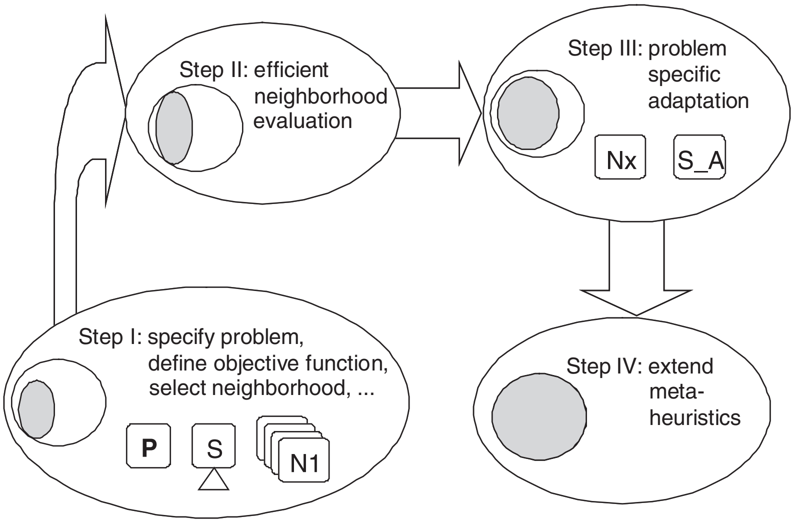


Fig. 8. Adoption path.

In Fig. 8, we visualize a typical adoption path for the case that some of the problem-specific standard components are appropriate for the considered application (see Fink and Voß, 2002). In this process, we quickly — after completing the first step — arrive at being able to apply several kinds of meta-heuristics to the considered problem, while efficiently obtaining high-quality results may require following the path to a higher level. The steps of the adoption path may be described as shown in Fig. 8.

I. *Objective function*: After selecting an appropriate solution component, one has to derive a new class and to code the computation of the objective function. Of course, one also needs some problem component, which provides problem instance data. All other problem-specific components may be re-used without change.

II. *Efficient neighborhood evaluation*: In most cases, the system that results from Step I bears a significant potential with regard to improving run-time efficiency. In particular, one should implement an adaptive computation of the move evaluation (which replaces the default evaluation by computing objective function values for neighbor solutions from scratch). In this context, one may also implement some efficient move evaluation that differs from the default one (implied change of the objective function value).

III. *Problem-specific adaptation*: Obtaining high-quality solutions may require the exploitation of problem-specific knowledge. This may refer to the definition (and implementation) of a new neighborhood structure or an adapted tabu criterion by specific solution information or attribute components.

IV. *Extension of Meta-heuristics*: While the preceding steps only involve problem-specific adaptations, one may eventually want to extend some meta-heuristic or implement a new heuristic from scratch.

Conclusions

There is probably no discussion about the necessity of transferring the content of traditional lectures to virtual learning environments to effectively enhance traditional learning and training methods in a society, which is becoming increasingly knowledge-based. However, because network-based learning over the Internet provides new dimensions of organizing individual and group-based learning processes, it will probably take years until appropriate ways of using the new possibilities will be established. Many current learning environments on the Internet are not even using the simplest options of integrating new didactical methods for supporting the self-guided learning process but build static web-sites with some hyperlinks. Furthermore, even more interactive learning environments do not support the idea of using the gained knowledge about a certain research field to handle real-world questions and problems.

In this paper we outlined a learning unit for meta-heuristics. As in traditional virtual learning environments we introduce the learner to the basics by demonstrating small interactive learning units, including descriptive examples. Afterwards the learner is not left alone but being supported and guided regarding the first steps of transferring the gained knowledge to larger and more realistic problems. Especially allowing the learners to use software capable of solving large problem instances implies that the learner is taken seriously — which helps the learner to gain self-confidence — and that the learned methods can really be used in real-world problems — which also helps to eliminate the prejudice that methods of operations research might be only useful in theory.

Acknowledgements

This work was supported by grants from the BMBF (Bundesministerium für Bildung und Forschung) in the program 'Neue Medien in der Bildung' (08NM094D).

References

Bell, P.C., 2000. Teaching Business Statistics with Microsoft Excel. *INFORMS Transactions on Education*, 1(1), 18–26. *http://ite.informs.org/Vol1No1/bell/bell.html*.

Björck, U., July 2002. *List of Virtual Universities*. http://www.ped.gu.se/ulric/vus.html

Blackboard, Inc., July 2002. Blackboard. http://www.blackboard.com

Evans, J.R., 2000. Spreadsheets as a Tool for Teaching Simulation. *INFORMS Transactions on Education*, 1(1), 27–37. http://ite.informs.org/VollNo1/evans/evans.html.

Fink, A., Voß, S., 2002. *HotFrame: A Heuristic Optimization Framework*. In: Voß, S., Woodruff, D.L. (Eds.). *Optimization Software Class Libraries*. Kluwer, Boston, pp. 81–154.

Fink, A., Voß, S., Woodruff, D.L., 1999. *An Adoption Path for Intelligent Heuristic Search Componentware*. In: Rolland, E., Umanath, N.S. (Eds.). *Proceedings of the 4th INFORMS Conference on Information Systems and Technology*. INFORMS, Linthicum, pp. 153–168.

Glover, F., Laguna, M., 1997. *Tabu Search*. Kluwer, Boston.

Hazemi, R., Hailes, S., Wilbur, S. (Eds.), 1998. *The Digital University: Reinventing the Academy*. Springer, London.

IFORS, July 2002. tutORial. http://www.ifors.ms.unimelb.edu.au/tutorial

Jones, C.V., 1995. *Visualization and Optimization*. Kluwer, Boston.

Massachusetts Institute of Technology, July 2002. *Educational Uses of the WWW at MIT*. http://web.mit.edu/is/courseweb/courses.html

OR-Library, July 2002. http://www.ms.ic.ac.uk/info.html

OR-World, July 2002. *OR-World*. http://www.or-world.com

Osman, I.H., Kelly, J.P., 1996. *Meta-Heuristics: An Overview*. In: Osman, I.H., Kelly, J.P. (Eds.). *Meta-Heuristics: Theory & Applications*. Kluwer, Boston, pp. 1–21.

Reiners, T., Reiß, D., Voß, S., 2002a. *Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts*. Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002). ICSC-Naiso Academic Press, Millet Alberta, #100029-03-TR-026, 1–7.

Reiners, T., Reiß, D., Voß, S., 2002b. *XML-basierte Kodierung von Lernobjekten*. Technical report, in preparation. University of Technology, Braunschweig.

Reisman, A., 1997. *Flowshop Scheduling/Sequencing Research: A Statistical Review of the Literature, 1952–1994*. *IEEE Transactions on Engineering Management*, 44, 316–329.

Ribeiro, C.C., Hansen, P. (Eds.), 2002. *Essays and Surveys in Metaheuristics*. Kluwer, Boston.

Sniedovich, M., Byrne, A., 2002. *tutOR*. http://www.tutor.ms.unimelb.edu.au/frame.html

University of Phoenix, July 2002. University of Phoenix Online. http://online.phoenix.edu

VORMS, 2002. *Virtual Operations Research/Management Science*. http://www.vorms.org

Voß, S., 2001. *Meta-Heuristics: The State of the Art*. In: Nareyek, A. (Ed.). *Local Search for Planning and Scheduling*. Lecture Notes in Artificial Intelligence 2148, Springer, Berlin, pp. 1–23.

Voß, S., Martello, S., Osman, I.H., Roucairol, C. (Eds.), 1999. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer, Boston.

WebCT, Inc., July 2002. *WebCT.com*. http://www.webct.com

Wedemeyer, C.A., 1981. *Learning at the Back Door: Reflections on Non-Traditional Learning in the Lifespan*. University of Wisconsin Press, Madison.

Wolsey, R.E.D., 1979. *Pragmatism Triumphant or Past Sophistication and Future Elegance*. In: Haley, K.B. (Ed.). *Operational Research '78*. North Holland, Amsterdam, pp. 80–86.

Woodruff, D.L., 1994. *Simulated Annealing and Tabu Search: Lessons from a Line Search*. *Computers & Operations Research*, 21, 823–839.

Torsten Reiners, Dirk Reiß, Imke Sassen und Stefan Voß

Die virtuelle Lernumgebung SmartFrame Kodierung und didaktische Aufbereitung von Lernmaterialien durch Lernobjekte

The Virtual Learning Environment SmartFrame

Mensch-Computer-Interaktion_Grafische und multimediale Benutzungsschnittstellen_Virtuelle Lernumgebung

Zusammenfassung. Die Darstellung von Lernmaterialien erfolgt in den meisten Fällen durch statische, mit interaktiven Applets aufgewertete HTML-Seiten. Ein ganzheitliches Lernerlebnis mit individueller Adaptivität wird mit dieser einfachen Kodierung von Lernmaterialien allerdings nur in einem stark eingeschränkten Sinne ermöglicht. Erst durch eine verstärkte Einbindung von interaktiven Lernelementen in Kombination mit dynamisch generierten Seiten und einer individuellen Adaption der Lernumgebung an den jeweilig Lernenden kann echter Mehrwert im Sinne weitergehender didaktischer Unterstützung erzielt werden.
Zunächst präsentieren wir die mediendidaktische Konzeption und Umsetzung der virtuellen Lernumgebung SmartFrame (www.smartframe.de). Aus konzeptioneller Sichtweise soll aufgezeigt werden, wie eine ganzheitliche Adaption der Lernumgebung an individuelle Lernprozesse erreicht werden kann, die über die bereits übliche Interaktivität einzelner Lernmodule und benutzerspezifischer Eingangskonfigurationen hinausgeht.
Im Anschluss wird schematisch die technische Realisierung des Transformationsprozesses der XML-kodierten Lernobjekte beschrieben, der sich durch eine dynamische – d.h. beim Abruf der Inhalte – lernerspezifische Adaptivität der Lernmaterialien in Bezug auf deren Zusammensetzung und Präsentation auszeichnet.

Summary. In most cases, the visualization of learning material is realized by static web-sites including interactive applets. This simple form of coding does not allow a fully integrated learning experience with an individual adaptive approach to the learners' requirements. To achieve a surplus in terms of a didactical support, the inclusion of highly interactive learning elements combined with on-the-fly generated content pages for the learner has to be given.
In this paper we describe the media-didactical concept of the implemented virtual learning environment SmartFrame (www.smartframe.de). Within SmartFrame, the XML-coded learning objects are dynamically combined to larger learning units that are visualized according to the current learner specification based on the learning process. Thereby, the adaptation of the learning material to the learner itself is on a more advanced level as provided by most common learning systems.
The implementation of the here shown concepts is described by presenting the main aspects of the transformation process from XML-coded learning objects to the learner specific presentation of the learning material.

1. Einleitung

Die Einbindung virtueller Lernumgebungen in das universitäre Curriculum ist ein Ziel, das angesichts steigender Studierendenzahlen und verringerter finanzieller Ressourcen der Ausbildungsinstitutionen von einem großen Teil nationaler sowie internationaler Universitäten angestrebt wird; siehe z.B. (Kretschmer 2002). Die zur Zielerreichung eingesetzten, meist kommerziellen Lernplattformen wie Blackboard, WebCT oder Clix – welche an vielen Universitäten im großen Stil eingeführt werden - besitzen bereits Tools, die Lehrenden auf relativ einfache Weise die Einstellung und Distribution von digitalen Lernmaterialien ermöglichen (Baumgartner, Häfele, Maier-Häfele 2002).

Derartig strukturierte Lernumgebungen besitzen auch für Lernende im Bezug auf die Verfügbarkeit relevanter Lernmaterialien Vorteile, bieten jedoch – da sie zumeist aus rein statischen miteinander verlinkten HTML-Seiten bestehen – kaum didaktische Unterstützung, so dass die Frage nach dem so genannten Mehrwert solcher Lernumgebungen im Vergleich zu herkömmlichen Präsenzveranstaltungen und Lernmaterialien in den meisten Fällen unbeantwortet bleiben muss. Ein anderer Ansatz, wie er z. B. von (Methodenlehre-Baukasten 2003) oder (DFKI GmbH 2003) verwendet wird, verfolgt eine XML-basierte Kodierung des eigentlichen Lernmaterials und deren Beschreibung mittels Meta-Daten, wobei die Lernobjekte selbst in einzelnen Dateien vorliegen, welche im Hinblick auf eine Wiederverwendung frei miteinander kombiniert werden können. Jedoch sind auch hier Defizite hinsichtlich lernerspezifischer Präsentationsmöglichkeiten festzustellen, die sowohl ein individuelles ästhetisches Empfinden als auch bestimmte Lernpräferenzen betreffen, wie im Folgenden näher erläutert werden soll.

Anhand eines Szenarios wird die technische Umsetzung mit der frei verfügbaren Anwendung Apache Cocoon (Apache Software Foundation 2003) dargestellt. Cocoon ist eine Web-Anwendung zur Transformation von XML mittels XSLT in ein passendes Ausgabeformat wie HTML oder PDF unter Einbindung von einer in Java geschriebenen Programmlogik. Dadurch können Lerninhalte dynamisch nach jeweiligen Benutzervorgaben zusammengesetzt und in einer für den Lernenden angenehmen Art und Weise präsentiert werden. Hierbei ist sowohl die Art der angebotenen Informationen als auch die optische Darstellung wählbar.

Ein exemplarisches Szenario für eine solche Anwendung ist die Wiederholung von Lerninhalten für eine Prüfung, wobei der Lernende für ihn aktuell weniger relevante Teile des Kurses, wie z. B. Motivationen oder Übungen, ausgeblendet haben möchte, um sich statt dessen auf wesentliche Teile des Lernstoffes zu konzentrieren. Durch Selektion der gewünschten Teile des Lernmaterials werden beim Aufruf einer Lerneinheit nur diese zur Darstellung herangezogen. Des Weiteren ist es dem Lernenden möglich, seine Präferenzen hinsichtlich gewünschtem Schwierigkeitsgrad oder auch bevorzugter Darstellungsform (eher textbasiert, eher graphisch) festzulegen, um dann diesen Wünschen entsprechend eine möglichst passende Lerneinheit präsentiert zu bekommen.

In Abschnitt 2.1 beschreiben wir die Kodierung des Lernmaterials, um dann

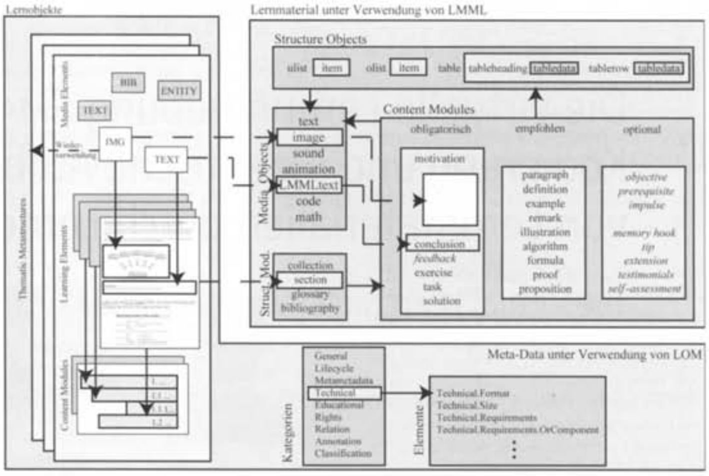


Bild 1: Aufbau des Lernmaterials mit Hilfe von Lernobjekten, die sich aus Meta-Daten und in LMML kodierten Inhalten zusammensetzen

in Abschnitt 2.2 auf die didaktische Gestaltung einzugehen. Die eingeführten Strukturen und Konzepte werden in Abschnitt 3 für die Beschreibung der in SmartFrame umgesetzten adaptiven Präsentationsformen verwendet. Hierzu zählen insbesondere die Navigation, die Präsentation und Visualisierung der Lernmaterialien sowie Konfigurationsoptionen. Abschnitt 4 beschreibt den Transformationsprozess der Lernobjekte zur Darstellung aus technischer Sicht. Abschnitt 5 schließt den Beitrag mit einem Ausblick auf weitere Entwicklungen ab.

2. Struktur und Aufbau der Lerninhalte

SmartFrame, als Prototyp einer virtuellen Lernumgebung, verfolgt den Ansatz modularisierter Lernobjekte, um neben einer leichten Wartbarkeit (Pflege), einer Redundanz im Sinne von mehreren Alternativen, die sich beispielsweise im Schwierigkeitsgrad oder Medieneinsatz unterscheiden, auch eine hochgradige Wiederverwendbarkeit zu erreichen (Seeberg 2003).

2.1 Kodierung des Lernmaterials

Bild 1 zeigt den grundsätzlichen Aufbau des Lernmaterials. Im linken Teil des Bildes wird die hierarchische Strukturierung der Lernobjekte aufgezeigt, wobei diese in verschiedenen Granularitätsstufen vorliegen.

Media Elements stellen die kleinsten Lernobjekte dar, welche zur Kodierung von Bildern, Texten oder bibliographischen Angaben dienen. *Learning Elements* stellen in der Regel eine Komposition der nicht weiter unterteilten *Media Elements* dar, um z. B. einzelne Lerneinheiten zu beschreiben. Weitere übergeordnete Aggregationsstufen sind die *Content Modules* für die Zusammenstellung von Kursen sowie *Thematic Metastructures* für gesamte Studiengänge oder Fachgebiete.

Die Lernobjekte sind hinsichtlich ihrer Eigenschaften zu spezifizieren, wobei hier neben technischen Angaben wie Größe oder notwendige Technologien auch didaktische Komponenten wie Schwierigkeitsgrad, Interaktivitätsgrad oder auch Verwendungskontext mit aufgenommen werden. Dafür wird in SmartFrame die um einige Elemente erweiterte IEEE-Spezifikation Learning Object Metadata (LOM (IEEE Learning Technology Standards Commitee 2003)) eingesetzt.

Die Inhalte werden in XML kodiert, wobei unterschiedliche Standards verwendet werden. In Bild 1 sind im rechten oberen Teil die aus didaktischer Sicht notwendigen Elemente zur Strukturierung von Lernmaterialien aufgeführt. Es handelt sich hierbei um die mit den kursiv dargestellten Elementen erweiterte Spezifikation Learning Material Markup Language (LMML (Süß 2003)). Anzumerken ist an dieser Stelle, dass die bei

LMML definierten Meta-Daten z.T. durch Erweiterungen der LOM-Spezifikation realisiert wurden (Reiners, Reiß, Sassen, Voß 2003). Weiterhin wird für Lernobjekte wie z.B. Übungsaufgaben die Spezifikation QTI, für bibliographische Angaben DocBook und für mathematische Gleichungssysteme MathML verwendet.

2.2 Didaktische Gestaltung

Die Architektur einer virtuellen Lernumgebung muss die individuelle kognitive Struktur des lernenden Individuums berücksichtigen können (Strittmatter, Niegemann 2002), wobei verschiedene theoretische Annahmen bezüglich der Informationsspeicherung bzw. -repräsentation von Lerninhalten bei den Lernenden existieren.

In diesem Zusammenhang werden im Folgenden kurz die drei wesentlichen, sich ergänzenden Theorieansätze aufgezeigt, da sie u.a. bedeutsam für die Darstellung des Lerninhaltes sind (Tulodziecki, Herzig 2002).

Die Theorie der Bedeutungsstrukturen nimmt an, dass Wissen in Form von semantischen Netzwerken als begriffliche Strukturen, die aus begrifflichen Elementen und ihren Relationen bestehen, mental repräsentiert wird. Die Theorie der Doppelkodierung besagt, dass Informationen nicht nur als begriffliche Repräsentation, sondern auch in analoger Form, wie z.B. in Bildern gespeichert werden, wobei Verbindungen zwischen beiden Repräsentationssystemen angenommen werden. Die Theorie der mentalen Modelle dagegen nimmt eine zusammenhängende mentale Repräsentation bestimmter Wirklichkeitsbereiche bzw. Probleminstanzen bezüglich ihrer strukturalen und funktionalen Aspekte im Gehirn an, wobei diese Repräsentation auf der Verarbeitung unterschiedlicher Zeichensysteme bzw. medialer Präsentationsformen beruhen kann.

Als eine wesentliche Konsequenz ist in der Architektur von SmartFrame bezüglich der Darstellungsmöglichkeiten der Lerninhalte ein breites Varianzspektrum hinsichtlich Darbietungsmodalität, Struktur, Schwierigkeitsgrad sowie Sequenzlänge und -abfolge bereits in der strukturellen Konzeption angelegt, um auch einer sehr heterogenen Gruppe von Lernenden einen positiven und effektiven Umgang mit der Lernumgebung zu ermöglichen. Auf diese Weise wird eine individualisierbare Umgebung geschaffen, die für vielfältige Lernstrategien stets offen bleibt.

SmartFrame ermöglicht den Lernenden die individuelle Gestaltung ihrer Lernumgebung entsprechend eigener Ansprüche, Vorlieben und kognitiver Stile auf mehreren Ebenen. Auf einer eher formalen Ebene sind dabei Konfigurationsmöglichkeiten betreffend Farbgestaltung, Anordnung der Buttonleisten, Buttons, Fenster, Spracheinstellungen, Kommunikationsformen sowie verschiedener Einstellungsmöglichkeiten der SmartBars zu nennen; siehe Abschnitt 3.2.

Aufgrund des modularen Aufbaus der Lernumgebung werden weitergehende Konfigurationsmöglichkeiten, die auf einer inhaltlichen Ebene angesiedelt werden können, realisierbar. Die Lerninhalte werden unter der Verwendung von LMML in semantische Module unterteilt. LMML verwendet zur Kodierung der Lerninhalte eine hierarchische Struktur und teilt die Elemente in vier größere Kategorien ein, die im Prinzip ihrer strukturellen Funktionalität innerhalb des Objektes entsprechen.

Bild 1 verdeutlicht weiterhin die Kombinierbarkeit der Elemente hinsichtlich der Möglichkeit ihrer gegenseitigen Aggregation. Ausgehend von den Strukturmodulen (*Structure Modules*) werden Inhaltsmodule festgelegt, die dem einzelnen Modul eine bestimmte semantische Position zuweisen. Inhaltsmodule (*Content Modules*) bestehen wiederum aus Struktur- (*Structure Objects*) sowie Medienobjekten (*Media Objects*), mit denen der Inhalt angegeben bzw. strukturiert wird.

Diese Inhaltsmodule dienen dem Aufbau von Lerneinheiten unter didaktischen sowie dynamischen und individuellen Aspekten. Dabei wird zwischen obligatorischen und optionalen Inhaltsmodulen unterschieden. Lerneinheiten in SmartFrame sind grundsätzlich durch ein Motivationsmodul (motivation) im Anfangsbereich gekennzeichnet, dem weitere Inhaltsmodule nachgestellt sind.

Der abschließende Bereich einer Lerneinheit besteht aus Konklusionsmodulen (*conclusion*) sowie ihrerseits wieder zu differenzierenden Übungsmodulen (*exercise*). Durch die Integration von optionalen Inhaltsmodulen kann eine Lerneinheit im Hinblick auf weitergehende Information, Lernmotivation oder Interaktionsmöglichkeiten bereichert werden.

Die Aufteilung des Lerninhaltes in semantische Module, die nach bestimmten Regeln zu einer Lektion zusammengesetzt werden können, bietet sowohl für die Autoren, die Kurse zu einem bestimmten Inhalt in die virtuelle Lernumgebung einstellen möchten, als auch für die Lernenden, die bei der Bearbeitung der eingestellten Lerninhalte verschiedene Formen von Wissen erlangen möchten, hinsichtlich Aspekten der Qualitätssicherung wesentliche Vorteile.

Autoren, die in didaktischen Fragen eher unerfahren sind, bekommen durch die grundsätzliche Festlegung der semantischen Module und der ihnen übergeordneten Struktur ein hilfreiches Werkzeug an die Hand, mit dem sie didaktisch sinnvoll strukturierte Kurse entwickeln können. Eine Umsetzung des Lernmaterials in die Module ist für Autoren arbeitsintensiver als ein Vorlesungsskript einfach digital in einem Content Management System abzulegen, doch gerade bei der Auswahl des geeigneten Moduls für einen bestimmten Inhalt oder eines bestimmten Inhaltes für ein spezifisches Modul, wird die für einen bestimmten Bereich charakteristische didaktische Struktur sinnvoll in das virtuelle Medium übertragen.

Die Vorteile, die aufgrund der differenzierten semantischen Modularisierung für die Lernenden entstehen, bestehen, neben didaktisch gut strukturierten Lektionen und Kursen, in den verschiedenen adaptiven Individualisierungsmöglichkeiten der präsentierten Inhalte, wie in den folgenden Abschnitten noch detailliert aufgezeigt werden soll.

3. (Adaptive) Präsentation

Der folgende Abschnitt bietet einen Überblick über Navigations- und Konfigurationsmöglichkeiten von SmartFrame. Der Übergang von Konfiguration zu Adaptivität im Sinne einer automatischen Adaption, ohne direktes bzw. explizites Einwirken des Lernenden, an bestimmte Vorlieben bzw. Merkmale des

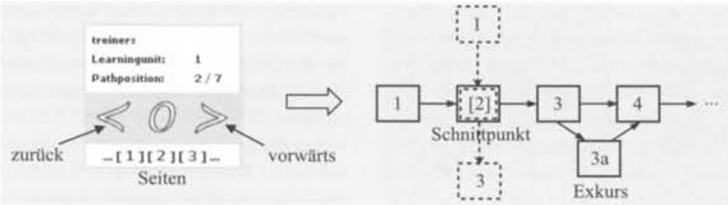


Bild 2: Lernpfade und Navigationsschnittstelle

Lernenden, kann hierbei als fließend gelten, da insbesondere auch die Möglichkeit besteht, den Adaptionsgrad hinsichtlich bestimmter Kriterien selbst festzulegen.

3.1 Navigation

Um individuelle Lernprozesse zu fördern, sind neben zahlreichen Konfigurationsmöglichkeiten ebenso viele Freiheitsgrade bei der Navigation innerhalb des Lernmaterials als relevant anzusehen. Auf diese Weise können in Abhängigkeit eines aktuellen Lernmodus sowohl die Ansichten (Menge, Art und Form) als auch die Navigationsmöglichkeiten variieren. Sämtliche Navigationsmöglichkeiten innerhalb von SmartFrame spannen sich grundsätzlich zwischen den beiden nachfolgenden gegensätzlichen Navigationsarten auf.

Lernpfade (siehe Bild 2) stellen in gewisser Hinsicht die klassische Form der Präsentation von Lernmaterialien in Analogie zu textbasierten Lernutensilien wie z. B. Lehrbüchern dar (Schulmeister 2002). Der Lehrende definiert aus der Menge aller Lernobjekte eine Folge, die dem Lernenden angeboten wird. Dieser kann den Lernpfad iterativ durchlaufen und somit die angedachte Struktur und den semantischen Ablauf vollständig übernehmen. Lernpfade in virtuellen Lernumgebungen können Erweiterungen wie z. B. Exkurse bereitstellen. D.h., möchte der Lernende zum aktuellen Zeitpunkt seines Lernprozesses einen bestimmten Inhalt vertiefend erarbeiten, so kann er auf einen untergeordneten Lernpfad wechseln und anschließend mit dem übergeordneten Lernpfad fortfahren. Es werden also Schnittpunkte bzw. Kreuzungen zu anderen Lernpfaden geschaffen, bei denen es sich um gemeinsame Lernobjekte handelt, welche in unterschiedlichen Lernpfaden, und damit Kontexten, eingesetzt werden.

Browsing bedeutet ein mehr oder weniger freies Durchwandern der verfügbaren Lernmaterialien durch das Verfolgen von Verknüpfungen zwischen Objekten. Hierbei wird keine strenge Reihenfolge wie bei den Lernpfaden vorgegeben, sondern vielmehr kann und soll der Lernende selbstständig entscheiden, welches Lernobjekt als nächstes angezeigt werden soll. Verlinkungen, Stichwortverzeichnisse und Suchoptionen unterstützen den Lernenden bei dieser freien Form der Navigation.

Die Darstellung der Lernobjekte kann rein textuell wie bei einem klassischen Inhaltsverzeichnis erfolgen, bei dem die Titel in einer hierarchischen Struktur angezeigt werden. Eine weitergehende Darstellung ist die Verwendung von hyperbolischen Bäumen, in denen das Prinzip des advance organizers erweitert wird. Das aktuell angezeigte Lernobjekt wird zentriert dargestellt und verlinkte Lernobjekte radial (je nach Tiefe mit einem größer werdenden Radius und verkleinert) um das Zentrum herum angeordnet.

Noch mehr Funktionalität und eine ganzheitliche Ansicht sämtlicher Beziehungen zwischen den einzelnen Lernobjekten bieten hyperbolische Graphen; siehe Bild 3. Hier sind zusammengehörige Lernobjekte mit Kanten in einer netzwerkartigen Struktur miteinander verbunden. Der Lernende bekommt dadurch einen Überblick über Zusammenhänge, der über die schlichtere Anzeige der Hierarchiebeziehungen von hyperbolischen Bäumen hinausgeht, und kann durch die Inhalte in einer Form navigieren, die man in Analogie zu dem Gebrauch interaktiver Landkarten verstehen kann.

Zusätzliche Anwendung von Filtern erlaubt eine selektive Darstellung beispielsweise prüfungsrelevanter bzw. noch zu lernender Lernobjekte.

3.2 Konfiguration der Darstellung sowie Zusammenstellung des Lernmaterials

Es wird eine Standardkonfiguration von SmartFrame für die Oberfläche wie auch für die Lernmaterialien vorgegeben, die jedoch sowohl im Aussehen – hierzu gehören Farben und Anordnung der Elemente – als auch in der Zusammenstellung – Auswahl der darzustellenden Lernobjekte – modifiziert werden kann. Bei der Konfiguration erfolgt eine Unterteilung in **inhaltliche Optionen** für die Darstellung des Lernmaterials, z. B. Formalität, Interaktivitätsgrad, Orientierung, Schwierigkeitsgrad oder Adaptivität sowie **formale Optionen** für Darstellung von Lernobjekten, Farbauswahl, Sprache, SmartBars oder Anzeige von Informationen.

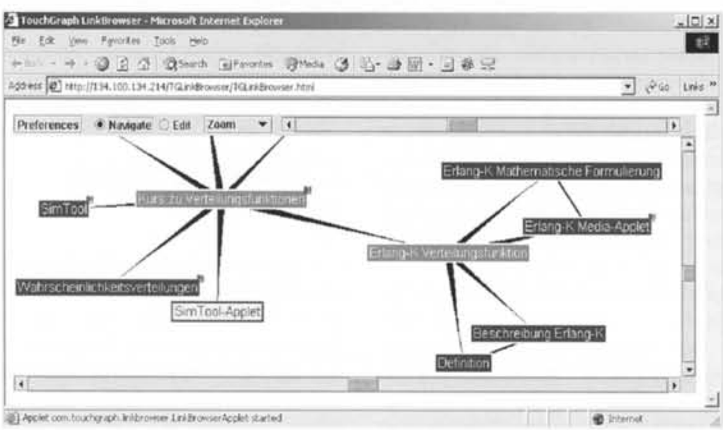


Bild 3: Hyperbolischer Graph

Inhaltliche Optionen: Durch die in Abschnitt 2.1 beschriebene explizite Unterteilung des Lernmaterials in die spezifischen semantischen Module ist es den Lernenden gestattet, auch hinsichtlich ihrer Darstellung individuelle Einstellungen vorzunehmen.

Der Lernende kann entscheiden, inwiefern bestimmte Modultypen für ihn relevant sind und bei der Präsentation vorkommen sollen. So können z.B. die Motivations- oder Impulsmodule besonders hervorgehoben werden oder nur die Übungsmodule angezeigt werden, falls der Lernende die Umgebung nicht zur Aneignung neuer Inhalte sondern nur zur Vertiefung des bereits angeeigneten Lernstoffes nutzen möchte.

Die Auswahl kann sich auch auf die inhaltliche Ausrichtung der Lernmodule beziehen, so dass auch bestimmte Lernerprofile unterstützt werden. Je nachdem, ob der Lernende beispielsweise eher mathematisch-naturwissenschaftlich oder künstlerisch-geisteswissenschaftlich orientiert ist, kann die Lernumgebung die dementsprechenden Lernmodule in bevorzugter Präsentationsform anzeigen.

Wie die Adaptivität der Umgebung an den Lernenden umgesetzt ist, wird exemplarisch in Bild 4 gezeigt. Die mit den Meta-Daten spezifizierten Lernobjekte können über Relationen miteinander in Beziehung gesetzt werden. So wird das Learning Element in zwei Versionen vorgehalten, welche sich in Zusammensetzung (Text und Applet versus Text, Bilder und mehrere Verweise auf Literaturstellen), Schwierigkeitsgrad und Kontext unterscheiden. Der Lernende wählt direkt oder über Lernerprofile sein präferiertes Lernerlebnis, welches dann dynamisch während des Lernprozesses durch eine geeignete Auswahl der Lernobjekte geboten wird.

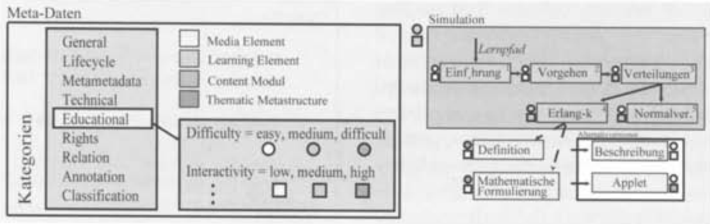


Bild 4: Adaptivität des Lernmaterials

Die sehr wünschenswerte Unterstützung individueller Lernprozesse durch eine hochgradig adaptive und konfigurierbare virtuelle Lernumgebung birgt leicht die Gefahr in sich, den Lernenden aufgrund der ihm gebotenen Handlungsvielfalt zu überfordern und von seinem eigentlichen Lernvorhaben abzubringen anstatt es zu fördern. In SMARTFRAME soll eine solche Überforderung des Lernenden vermieden werden, indem die ihm gebotene Oberfläche trotz vielfältiger Nutzungsmöglichkeiten so einfach wie möglich gehalten wird.

Formale Optionen: Die Gestaltung der virtuellen Lernumgebung und insbesondere der Lernmaterialien besitzt ein wesentliches Motivationspotenzial, das durch einen hohen Einflussgrad der Lernenden bei der Konfiguration noch gesteigert werden kann.

Im Gegensatz zu den zuvor beschriebenen adaptiven Komponenten können die formalen Optionen bezüglich einer Standardeinstellung vielfältig variiert werden. Für die Lernobjekte sind mehrere Stilvorlagen entsprechend des Bildes 5 vorgesehen, bei denen jede Komponente durch Einträge innerhalb von Cascading Style Sheets modifiziert werden kann. Dadurch können individuelle, lernobjektspezifische Darstellungen ge-

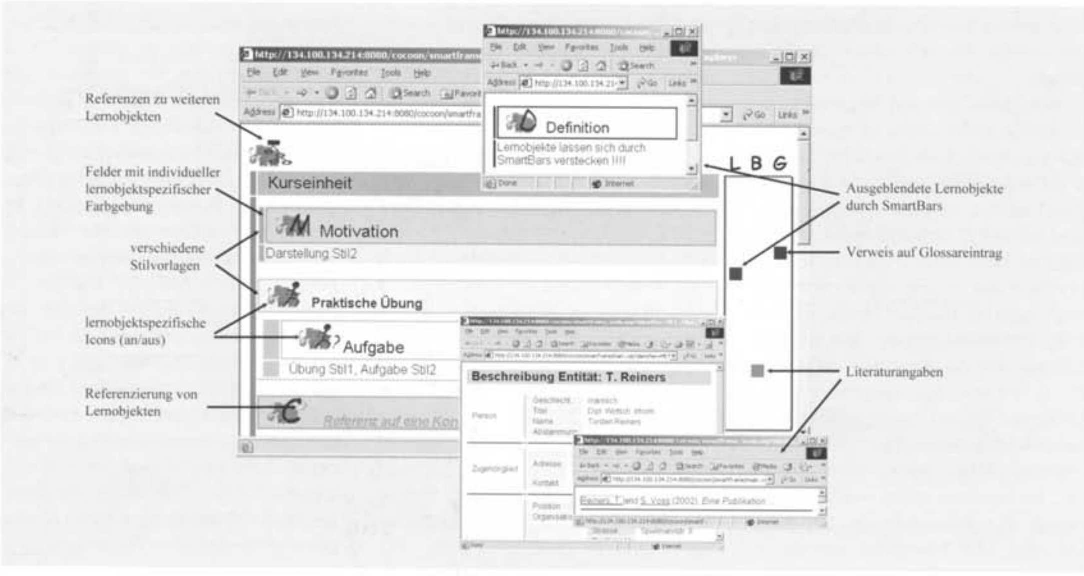


Bild 5: Gestaltung des Lernmaterials

wählt werden, die den Lernvorgang durch geeignete Schriften, Farben, Kompositionen und Ausrichtungen nachhaltig fördern. Weitere formale Optionen, die durch die Lernumgebung nicht modifiziert werden sollten, sind die Wahl der Sprache und angezeigte Informationen innerhalb der SmartBars.
Dies sind am Rand dargestellte Markierungen, welche auf unterschiedliche Informationen wie z. B. Literaturangaben, Glossareinträge oder Lernobjekte verweisen, die erst bei Aktivierung durch Anklicken oder Überfahren dargestellt werden; siehe rechte Seite in Bild 5 sowie (Reiners, Reiß, Voß 2002) und (Reiners, Reiß, Sassen, Voß 2003) für eine detaillierte Beschreibung.

4. Lernmaterial

Die Komposition von Lernmaterialien und deren Visualisierung in Form von z.B. einer Web-Seite oder eines PDF-Dokumentes sollte hinsichtlich einer größtmöglichen Adaptivität an den Lernenden zeitnah, d.h. beim Aufruf der Seite, erzeugt werden. Dadurch kann direkt auf vom Lerner spezifizierte Lernmodi und Bedürfnisse reagiert werden, welche in dem hier beschriebenen Ansatz in einer XML kodierten Konfigurationsdatei vorliegen müssen; siehe den Datei-Auszug in Bild 6. Ausgehend von einer Menge an Lernobjekten werden dann mittels Transformationen die entsprechenden Seiten generiert und ausgegeben.

Hier gehen wir von folgenden Annahmen aus: Der Lernende nimmt erstmalig an einem Kurs *Simulation* teil und ist somit als Anfänger einzustufen. Aufgrund dessen wird der Schwierigkeitsgrad (difficulty) auf leicht (easy) und der Wunsch nach Interaktivität (interactivity) innerhalb der Lernmaterialien auf hoch (high) gesetzt. Darüber hinaus werden in der Konfigurationsdatei Darstellungsoptionen für die einzelnen LMML-Elemente definiert. Nach dem bereits eingeführten Szenario, bei dem der Lernende keine Übungen durchführen möchte, wird das LMML-Element exercise über die Konfigurationsdatei ausgeblendet (none). Für darzustellende Elemente, hier das LMML-Element motivation, können unterschiedliche Visualisierungsstile (z. B. style2), Farbgebungen

```
<user>
  ...
    <param group='global' name='interactivity'>high</param>
    <param group='global' name='difficulty'>easy</param>
      ...
    <param group='exercise' name='display'>none</param>
      ...
    <param group='motivation' name='display'>style2</param>
    <param group='motivation' name='display_color'>css</param>
    <param group='motivation' name='icon'>icon</param>
</user>
```

Bild 6: Konfigurationsdatei

(z. B. aus einem Cascading Style Sheet (CSS)) sowie die Einblendung von Icons definiert werden.

4.1 Aufbau und Strukturierung

Im Folgenden gehen wir von dem in Bild 4 dargestellten Kursausschnitt aus. Der Kurs *Simulation* (*Content Module*) setzt sich aus mehreren *Learning Elements* zusammen (*Einführung, Vorgehen, Verteilung*), die wiederum in weitere *Learning Elements* (*Erlang-K, Normalver.*) bzw. *Media Elements* aufgeteilt sind. Das *Learning Element Erlang-K* bindet zwei weitere *Media Elements* ein. Weiterhin liegen zu diesen *Media Elements* jeweils alternative Versionen mit unterschiedlichem Schwierigkeits- sowie Interaktivitätsgrad vor. Jedes Lernobjekt wird mit Meta-Daten beschrieben, wobei Lernobjekte mit mehreren untergeordneten Objekten eine zusammenfassende Bewertung haben. Der Lernpfad – eine vom Autor festgelegte Folge von Lernobjekten, in der der Lernende die Kurseinheit durchlaufen soll – wird über die *Learning Elements* definiert.

Die Inhalte selbst sind für eine Verwendung in Cocoon wie in der in Bild 7 dargestellten Datei zu kodieren. Neben einer Einbindung von notwendigen Java und JavaScript-Komponenten durch <smartframe:header> wird durch <smartframe:le identifier='erlangk'> die Granularitätsstufe *Learning Element* festgelegt sowie das Objekt mit einem eindeutigen Bezeichner erlangk versehen. Durch das Tag <smartframe: link ... /> werden die dem *Learning Element* untergeordneten *Media Elements* eingebunden, wobei hier ein direkter Bezug auf die Lernobjekte *definition* sowie *mathform* genommen wird, d. h. diejenigen des höheren Schwierigkeitsgrades.

Bei der Einbindung von untergeordneten Lernobjekten werden grundsätzlich zwei Formen (mit je zwei Variationen) unterschieden. Bei einer direkten Einbeziehung des Lernobjektes (include) wird der komplette Inhalt eingebunden, bei einer Referenzierung (reference) werden Informationen aus dem Lernobjekt extrahiert, die zur Darstellung eines Verweises benutzt werden. Bei internen Referenzierungen (internal) werden die auf dem Server lokal gespeicherten Lernobjekte angesprochen, bei externen Verweisen (external) kann eine beliebige URL Verwendung finden.

4.2 Selektion und Komposition

Mittels der URL http://www.smartframe.de:8080/cocoon/smartframe/main.xsp?identifier=id wird über eine Cocoon-interne Konfigurationsdatei die Datei *main.xsp* mit id = *erlangk* aufgerufen, wobei erlangk das konkrete angeforderte Lernobjekt spezifiziert. Diese Datei wird von einem so genannten Serverpages-Generator eingelesen und verarbeitet. Die vorrangige Funktionalität innerhalb dieser Datei wird von einem so genannten Logicsheet übernommen, welches bei Übereinstimmung mit einem vorher definierten XML-Namespace sowie Tag-Namen bestimmte Ersetzungen bzw. Funktionen ausführt. In diesem Fall wird zunächst das XML-Tag *<smartframe:header/>* durch für weitere Schritte benötigten Java-Code ersetzt. Die Elemente *<smartframe:link ... />* bewirken entweder eine direkte Einbeziehung (includeinternal) des dort referenzierten (und bereits in einem untergeordneten Transformationsprozess bearbeiteten) Lernobjektes oder den Einbau des Elementes <a href='main.xsp?identifier=id'/> als Verweis auf das entsprechende Lernobjekt (referenceinternal).

```
<xsp:page language='java'
  xmlns:xsp='http://apache.org/xsp'
  xmlns:smartframe='http://www.smartframe.de/xsl'
  xmlns:cinclude='http://apache.org/cocoon/include/1.0'>

  <smartframe:header/>

  <smartframe:le identifier='erlangk'>
   <lmml xml:lang='de'>
     ...
   <smartframe:link type='includeinternal' identifier='definition'/>
   <smartframe:link type='includeinternal' identifier='mathform'/>
     ...
   </lmml>
   <lmml xml:lang='en'>
     ...
   </lmml>
  </smartframe:le>
</xsp:page>
```

Bild 7: Kodierung von Lerninhalten in XML

Für eine Zuordnung von id zu der in der LOM-Datenbank vorliegenden URL des eigentlichen Lernobjektes werden sowohl eine diesbezügliche Anfrage gestellt als auch Angaben zum MIME-Typ das Lernobjektes angefordert. Bei Lernobjekten vom MIME-Typ image/gif ist beispielsweise keine weitere Transformation notwendig, so dass direkt die Referenz auf das Bild in den HTML-Code eingefügt wird. Bei Lernobjekten in XML wird innerhalb eines Transformationsprozesses der Inhalt des referenzierten Objektes mittels eines so genannten CInclude-Transformers in das ursprüngliche Dokument eingebunden, wobei ein rekursiver Aufruf des Vorgehens für jeden weiteren enthaltenen Link (<smartframe:link .../>) erfolgt (vgl. Bild 8). Der Transformationsprozess erfolgt mittels einer so genannten Pipeline, bei der vermöge eines Generators (bei der hier beschriebenen Transformation von Lernobjekten ein Serverpages-Generator zum

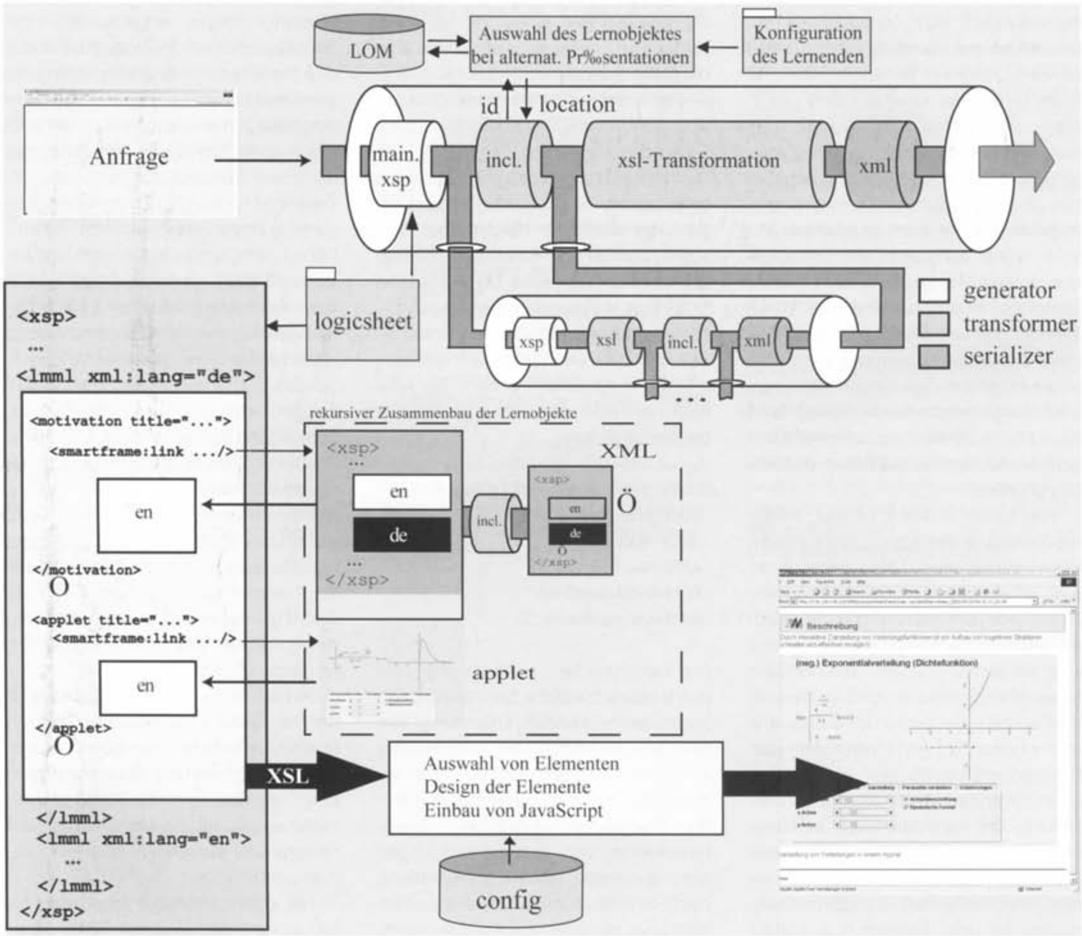


Bild 8: Transformationsprozess

Einlesen und Verarbeiten einer XSP-Datei) interne SAX-Events generiert werden, welche innerhalb der Pipeline entsprechend potenziell eingehängter Transformer (z.B. ein XSLTransformer zur Bearbeitung des SAX-Stromes mittels XSLT) modifiziert werden. Am Ende einer jeden Pipeline erzeugt ein Serializer ein bestimmtes Ausgabeformat, beispielsweise XML oder HTML.

Bei der Transformation der Lernmaterialien in das Ausgabeformat sind zwei wesentliche Schritte bezüglich der Adaption an den Lernenden hervorzuheben. Dieses sind zum einen die bereits angesprochene Auswahl der lernprozessspezifischen Lernobjekte sowie zum anderen die Darstellung in der geforderten Sprache (sofern verfügbar). Für die Auswahl der Lernobjekte ist in LOM der Beziehungstyp *isformatof* bzw. *hasformat* vorgesehen, mit dem alternative Darstellungen von Lernobjekten mit demselben Inhalt in Bezug gebracht werden. Aufgrund der Meta-Daten für jedes Lernobjekt kann aus den möglichen Lernobjekten jenes gewählt werden, welches die größtmögliche Übereinstimmung mit den in Bild 4 gegebenen lernerspezifischen Konfigurationsangaben hat. In dem gegebenen Beispiel wird aufgrund der Entscheidungsregeln das ursprünglich direkt im Lernobjekt erlangk vorgesehene *Media Element definition* durch das *Media Element* description substituiert, indem der in die Transformation eingehende identifier ausgetauscht wird. Analog wird dem *Media Element applet* gegenüber dem *Media Element mathform* der Vorzug gegeben.

Wie bereits in Bild 7 gezeigt, erfolgt die Darstellung der eigentlichen (vollständigen) Inhalte eines Lernobjektes in jeweils einem eigenen XML-Tag <lmml>, wobei jede existierende Sprache durch das Attribut xml:lang gekennzeichnet wird. Bei der XSL-Transformation wird der entsprechende Inhalt in Abhängigkeit der Konfiguration des Lernenden ausgewählt und nur dieser Teil des Lernobjektes in der Pipeline weiterverarbeitet.

Im folgenden Abschnitt wird die Darstellung der Lernmaterialien beschrieben. Hierzu wird nach dem Zusammenfügen aller verlinkten Lernobjekte in einem abschließenden Transformationsprozess für jedes Element (z.B. LMML) die vom Lernenden vorgegeben Konfiguration angewendet. D.h., dass Verweise auf Cascading Style Sheet-Elemente eingebunden werden, mit Hilfe von Tabellen eine Anordnung der Elemente erfolgt oder komplette Elemente nicht dargestellt werden. Darüber hinaus werden notwendige JavaScript-Komponenten einbezogen, die z.B. bei der Darstellung von SVG-Grafiken notwendig sind.

4.3 Visualisierung

Für ein ganzheitliches Lernerlebnis wird die Adaptivität bei der Darstellung fortgeführt. Neben kosmetischen Details, wie die unterschiedliche Darstellung von Autoren bei bibliographischen Angaben, stehen dem Lernenden vielfältige Variationen zur Verfügung, die bei der Transformation nach HTML Berücksichtigung finden. In der zuvor gezeigten Konfigurationsdatei kann der Lernende elementspezifisch einen Stil festlegen, zusätzliche Icons einblenden sowie über die Farbgebung entscheiden.

Bild 5 zeigt eine Übersicht über die aktuellen Designmöglichkeiten. Dem Lernenden bietet sich dabei über CSS die Möglichkeit, jeden Bereich eines Lernobjektes individuell festzulegen; z.B. wird durch den dargestellten Ausschnitt aus der CSS-Datei das Aussehen des Überschriftentextes der Motivation festgelegt. Insgesamt lassen sich auf diese Art und Weise alle Komponenten eines jeden Stils unterschiedlich variieren; z.B. die vorgestellten Balken, die Rahmen um Felder oder die Hintergründe bei Textbereichen.

```
h3.me_text_motivation_titlebar {
 background-color:#dbe8ac;
 color:#000000;
 font-size:110 %;
 font-family:verdana;
 font-weight:bolder; }
```

Die hierarchische Strukturierung wird durch unterschiedliche Einrückungen erreicht, eine schnelle Erkennung von Lernobjekten durch die Verwendung von Icons. Weiterhin werden (individuell zusammengestellte) Lernerprofile mit in die Darstellung einbezogen, indem Lernobjekte von bestimmten Typen nicht dargestellt, als Icons am Anfang der Seite oder als reine Referenz positioniert bzw. nur über so genannte SmartBars visualisiert werden.

5. Konklusion

SmartFrame ermöglicht eine ganzheitliche Adaption der Lernumgebung an individuelle Lernprozesse, die über die bereits übliche Interaktivität einzelner Lernmodule und benutzerspezifischer Eingangskonfigurationen weit hinausgeht. Inwieweit die Lernumgebung auf das Lernverhalten (d.h. auf die Auswahl der Lernobjekte, Ergebnisse von Übungen, Aufsuchen von Hilfedateien oder Zusatzinformationen) adaptiv reagiert, kann von dem Lernenden selbst ausgewählt werden; auf diese Weise wird dem Gefühl einer „Bevormundung durch die Lernumgebung" vorgebeugt.

Die beschriebene Transformation der Lernobjekte kann aufgrund des gewählten Ansatzes des webbasierten Aufrufs direkt in bereits existierende virtuelle Lernumgebungen eingebunden werden. Bezüglich der Konfigurationsdateien kann einerseits eine Erweiterung der virtuellen Lernumgebung erfolgen, andererseits bietet sich aber auch die Definition einer Standardkonfiguration an, die jedem Benutzer zur Verfügung gestellt wird, wodurch sich zumindest bestimmte Lernszenarien abbilden lassen. Die vorgestellten Komponenten sind Bestandteil eines größeren Projektes, bei dem ein Prototyp für eine virtuelle Lernumgebung sowohl konzipiert als auch implementiert wird (Reiners, Voß, Reiß, Schulze 2003) und (SmartFrame 2003). Bei der Entwicklung der virtuellen Lernumgebung stehen im Wesentlichen die Evaluierung von neuen Technologien, innovative Ansätze wie z.B. SmartBars, Navigationsoptionen innerhalb der Lernmaterialien wie z.B. durch Einsatz von hyperbolischen Graphen, weitergehende Kommunikationsformen sowie didaktische Aufbereitung von Lernmaterialien inklusive einer Adaptivität an den Lernenden im Vordergrund.

Aktuell befindet sich SmartFrame in der Test- und Evaluationsphase. Mit Hilfe eines gestaffelten Evaluationskonzeptes soll eruiert werden, ob und inwiefern durch das beschriebene Konzept eine Verbesserung für die Situation der Lernenden und Lehrenden insgesamt erzielt werden kann.

Mit einem ressourcensparenden Effekt ist zumindest in der Einführungszeit nicht zu rechnen, da die inhaltliche Neustrukturierung der jeweiligen Lernmaterialien mit Hilfe der in diesem Beitrag vorgestellten (semantischen) Module für die Lehrenden bzw. Autoren von Kursen eine nicht zu unterschätzende Aufgabe bleibt. Neben der Umsetzung wissenschaftlicher Erkenntnisse ist auch eine vielfach intuitive und kreative Arbeit nötig, um didaktisch hochwertige Lernmodule zu erstellen. Daher kann es keine Lerntechnologie geben, die schablonenartig angewendet stets gute Ergebnisse erzielt. Durch die Entwicklung eines differenzierten, auf die Architektur der Lernumgebung abgestimmten Autorentools, das in naher Zukunft gemeinsam mit SmartFrame eingesetzt wird, kann jedoch den Autoren viel Arbeit im Vorfeld durch Vorgehensmodelle sowie Anwendungen zur einfachen Eingabe der Lerninhalte erspart und auf diese Weise eine wesentliche Unterstützung geboten werden.

Zukünftig ist geplant, SmartFrame in ein Open-Source-Projekt zu überführen, um Interessierte direkt an der Entwicklung moderner Lehr- und Lerntechnologien zu beteiligen.

6. Danksagung

Diese Arbeit wurde vom BMBF (Bundesministerium für Bildung und Forschung) im Rahmen des Programms „Neue Medien in der Bildung" (01NM094D/ 01NM094E) unterstützt.

Literatur

Apache Software Foundation. Apache Cocoon. **http://xml.apache.org/cocoon (Letzter Zugriff: 01.10.2003).**

Baumgartner, P.; Häfele, H.; Maier-Häfele, K.: *E-Learning Praxishandbuch – Auswahl von Lernplattformen: Marktübersicht – Funktionen – Fachbegriffe.* Innsbruck: Studienverlag, 2002.

DFKI GmbH. Das ActiveMath Projekt. **http://www.activemath.org (Letzter Zugriff: 01.10.2003).**

IEEE Learning Technology Standards Committee: Standard for Information Technology: Education and Training Systems - Learning Objects and Metadata. **http://ltsc.ieee.org/' wg12 (Letzter Zugriff: 01.10.2003).**

Kretschmer, M.: *E-Learning in der Hochschulausbildung: Projektmanagement für ein hochschulübergreifendes Partner-Netzwerk.* Books on Demand, 2002.

Methodenlehre-Baukasten. **http://www.izhd.uni-hamburg.de/ mlbk/home.html (Letzter Zugriff: 01.10.2003).**

Reiners, T.; Reiß, D.; Sassen, I.; Voß, S.: Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: Von der Technik zur Anwendung. In: *E-Learning in Wirtschaftsinformatik und Operations Research.* (Hrsg. Suhl, L.; Voß, S.) Books on Demand, 2003, 77-115.

Reiners, T.; Voß, S.; Reiß, D.; Schulze, H.: *SMARTFRAME: An Integrated Environment for XML-coded Learning Material.* In Uhr, W.; Esswein, W.; Schoop, E. (Hrsg.): Wirtschaftsinformatik 2003/ Band I. Physica, Heidelberg, 2003, 613-632.

Reiners, T.; Reiß, D.; Voß, S.: Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts. In: *Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002)* **#100029-03-TR-026** (2002). Millet Alberta, ICSC-Naiso Academic Press [ISBN: 3-906454-31-2], 1-7.

Süß, C.: Learning Material Markup Language. **http://www.lmml.de (Letzter Zugriff: 01.10.2003).**

Schulmeister, R.: *Grundlagen hypermedialer Lernsysteme: Theorie - Didaktik - Design.* Oldenbourg, 2002.

Seeberg, C.: *Life Long Learning: Modulare Wissensbasen für elektronische Lernumgebungen.* Heidelberg, Springer, 2003.

SmartFrame. **http://www.smartframe.de (Letzter Zugriff: 01.10.2003)**

Strittmatter, P.; Niegemann, H.: *Lehren und Lernen mit Medien.* Darmstadt, Wissenschaftliche Buchgesellschaft, 2000.

Tulodziecki, G.; Herzig, B.: *Computer & Internet im Unterricht: Medienpädagogische Grundlagen und Beispiele. Berlin, Cornelsen,* 2002.


1

2

3

4

1 Dipl. Wirtsch.-Inform. Torsten Reiners, wiss. Mitarbeiter an der Abteilung für ABWL, Wirtschaftsinformatik und Informationsmanagement, Braunschweig; Forschungsschwerpunkt: Virtuelle Lernumgebungen
E-Mail: t.reiners@tu-bs.de

2 cand. Dipl. Wirtsch.-Inform. Dirk Reiß, Institut für Wirtschaftsinformatik, Hamburg; Forschungsschwerpunkt: Virtuelle Lernumgebungen
E-Mail: d.reiss@tu-bs.de

3 Imke Sassen, Kunst- und Medienpäd., wiss. Mitarbeiterin am Lehrstuhl für Betriebswirtschaftslehre: Management Science der Universität Magdeburg; Forschungsschwerpunkt: Virtuelles Lernen
E-Mail: imke.sassen@wirtschafts-w.uni-magdeburg.de

4 Prof. Dr. Stefan Voß, Geschäftsführender Leiter des Institutes für Wirtschaftsinformatik der Universität Hamburg; Forschungsschwerpunkt: Wirtschaftsinformatik
E-Mail: stefan.voss@uni-hamburg.de

Synchronized Blended Learning in Virtual Learning Environments

Christine Frank, Leena Suhl
University of Paderborn, Germany
{frank, suhl}@dsor.upb.de

Torsten Reiners, Imke Sassen, Stefan Voß
{reiners, sassen}@econ.uni-hamburg.de, stefan.voss@uni-hamburg.de

Abstract: The main focus of this paper is to show state of the art technologies and existing components of didactical concepts which can be composed in certain ways to enable collaborative learning via the web. We introduce pedagogical paradigms, upon which we build our understanding, approach, and implementation of synchronized blended learning. This paper also describes our technological and system architecture including up-to-date standards for a possible virtual learning environment and how it can be applied to support the synchronized blended learning approach. Practical examples of our approach, as well as diverse didactical settings in which our virtual learning environment can be used are also given.

1. Introduction

Didactical beliefs are changing over time and new ones are emerging to compete against already approved models. Within last years, one development of the belief was going in the direction of so-called blended learning. Blended learning implies that already known components of different methods, concepts, and didactical beliefs, usually used in isolation, are now combined flexibly to suit the needs of the learner for a holistic learning experience. However, blended learning seemed to be a buzzword for companies to improve their e-learning products over a long period, whereas the number of scientific publications is growing; see, e.g., Bonk (2005), Wilson and Smilanich (2005). In this paper we demonstrate how the idea of blended learning can be transformed into a consistent concept combining methods, didactical settings, and learning paradigms of already existing and successful conceptions of learning using a *virtual learning environment* (VLE) as a learning framework. Furthermore, we extend the term blended learning with the additional term *synchronized* to characterize the integration of innovative technologies and learning methodologies into our concept. Synchronized blended learning incorporates a contemporary, (semi-)automized synchronization process of the learning material between different learning modes such that all learners have access to the latest material (e.g., changes of the learning path within the classroom presentation are projected directly to the self-paced study; see Section 2.2).

The concept outlined in this paper incorporates an approach that allows learners to obtain simple, flexible, and configurable support for an independent—in terms of time and location—learning process without losing the social aspect of inter-personal communication. Especially factors like location independent learning, navigation, communication, information presentation, and configuration are not extensively integrated in most system architectures. In Section 2 we present the pedagogical paradigms of the VLE. The advantages of synchronized blended learning are demonstrated in a short example. In Section 3 we briefly summarize our system architecture, which integrates several components to support the tutor and learner as well as synchronized blended learning, but also like to refer to other publications due to the focus of this paper; see, e.g., Reiß and Reiners (2004), Reiners et al. (2003b,c).

2. Pedagogical Paradigms

Because pedagogical and didactical aspects of possible teaching scenarios play a major role within the development and implementation of a VLE it is essential to identify and characterize the main target group before starting to develop a concept or design. One of the main success factors is a preliminary analysis of the target group concerning their affective, situative and cognitive affiliations (Frank et al., 2002a) towards the use and acceptance of new technologies and communication (e.g., e-mail, discussion boards or internet in general). Afterwards, the expectations of the target group should be evaluated qualitatively as well as quantitatively. Frank et al., 2002b, describe how this empirical research can be done to support customization of a VLE. To fulfill the expectations of those who are going to use the later implementation it is essential to build upon the learner-given fundamentals.

Foundation for our framework is a moderate constructivist belief thus emphasizing the importance of authentic, multi layered, and diverse contexts as well as the social interaction. One main advantage of using hypermedia is its non-linear structure and the possibility to integrate diverse forms of presenting the learning material. Using hypermedia is especially effective when practicing problem-based learning because the structure of an extensive VLE closely resembles traditional learning environments. Learners can access references, literature, or articles in a similar way as in a traditional constructivist setting.

2.1. Variations of the Didactical Context

One of the main success factors regarding the motivation of the learners is the integration of diverse didactical settings within a VLE allowing an adequate selection in terms of finding the best individual learning experience. Personal preferences need to be considered; especially the possibility to a change of the medium of the learning material must be given at every point of time. The settings should be used according to their strengths, emphasizing the enhancement given by the inclusion of other methods. We describe three interesting didactical settings for a VLE, which may be combined miscellaneously within a synchronized blended learning approach. Note that all didactical settings can also be combined with non-virtual forms of presentation within a blended learning process.

Self-paced Study: Learners navigate or browse (freely) through the VLE. On the one hand the navigation can be self-guided in terms of allowing the learner to follow links and references presented within the learning material similar to hyperlinks used to connect websites on the Internet. On the other hand, the learner may follow a certain given link (similar to a guided tour but not corresponding with certain course material) or use an extensive index, glossary, or search option to find interesting learning material.

Virtual Course: Virtual courses resemble guided tours, for example, a tour of a given lecture. Based on the presented learning material in the classroom, the tutor defines a learning path that covers all relevant topics in a specified order. The learner is guided through the synchronized material from the classroom allowing him to learn the same content which was presented within the classroom. Furthermore, the virtual course extends the material by links and references to other relevant topics as well as allowing the change of the learning path to individual preferences like the order of the material or the style of presentation.

Synchronized Blended Learning: As mentioned in the introduction, the term *blended learning* is not yet scientifically defined. It is mostly used in a popular scientific way to give the combination of existing approaches a new name. Our definition for synchronized blended learning integrates blended learning into a global approach thus improving the learning experience within a virtual learning environment.

> *Blended learning is the combination of different learning methods and didactical concepts allowing the learner to select the most promising form of learning for a successful learning experience. The learner participates in a learning program to achieve a certain goal given by the teacher. "Plain" blended learning uses this context to offer different learning methods and didactical concepts. Synchronized Blended Learning builds on this and also supports the freedom to choose between didactical methods and technologies. Synchronized Blended Learning integrates an adaptive synchronization process to directly correlate pedagogical components of the different forms of learning methods and didactical concepts. This has to be realized and assisted, respectively, by one comprehensive technology allowing the learner to study within a uniform environment without the need to change media.*

Learners belonging to a specific group who all have a common learning goal and a common time schedule in mind (e.g., preparation for an exam) have the opportunity to learn with a mixture of presence phases and self study phases. This approach utilizes and innovatively combines pedagogical components—that have been shown to be effective within different settings—and integrates them into a VLE, which can be used for different forms of presentation. We are in alignment with Nixon and Salmon, 1996, who believe in the importance of social interaction within a learning process. Social components play a major role in the learning process. Social interaction can be realized within web-based VLEs by combining various didactical settings. For this reason use of synchronous and asynchronous communication within the mere virtual parts of the VLE must be integrated and encouraged. Interaction is supposed to be highly participative, therefore, it is a goal to motivate as many learners as possible to interact with each other and with the tutor (i.e., in form of collaborative learning). Once learners are doing this, they are taking control of their own learning process (see Nixon and Salmon, 1996) which is a superior goal of a successful learning process.

In the context of synchronized blended learning communication possibilities play a major role. Our approach emphasizes the importance to offer technological possibilities to enable electronic communication while also taking into account the possibilities and boundaries of communication through and with new media. When designing learning objects for use on the web, it is very important to stay within the limits of what seems natural communication to the learners. "Natural electronic communication" may sound like a contradiction itself but many new communication channels have evolved over the last years and have been accepted by the younger generations (e.g., short message system (SMS), e-mail, chat); see Brunn and Frank, 2002. What is seen as "natural" electronically enhanced communication differs across different target groups. If the main target group has a high cognitive, situative, and affective disposition towards computers and new technologies it is reasonable to assume that e-mail, discussion forums, and chat rooms are seen as natural communication; see Frank et al., 2002b.

2.2. Supporting Synchronized Blended Learning

Before describing how synchronized blended learning can be supported, we leap ahead and give a short overview of the structure of the learning material and how it is semantically coded. Instead of having static learning units for different contexts, the learning material is stored in small units that can be combined to larger units or courses. As shown in Figure 1, we have four levels of units or objects having different granularity levels with the following meaning:

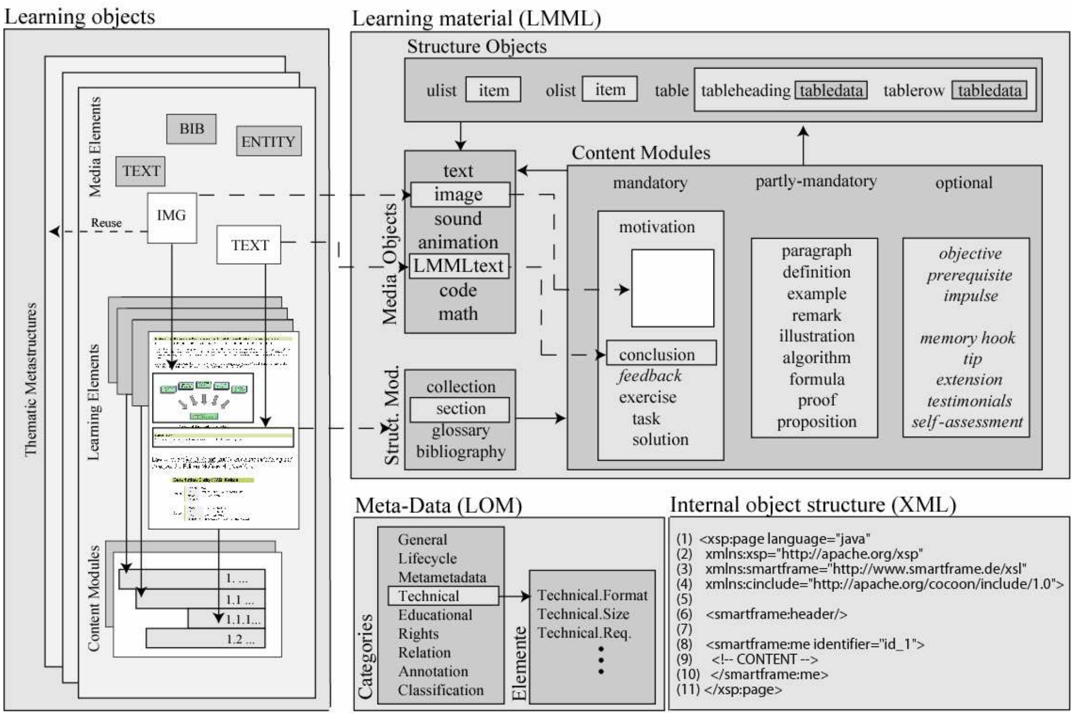


Figure 1: Structure of the learning material

Media element (ME) is a small unit without further partitions. Other objects are not included only references to other objects may be given. A media element can be text, animation, simulation, video, or audio sequence.

Learning element (LE) is a composition of objects (media elements as well as learning elements) to produce a semantic unit.

Content module (CM) is superior to learning elements and groups these to even larger units. The presentation to the learner within a VLU should be a table of content.

Thematic metastructure (TM) also called course covers a whole thematic field in the VLU.

Authors of learning material may start to construct media elements—the smallest units of "information". The granularity of these ME has to be specified with certain rules in advance which should be followed by the authors. Especially the coding of non-sensitive ME is important to allow a reusability of the same ME in different contexts.

Examples for ME are components like text fragments, tables, glossary entries, formula systems, or media objects. MEs have to be encoded in a structured and standardized format so that they are independent from the visualization.

Learning elements represent well-defined units—or containers—that encompass several media elements. Examples are sections within documents, complete pages for the presentation in a browser window, or a question for a test which might have the components text, graphics, and user interactivity. Learning elements are then combined to larger units—called content modules—which represent complete documents, courses, or the slides for a lecturer. Due to the adaptation of ME to their context within LE or CM, the learning material structure provides mechanisms to derive new ME from an existing one which can be modified to fit the context (see the German draft for a technical report by Reiners et al., 2003a).

Furthermore, each media element—as well as the hierarchically constructed learning element and course modules—is wrapped by a learning object meta-data wrapper (LOM wrapper), a certified standard to specify the features and content. LOM is a standard by the IEEE (IEEE Learning Technology Standards Committee, 2005, see also Reiners et al., 2003a, for a critical description of the standard) but there are also other standards like ARIADNE, which is based on LOM, SCORM, and IMS. For each ME different categories are used to keep information about the content itself, the lifecycle (i.e., the history of changes or its ancestors in case of inheritance from other ME), technical and educational specifications as well as descriptions of rights, relations, classifications, and annotations. The categories contain several elements, for example, to classify the ME in respect to the learning level (beginner, intermediate, or expert) or type of audience (undergraduate learner, graduate learner, or practitioner) by using an appropriate and defined vocabulary and level of detail.

Here, XML seems to be the best choice according to existing technologies because it possesses the required implementation features, can be easily transformed to other standards by using XSL-transformations (e.g., to reuse learning materials in software from other vendors), and allows the usage of standard operations like full textual search. There are several established standards to encode different types of content, the following list specifies some used in the original or slightly adapted form within our project: MathML is a standard for the representation of equations, QTI for the representation of questionnaires and multiple choice tests, and especially LMML (Learning Material Markup Language) for classifying the learning objects in items like "exercise", "theory module", "algorithm", or "multimedia objects"; see Süß, 2005 for a detailed description as well as Reiners et al., 2003a, and Reiners et al., 2003b, for a discussion in respect of combining LMML with LOM.

The suggested structure bares its advantages especially for authors (see, e.g., Reiß and Reiners, 2004) of new learning material as well as for tutors because existing learning material can be reused to derive new material. On the other side, new material can be integrated by using references to other learning objects which improves the comprehensibility for the learners. Furthermore, the learning material can be easily combined as building blocks to create slides for a lecture, handouts and scripts for classes, or documents like papers or books for publication. The following example for a general classroom scenario demonstrates how the structure of the learning material supports synchronized blended learning.

The main aspect of synchronized blended learning is the integration and (new) combination of different learning methods. The design of the learning material has to be developed with respect to fulfilling different requirements of learners. While some learners prefer to learn in a classroom listening to a course presented by a tutor, others have time-schedules or other reasons like child care or illness which do not allow them to attend the lecture in the classroom regularly. The idea behind synchronized blended learning is the creation of equal prerequisites to all learners, whereas the different forms of presenting the content have to be synchronized.

Figure 2 shows an example where the tutor presents learning elements using the visualization of slides (LE_{1S}-LE_{3S}) and gets interrupted by a question from the audience. Assuming that the tutor can answer the question by showing additional slides in form of an excursion ($LE_{Excursion1S}$, $LE_{Excursion2S}$), the classroom presentation would have been interactively changed in contrast to the planned course outline but would also be diverged from the self-paced course material covering this special classroom session. Therefore, the self-paced course has to be modified synchronically by adding special learning elements containing the textual counterparts of the slides ($LE_{Excursion1}$, $LE_{Excursion2}$). Furthermore, audio and video files of the learner asking the question as well as the answer of the tutor can be projected to the self-paced course allowing the distant learner to get the same information as the attending ones. Later on, the material, currently containing predefined learning elements without textual integration into the existing course material, can be adapted by rewriting or adding components as well as including further material from other sources like whiteboard animations. The synchronization and adaptation is not limited to classroom and self-paced course presentations but also guarantees up-to-date scripts and handouts.

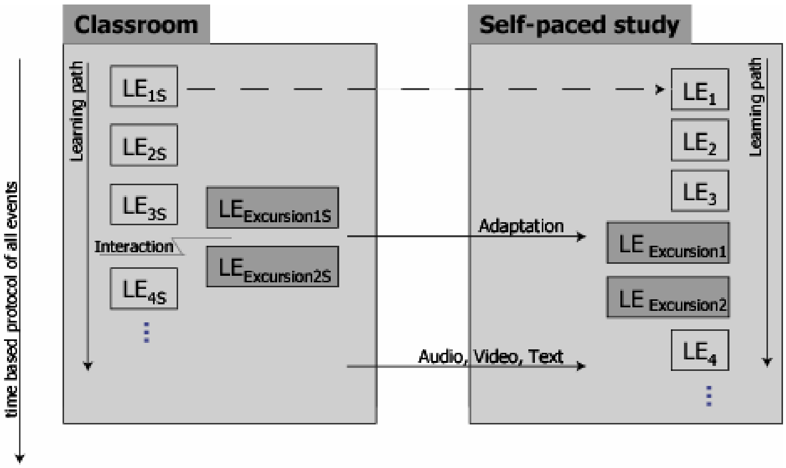


Figure 2: Synchronization process between the classroom presentation and the self-paced study

Therefore, several methods of passing the knowledge to the learner have to be given, combined, and integrated into varying didactical conceptions, as described in the following list:

Instructor-led classroom: The tutor presents the material to learners in a traditional face to face classroom. Learners can interactively communicate with each other and the tutor. To integrate this kind of presentation into a VLE, the tutor has to use the VLE in combination with an LCD-projector showing the same guidance and learning material the learner will have in later self-paced reviews. The tutor can use interactive elements like applets, exercises, or videos to design a more descriptive class; e.g., an electronic blackboard should be used for answering questions to combine digital with handwritten elements allowing a storage and later retrieval.

Instructor-led virtual classroom: Similar to the previous approach except that the tutor and the learners are not in a traditional face to face classroom but virtually connected through a network. Learners can follow the actions of a tutor within their VLE—keeping a record of the path for a later review—and asynchronously communicate over a chat or e-mail system—using voice as well as typing—as well as a whiteboard, the electronic equivalent of a traditional blackboard. Access to the whiteboard is granted if the learner needs to formulate questions or answers. Especially server-based applications ensure that every participant can demonstrate interactive components to everyone. The learners have to be online if they want to interact but are also able to download the required learning material including path descriptions for a later review.

Online mentoring: Compared to the previous two methods, the tutor is not actively involved in the learning process but is available via chat, e-mail, voice- or video-stream in case the learners have questions or need assistance. Furthermore, the tutor might be able to passively watch the learners and interfere if problems seem to be occurring.

Interactive computer-based training: The learner uses the VLE to access the learning material. This can be done either by browsing—where the learner decides the order of the material—or by using a guided tour, which can be the (dynamically created) learning material of the last given lesson of a course. This approach offers the learner the most flexibility because the learning process is time and place independent and, therefore, the best alternative for learners who have a profession and can only learn in off-work hours.

Library: As in classroom learning the learner can use literature available in digital formats. Furthermore, references to printed material are provided as well as an extensive glossary, index, and FAQs (frequently asked questions). Filters are used to limit the results to the current course of learning material. Intelligent searches are used to help the learner find adequate material.

Exercise and quizzes: The learning process has to be accompanied by exercises to both test the learners' progress and give the learner feedback. The exercises adapt the material to the learners' background. At the end of a content module quizzes are performed to grade the knowledge of the learner. Preparation for a quiz is supported by the VLE by visualizing the learning material which is relevant for the test and which is meant for further reading.

Collaborative learning is a substantial part of a blended learning approach and is, therefore, supported by the VLE in various forms. The following list describes the most important features within our approach:

User management: The VLE supports an extended user management allowing the grouping of learners and tutors. These groups can either be the participants of a course or just a small learning group including only two learners. Furthermore, the user management is used for access restrictions within the learning material but also for further functionality (e.g., access to whiteboards or chat rooms). Finally, the configuration for the visualization of the VLE is part of a user management system.

Time- and place-independent learning: In most cases, members of a group prepare the learning material by themselves and come together for further discussion and to solve problems. Therefore, the VLE has to support the scripting of the learning path as well as annotations and markings within the learning material by the learner. It can be used on- and offline by transferring all relevant information to a server, thus allowing the change of the working place whenever wanted. When working as a group, the information of all group members can be accessible for all.

Communication/Sharing of learning information: Communication is probably the most important part in collaborative learning. Therefore, the VLE supports most common forms of communication whereas a restriction to certain learners or groups can be applied. Further methods are the exchange of bookmarks, annotations to explicit text passages, or learning paths. In cases where the group itself needs further assistance, the tutor (or a person having the required knowledge) can be contacted using chat, e-mail, or in case of being online, live voice or video streams as well as instant messaging.

2.3. Example for Synchronized Blended Learning

Teaching and learning simulation requires—besides good lectures—practical demonstrations and exercises. Therefore, we initiated a student project where students were asked to identify the requirements of a discrete event-oriented simulation tool that can be used within the classroom, for exercises as well as exams. The list included standards like simple but powerful user interface, appealing graphics, and visualization of all processes but also further features which are described in more details before we demonstrate the integration into the synchronized blended learning approach; see Klie and Schalong, 2002.

One of the main features of SimTool is the possibility of being able to configure basically everything. Next to selecting the language of the interface, the degree of functionality can be changed using XML-based configuration files, also called lessons. The user interface (menu, toolbar, list of components to build simulation models, popup-menus for the components as well as items within the menus) can be adapted to the goal of a lesson; see also Figure 3 part a) for a screenshot. For example, in the beginning, it might not be useful to give the learner all the functionality of the application but limit it to certain components (e.g., source, working station, and destination) as well as controls to start the simulation. Other components (e.g., parallel station) are blocked and can not be used or are not visible at all. Another feature is the script-oriented event-action mechanism by which the designer of the lesson can guide the learner. The XML-based script language can be used to do everything that the user can do with the user interface. Therefore, a lesson could start with a script building parts of a simulation model and asking the learner to finish it by inserting the missing components. The event-action mechanism allows reacting on events (e.g., inserting a new component, changing the attributes of a component, or fulfilling a certain outcome of production of goods). For example, the learner is asked to insert the missing components while events could be used to show messages whenever the learner is doing the right actions but also prevent—including the display of a message—that existing parts of the model are changed.

According to the idea of synchronized blended learning SimTool can be interactively used within the classroom as well as in the self-paced study. Subject is the design of a small production line with three products which should be combined for the final product. Starting with a prepared lesson, the lecturer inserts a source for the first good while giving oral explanations about the source and its attributes. Inserting the second source triggers an event which adds the other sources as well as the corresponding attributes. Therefore, the lecturer does not have to care about repeating the same procedure three times. Afterwards, the model is finished involving the audience by asking for suggestions; see Figure 3 part a) and b). SimTool protocols every step in a new script—and here is the synchronized blended learning concept—being directly transferred to the self-paced study together with the audio comments from the lecturer. Within the VLE, the SimTool is used in view mode where no interactions besides start, pause, jump, and stop are possible allowing to follow the classroom lecture without missing material; see Figure 3 part c).

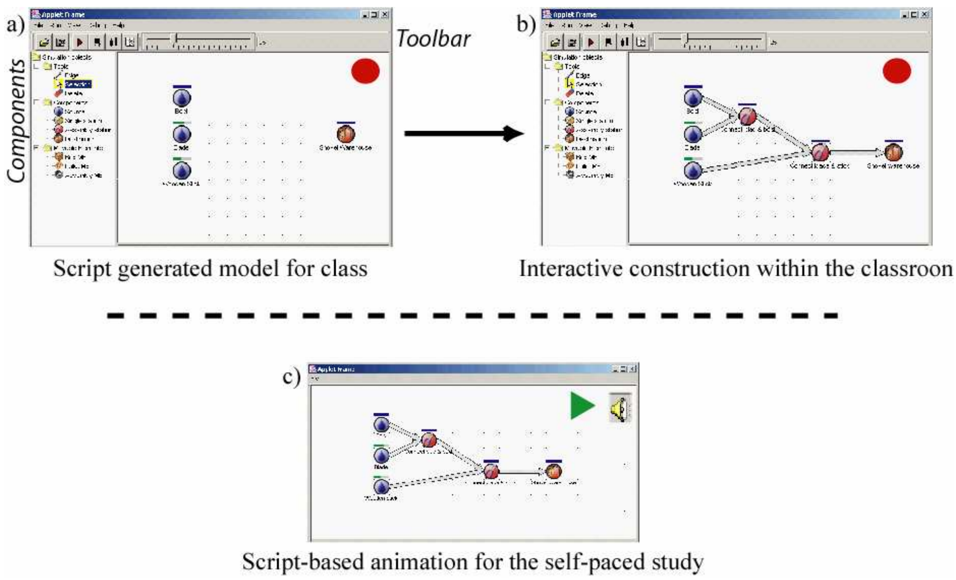


Figure 3: SimTool within a synchronized blended learning concept

3. Supportive Architecture for Synchronized Blended Learning

Nowadays, almost every university is developing a VLE. Nevertheless, we had to define a concept and design a system architecture that allows an integration of the synchronized blended learning approach besides other components to support the virtual learning. Furthermore, we ask for other characteristics like the usage of up-to-date technology, current internationally accepted standards, and specifications by committees like IEEE or W3C, with prospect of future validity as well as usage of software that is not based on proprietary components, is compatible with different operating systems as well as free of copyrights. Because of these criteria we developed a prototype for a VLE, called SMARTFRAME, in which the synchronized blended learning concept will be integrated. The main factor within the design of the system architecture is the modularity. Therefore, all components are independent from each other (learning material, transformation of the learning material for presentation based on the user configurations, and the VLE itself) using a minimal interface for communication.

The XML-based learning material is converted by special software (e.g., Cocoon (http://xml.apache.org/cocoon) in connection with the Tomcat-Servlet-Engine (http://jakarta.apache.org/tomcat/ index.html)) into the requested format which depends on the kind of usage, the settings of the learner—by using an adaptation mechanism for the style files—as well as the medium for the presentation. For slides presented within the browser it might be necessary to use a larger font while the same material can be transformed to a PDF-document with six slides on one page for a handout to the learners. Our approach for a VLE provides further innovative components like SmartBars to aggregate information about the content in a compressed presentation, Hyperbolic Tress to display navigational information as well as a history, different forms of communication and a high degree of freedom for configuration. For a detailed description, we refer to Reiners et al., 2002, Reiners et al., 2003a, as well as Reiners et al., 2003b.

4. Conclusion

Implementation and further development of technological innovations are crucial when designing a VLE. Even though the main priority is still the usage and further development of state of the art technologies we also place a major focus on pedagogical aspects. The pedagogical context, however, has been neglected by most related projects for too long. If a VLE is not built up according to pedagogical recommendations and guidelines, the success, acceptance, and the use of the VLE are questionable. Modern technology can only be used successfully to support learning if it is integrated into a complete approach which integrates technology, communication, and a social setting for learning. Our approach demonstrates how synchronized blended learning can be realized using a system architecture as well as design of the learning material described in this paper. The main target group and their wishes and needs must be evaluated to provide the necessary adaptation of the VLE to the requirements of the learners. We think that our approach of synchronized blended learning is one way of combining innovative technology with proven successful methodologies of teaching. Offering learners the possibility to choose between different types of learning processes which are adaptable to their needs, prerequisites, time schedule, and preferences will extend the acceptance of the VLE and increase the motivation and success of the learning experience in general.

Literature

Bonk, C. (2005). Handbook of Blended Learning. To appear in December 2005.

Brunn, H., and Frank, C. (2002). Online communication: A success factor for blended learning. In Driscol, M., & Reeves, T. (Eds.), *Proceedings of Elearn, World Conference on E-learning in Corporate, Government, Healthcare, & Higher Education*, pp. 367—373. Montreal, Canada.

Frank, C., Kassanke, S., and Suhl, L. (2002a). Meeting students expectations and realizing pedagogical goals within the development of a virtual learning environment. In Driscol, M., & Reeves, T. (Eds.), *Proceedings of Elearn, World Conference on Elearning in Corporate, Government, Healthcare, & Higher Education*, pp. 1124—1129. Montreal, Canada.

Frank, C., Suhl, L., and Tan, L. (2002b). Technological and pedagogical aspects of developing VORMS, a virtual course on operations research/management science. In *Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002)*, Volume #100029-03-LS-161, Millet Alberta, pp. 1—6. ICSC-Naiso Academic Press.

IEEE Learning Technology Standards Committee (2005). Standard for Information Technology: Education and training systems – learning objects and metadata. http://ltsc.ieee.org/wg12. April, date of last check.

Klie, T., and Schalong, C. (2002). *A Script-driven Simulation Tool as an Improvement to the Education in Web-based Virtual Learning Environments*. Diploma Thesis, University of Technology Braunschweig.

Nixon, T., and Salmon, G. (1996). Computer-mediated learning and its potential. In Mills, R., and Tait, A. (Eds.), *Supporting the Learner in Open and Distance Learning*, pp. 88—100. London: Pitman Publishing.

Reiß, D., and Reiners, T. (2004). Supporting the authoring process of hierarchical structured learning material. In Proceedings of Ed-Media 2004. World Conference on Educational Multimedia, Hypermedia & Telecommunications, Lugano, Switzerland 2004, pp. 917–924.

Reiners, T., Reiß, D., and Voß, S. (2002). Using hyperbolic trees and SmartBars within virtual learning environment concepts. In *Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002)*, Volume #100029-03-TR-026, Millet Alberta, pp. 1—7. ICSC-Naiso Academic Press.

Reiners, T., Reiß, D., and Voß, S. (2003a). XML-basierte Kodierung von Lernobjekten. Technical report, University of Technology Braunschweig.

Reiners, T., Reiß, D., Sassen, I., and Voß, S. (2003b). Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: Von der Technik zur Anwendung. Technical report. University of Technology Braunschweig.

Reiners, T., Reiß, D., Schulze, H., and Voß, S. (2003c). SMARTFRAME: An integrated environment for xml-coded learning material. In Uhr, W., Esswein, W., and Schoop, E. (Eds.), Wirtschaftsinformatik 2003/Band I, Heidelberg, pp. 613–632. Physica

Süß, C. (2005). Learning Material Markup Language. http://www.lmml.de. April, date of last check.

Wilson, D., and Smilanich, E.M. (2005). The Other Blended Learning: A Classroom-centered Approach. To appear in June 2005.

Supporting the Authoring Process of Hierarchical Structured Learning Material

Dirk Reiß
Technical University Braunschweig, D-38106 Braunschweig, Germany
d.reiss@tu-bs.de

Torsten Reiners
Technical University Braunschweig, Spielmannstr. 8, D-38106 Braunschweig, Germany
t.reiners@tu-bs.de

Abstract: The quality of virtual learning environments is mainly depending on the quality of the presented learning material. This encompasses the scientific quality of the content as well as the surplus the learner experiences when using the virtual learning environment in contrast to conservative learning methods. As our virtual learning environment offers a highly configurable platform for online learning content and the option to present the most adequate learning element according to a specific learners' profile, the authoring process, including the generation of the content itself as well as the description of each learning object with meta-data, is way more complex than the generation of static HTML-pages. To be able to provide this service without lacking the quality and quantity of the content, we assist the author by developing a set of authoring tools for generating the content in an integrated environment which will be described in this paper. The authoring tool is integrated in the virtual learning environment SMARTFRAME.

1. Introduction

In the last years, almost every institution being involved in the educational sector initiated e-learning projects. That is, they either integrated existing virtual learning environments in their curriculum or developed new technologies to improve the quality in respect of technical as well as didactical presentation. Nevertheless, e-learning never really gained a true breakthrough as it was always predicted (Heise Online 2003). The number of exclusively virtual courses at, e.g., universities, is still small. There are manifold arguments that can be found in literature; in terms of keeping the focus of the paper, we shall limit ourselves to the following ones:

- **Limited output formats:** Virtual learning courses require sophisticated content in such a way that a learner is attracted and motivated to prefer virtual learning over traditional offers. Whenever students are asked about the preferred media of the learning material, the result is "pdf or similar types" so it can be printed and read offline at every place. Even though other formats might not be required to get through to more learners, the disadvantages result mainly from static transformations before requesting the document. In terms of keeping the learning material up-to-date, the conversion to the requested output format should consider the users' specifications.
- **Static learning material:** The learners' characteristic should specify the presented learning material. Currently, most virtual learning environments follow a static model where the material is presented as given by the author with respect to structure and content. That is, the material is encoded in HTML-files that are given on a linear learning path.
- **Missing support of learning material specific authoring tools**: On the one hand, the virtual media has major advances as being adaptive to the current learning environment and learner, and being interactive. On the other hand, the authoring process exceeds the writing of static text by far. Therefore, the first big step to a successful virtual learning environment is content that uses the advances of virtuality and, therewith, advanced authoring tools are required.

The kind of authoring tool depends on the kind of learning material. In case of environments only supporting static HTML-pages, a preferred editor among a large amount of available software tools can be chosen. In case that the learning material should support modularity, different versions of objects with respect to language and content representation as well as classification of the content by semantical concepts, a sophisticated authoring tool is needed. The latter one classifies the content in specific groups allowing the learner to directly conceive the document structure. Without anticipating Section 2, a typical structure for learning units might be a motivational part, a textual or pictorial block, and a conclusion. The modularization allows a user-specific presentation according to the configuration as well as difficulty level by selecting required objects. A learner of a beginner level might require the learning objects *motivation*, *objective*, and *prerequisite* to follow the subject, whereas the expert might only want the main definitions or theses.

The authoring tool presented in this paper is integrated in the virtual learning environment SMARTFRAME (Smart Technology for Research and Modern Education; see SMARTFRAME 2003). The content is stored in hierarchically structured semantic units, so-called learning objects, which are directly composed and transformed according to user-specific configurations and meta-data descriptions to the requested output format. Depending on the learning style and goal, the learner can select the form and density of the presentation. That is, if the learner repeats course units to memorize the most important facts, the learning material can be (automatically) condensed to the most relevant parts regarding the learners' needs by hiding "non-relevant" and optional learning objects. This scenario is shown on the left-hand side of Fig. 1 (graphical presentation is left out), whereas the right-hand side shows the adaptation according to the difficulty level of the learner, where either an interactive (difficulty equals low) or the mathematical equation (difficulty equals medium) is shown.

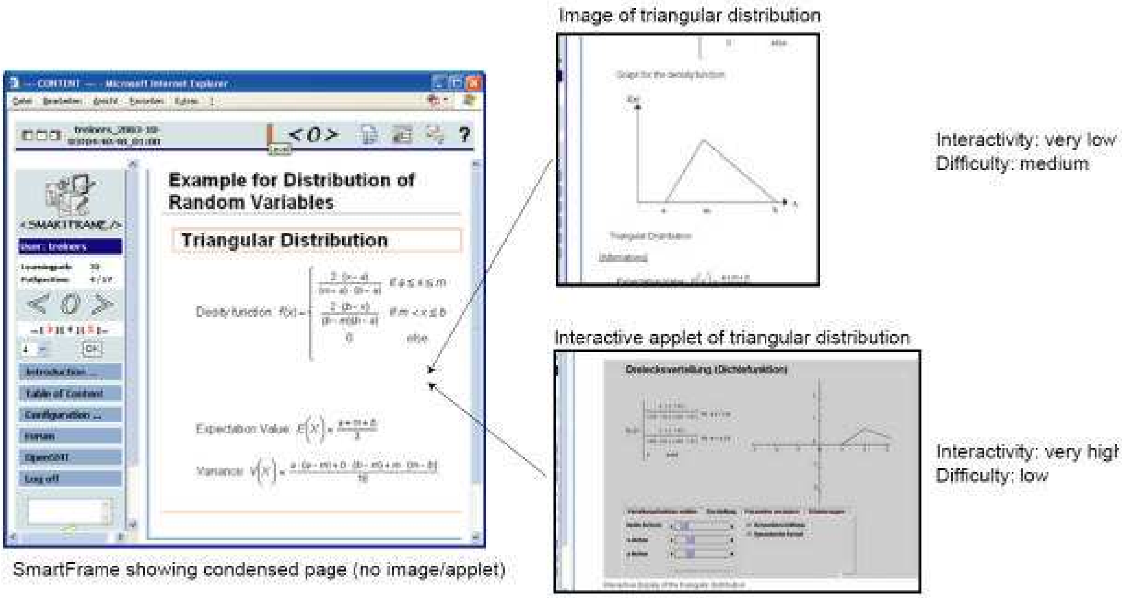


Figure 1: Example of adaptivity within SMARTFRAME

The paper is organized as follows: In Section 2, we describe the structure, encoding, and characterization of the learning material as a basis for Section 3, the description of the authoring tool used to create the learning objects and their relation. Section 4 covers the second part of the authoring tool, which is used to enter the meta-data of the learning object following the LOM-specification. The final chapter comprises the conclusion as well as a short outlook on the future research.

2. Structure of Learning Material

The structure and internal presentation of the learning material restricts the later options for presentation, reusability, manageability, and flexibility of being used in different contexts. Following the idea of other projects (see, e.g., Advanced Distributed Learning 2003, ARIADNE Foundation 2003, and OR-World 2003) we apply a hierarchical structure (aggregation level) by distinguishing four different types of learning objects: (1) media elements are small units without further partitioning, i.e., other objects are (besides few exceptions) only referenced, (2) learning elements are compositions of media as well as learning elements to produce semantic units, e.g., web-pages or sections in printed documents, (3) content modules combine subordinated learning objects to, e.g., courses or seminars, and (4) thematic metastructures describe a thematic field, e.g., the scientific work of a department or the course set to obtain a degree.

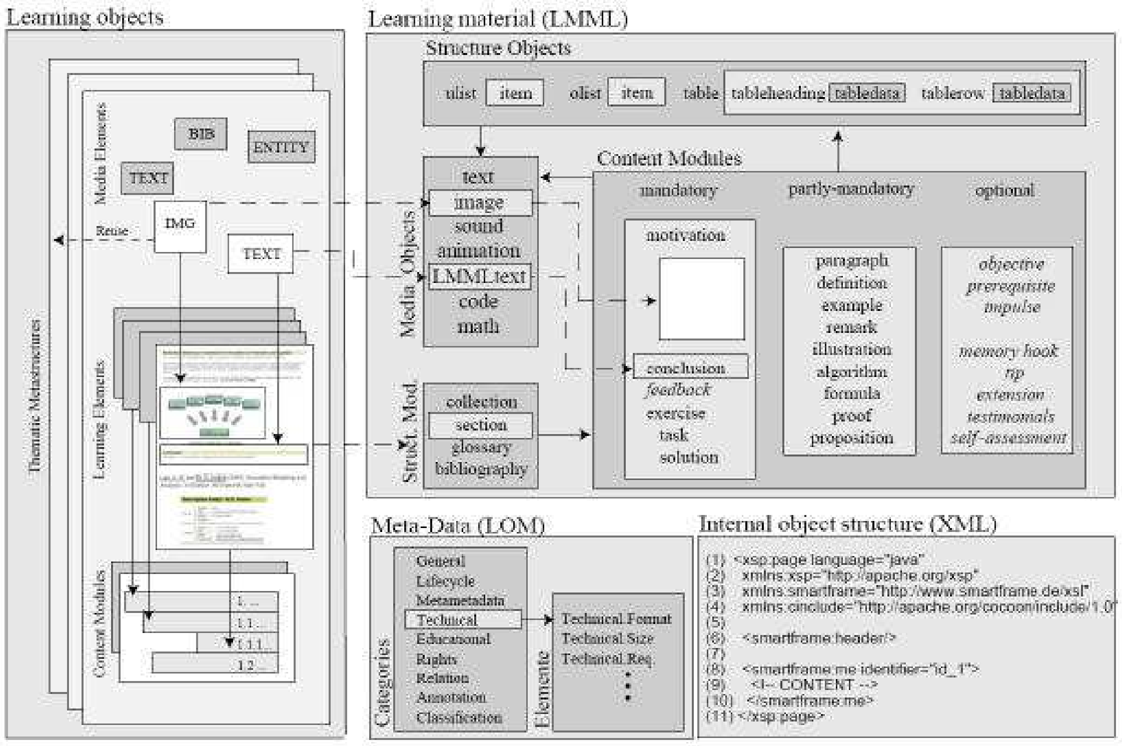


Figure 2: Structure of the learningmaterial

Furthermore, each learning object is split in the XML-coded content part and its describing meta-data following the slightly modified LOM-specification; see (IEEE Learning Technology Standards Committee 2003) as well as (Reiners et al. 2003a) for a list of changes). Meta-data allows reusability, classification with respect to several criteria, and recovery in a large pool of objects. Therefore, each learning object is specified by meta-data as shown in Fig. 2 on the lower right-hand side. In our approach we decided to use LOM, which basically arose from the other standards and is on the verge of being published as an IEEE-standard.

An XML-based learning object coding was chosen due to its popularity, future perspective, and readability but also from a technological point of view, i.e. simple transformation to various (output) formats by using XSLT (eXtensible Stylesheet Language Transformations; see (W3 Consortium 2003b)), processing by freely available tools, and standard operations like full textual search. Furthermore, we based our coding on LMML (see Süß 2003), whereas several components had to be adapted, especially in terms of refining the underlying didactical model as well as combining LMML with the meta-data specification LOM; see (Reiners et al. 2003b). LMML introduces further semantical classification of the learning material by distinguishing four categories. Fig. 2 shows on the upper right-hand side structure modules (used to enclose and structure the content), content modules (to compose the learning material), structure objects (to build tables and lists), and media objects to include and classify the content itself.

The learning objects are encoded as XSP-files (eXtensible Server Pages) using the structure as shown in Fig.2. Lines (1)-(4) define the type of the file as well as the used namespaces, line (6) will force the inclusion of code, e.g., to resolve links to other learning objects, and line (8) identifies the learning object by its granularity level (*smartframe:me* corresponds to a media element) and a unique identifier (*id_1*). The specific tag <smartframe:link …> allows the composition of objects in different ways; see (Reiners et al. 2003b). The XML-based learning material is transformed to, e.g., traditional printed scripts or web-pages. The transformation process of XML- and, especially, XSP-files is done with Cocoon (Apache Software Foundation 2003b), a specialized software running as a webapplication under the Tomcat Servlet Engine (Apache Software Foundation 2003c).

3. Authoring Tool for Learning Objects

3.1 Overview

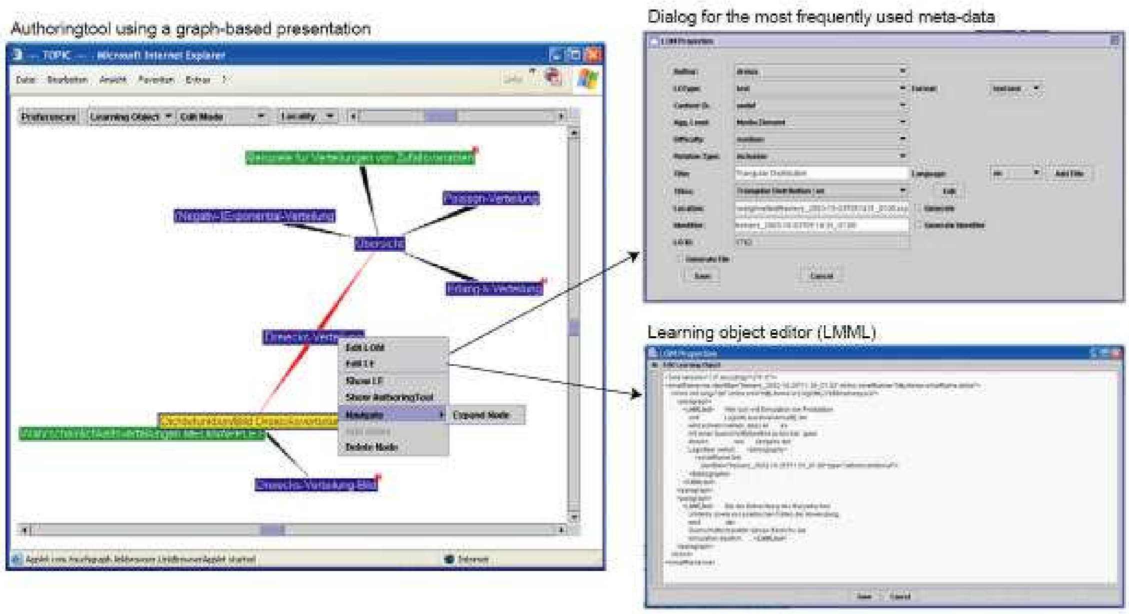


Figure 3: Screenshot of the authoring tool

As described before, the learning material within the SMART FRAME virtual learning environment is divided in small semantic learning objects, each of which is described using LOM-meta-data. Due to the structural affinity to a mathematical graph, we chose this presentation form as a basis for the authoring tool. A screenshot of the application is shown in Fig 3.

Within a graph representation, the particular learning objects are visualized as nodes and the interdependencies between the learning objects are represented as edges, e.g., directed edges for *is part of* relations or undirected edges for *is format of* relations. The nodes are labeled either with the name of the learning object (retrieved from the LOM-meta-data) or with the identifier which is used to name the learning object internally. As a powerful and user-friendly basis for the authoring tool, we chose the TouchGraph LinkBrowser (see Touchgraph 2003), which is a Java-application or a Java-applet respectively distributed under an open-source license. Originally, the TouchGraph application displayed a structure described by a XML document. According to our needs, we customized the interface in a way that the structural information is retrieved from a database management system containing the

LOM-meta-data. A matter of particular interest here is the information describing the relational dependency, as it determines the connections between the learning objects and thus their arrangement. The different aggregation levels (see Section 2) are displayed in different colors and, therefore, an easy differentiation is possible.

The user is able to select a single node (by clicking on it), drag a node including the attached ones on the screen (by holding down the mouse button and moving it around), zoom in and out for overview purposes, and collapse and expand child nodes, i.e., one can hide nodes attached to a specific one. The number at the corner of the node indicates the number of nodes that are currently collapsed, i.e. are not shown. Furthermore, the user may rotate the complete graph structure and change the locality level (i.e., the number of node levels that are visible starting from a selected node) using a slider at the upper border of the applet. Basically, there are two modes in which the authoring tool can be used: an *editing mode* where the user may add, delete, and edit nodes, i.e., the learning objects themselves, and a *navigation mode* where editing capabilities are disabled and only the browsing functionality is available to the user. In the editing mode, the author is able to create new learning objects or delete them, change the most relevant parts of the meta-data and create and delete relations between the particular learning objects. As the authoring tool needs to have editing capabilities for the learning objects themselves, several dialogs are available which include a mask for editing the basic meta-data parameters, a dialog for choosing a file (e.g., an image file) and an editing window for modifying the XML learning object itself. The navigation mode can also be used to display a certain learning path and indicate, e.g., the learners' progress within a certain context. For a convenient access to the learning object within the virtual learning environment, the author (or the user in browsing mode, respectively) can double-click on a node to receive a rendered HTML-presentation. Alternatively, being in the editing mode, the author may select the corresponding entry from the context menu.

3.2 Editing the learning material

In order to create a new learning object, the user may use a context menu (which pops up by pressing the right mouse button) or, if the new learning object has any relation (such as an *is part of* relation) to an already existing learning object, the author moves the mouse while holding the left mouse button from a selected node. Creating a new learning object the latter way, information about the appropriate relation is automatically inserted in the LOM database together with appropriate default values. Furthermore, a dialog allows the specification of further meta-data as well as generating a skeleton learning object (checkbox in the left lower corner), which depends on the object type. In case of binary files, e.g. an image, the author is asked to select a file on the local hard drive such that the new learning object can be uploaded to the server. For XML-encoded objects an editor with the skeletal structure is opened. In this window, the author is able to edit the XML in a habitual way assisted by a context-sensitive editing functionality. This functionality is available by a right mouse click within the editing text field and offers e.g. possible child elements at the given position in the XML document or possible attributes of the actual element. Information about the document structure is retrieved from a given XML Schema and can be customized by editing the schema-document. The written object is stored on the server using the technology as described in Section 3.3.

Both the LOM-meta-data and the learning object itself can also be subsequently modified. The author chooses the corresponding entry from the context menu of the desired node in the graph representation of the learning objects. The appropriate data (the LOM-meta-data or the learning object itself) are fetched and can be altered and stored again. If new relations between two learning objects need to be inserted, the author can connect two nodes by dragging (with the left mouse button hold down) from one node to the other, and choosing the type of the relation.

A frequently demanded task within the authoring process is the import of slides from lectures into the virtual learning environment; here we use the representation in form of images. As our concept of learning material demands the encapsulation of the image itself in another learning object (e.g. for giving the image an appropriate title, see also (Reiners et al. 2003c)), the author has to be assisted. Therefore, the author selects a set of images from the local drives, provides required meta-data, and starts the generation process where all slides with the corresponding learning objects are inserted. Furthermore, a new learning path containing the slides is created and made available within the virtual learning environment. That way, the author is able to insert the slides of the complete lecture which than can be iteratively browsed by the learner.

As we provide a special structure for learning paths within SMARTFRAME—not being in the scope of this paper—the author is able to edit the learning paths using a special input mask providing functionality to filter the different levels of a single learning path, add new learning paths as well as creating intersection and sub-learning paths.

Currently, an edit modus for learning paths using the hyperbolic graph structure is developed and will be available in the next release of SMARTFRAME.

3.3 Technical architecture

Technically, the authoring tool is a Java applet which can be loaded using a web browser. As the applet acts as authoring tool as well as a tool for browsing the structure of the learning material, there are several parameters which affect e.g. the available functionality and the parameters regarding some preferences like the author using this application or the learning path to be displayed. The applet retrieves meta-data information from a MySQL database using the Java DataBase Connectivity (JDBC). As there needs to be a way to access the XML learning objects on the server directly, we chose the Simple Object Access Protocol (SOAP) (see W3 Consortium 2003a for an overview about the SOAP protocol) to transmit the data from the server to our client-applet and back. As SOAP natively supports messages (in the sense of the data transferred) in XML format, it perfectly fulfils our needs. On the serverside, we implemented a web service running on an Apache Axis server (see Apache Software Foundation 2003a for further information). The XML document itself is wrapped in a SOAP envelope, send to the server, completed with the necessary technical parts of the document (e.g., cocoon-internal tags) and put there in the correct location. From the server, the document is transferred in the same way, likewise simplified by the technical parts of the document Binary data, such as images or video files, is encoded in the SOAP message as so called attachment and decoded at the server side.

4. Authoring Tool for Meta-data

The previous section described an authoring tool for the learning objects, whereas the meta-data edit functionality was limited to the most important elements while setting the other ones to default values. Therefore, an additional HTML-based tool for an assisted editing of the meta-data was created.

The meta-data specification LOM does not define the storage format, but XML seems to be the ideal format as it is structured, readable, and transferable into other formats by XSLT. Furthermore, the structure can easily be specified using an XML-schema. Unfortunately, an increasing number of learning objects results in the same number of meta-data-files which have to be stored and searched whenever objects are requested with certain attributes. Therefore, we chose to map the XML-information to corresponding tables in a relational database. Search processes based on characteristics of the learning objects can efficiently send SQL-requests to the database. That is, the learner or author is looking for specific learning objects or, during the adaptivity process, alternatives have to be determined.

The authoring tool for the meta-data is browser based. After extracting the meta-data for the object from the database to an XML-file, the file is read together with its schema, which specifies, e.g., how often elements have to be included, what type they are, and where in the hierarchy they occur (see extract in the top-right corner of Fig. 4). The author gets an overview of the elements on the current level as well as min/max-values and a list of sub-elements (elements that are required but not entered yet are underlined).

Furthermore, a second XML-file specifies additional support for the author. For each XML-tag (defined by its XPath-adress), default values or actions can be set. As shown in the lower-right example, the vocabulary source of the interactivity level is set to LOMv1.0, whereas the corresponding value of the vocabulary entry can be set to either one of the specified values from very low to very high (being displayed to the author in a drop-down selection box as shown in Fig. 4 on the left side). Using this mechanism we can also provide an object related setting of the meta-data, insertion of default values for most of the values, and, therewith, support the author in one of the most time-consuming and probably most disliked process. Based on author specific settings defining default values for the current editing process, most meta-data values are set during the creation of the learning objects as described in Section 3. Together with object specific rules, the author only has to deal with insertion of elements like key words, context, or description.

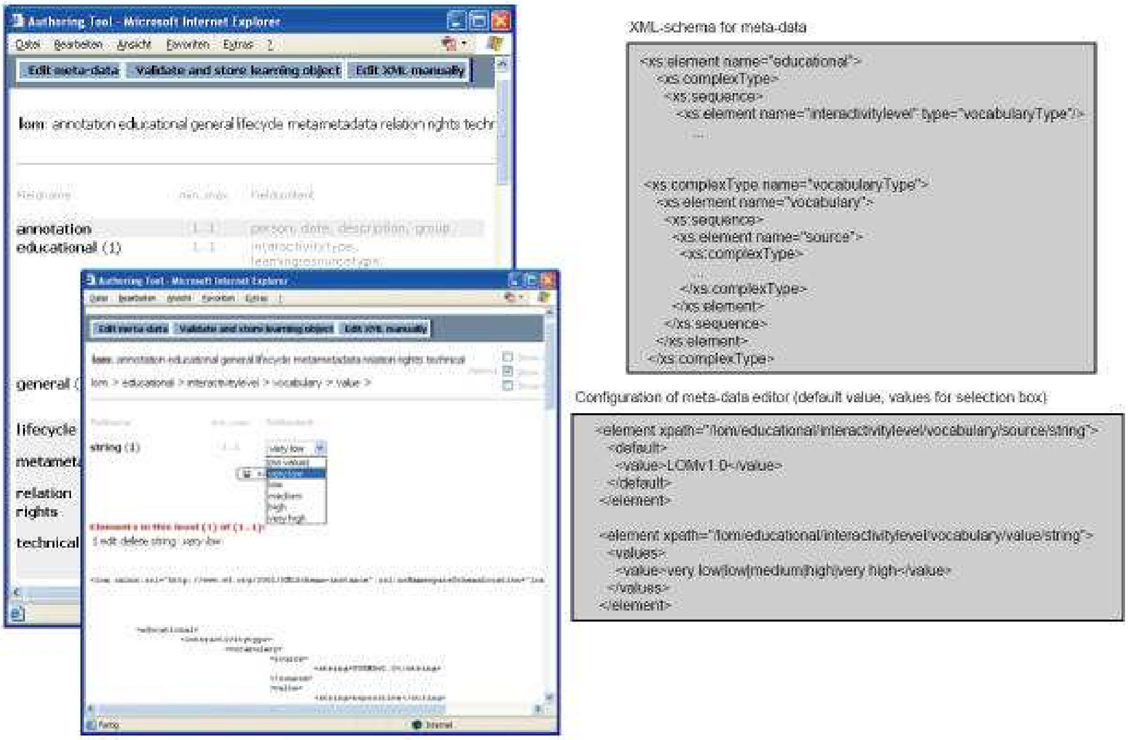


Figure 4: Screenshot of the meta-data authoring tool and extracts of its configuration files

Conclusion

Today, most virtual learning environments are based on static HTML-pages. This is completely understandable due to 1) the amount of time that has to be invested in creating highly modular learning material with each object being defined with meta-data and 2) the lack of authoring tools supporting the editing process. In this contribution, we demonstrated an approach, that maps the relation of objects using a hyperbolic graph, simplifies the insertion of new learning objects by allowing a direct extension of the graph, hiding internal structures of the encoding of the object, and supporting the meta-data editing process.

SMARTFRAME, the virtual learning environment including the authoring tool, is an ongoing research project. Therefore, several further features are currently in development and will be implemented during the remaining project duration. That is, integration of the IMS meta-data specification, further support of the object editing process like handling of different language versions, and import of various document formats including its transformation into our learning object structure.

Acknowledgements

This work was supported by grants from the BMBF (Bundesministerium für Bildung und Forschung 2003) in the program "Neue Medien in der Bildung" (01NM094D).

References

Advanced Distributed Learning (2003). *SCORM*. http://www.adlnet.org/index.cfm?fuseaction=scormabt. Date of last check 2003-12-19.

Apache Software Foundation (2003a). *Apache Axis*. December, 2003. http://ws.apache.org/axis. Date of last check 2003-12-19.

Apache Software Foundation (2003b). The Apache *Cocoon Project*. http://xml.apache.org/cocoon. Date of last check 2003-12-19.

Apache Software Foundation (2003c). *The Jarkate Site – Jakarta Tomcat*. http://jakarta.apache.org/tomcat. Date of last check 2003-12-19.

ARIADNE Foundation (2003). *ARIADNE Project Home Page*. http://ariadne.unil.ch. Date of last check 2003-12.19.

Bundesministerium für Bildung und Forschung (2003). *Bundesministerium für Bildung und Forschung bmbf.* http://www.bmbf.de. Date of last check 2003-12-19.

Heise Online (2003). Zukunft *der digitalen Lehre noch ungewiss*. http://www.heise.de/newsticker/data/anm-29.09.03-001. Date of the last check 2003-12-19.

IEEE Learning Technology Standards Committee (2003). *Standard for information technology: Education and training systems – learning objects and metadata*. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf. Date of last check 2003-12-19.

OR-World (2003). *OR-World*. http://www.or-world.com. Date of last check 2003-12-19.

Reiners, T. & Reiß, D. & Voß, S. (2003a). *XML-basierte Kodierung von Lernobjekten*. Technical Report, University of Technology Braunschweig.

Reiners, T. & Reiß, D. & Voß, S. (2003b). *Lernerspezifische Komposition und Visualisierung von Lernmaterialien*. Working Paper, University of Hamburg.

Reiners,T. & Reiß. D, & Schulze, H. & Voß, S. (2003c). *SMARTFRAME: An integrated environment for XML-coded learning material*. In: W. Uhr, W. Esswein, E. Schoop (Eds.) *Wirtschaftsinformatik 2003/Band I*, Physica, Heidelberg , 613-632.

SMART FRAME (2003). http://www.smartframe.de. Date of last check 2003-12-19.

Süß, C. (2003). *Learning Material Markup Language*. http://www.lmml.de. Date of last check 2003-12-19.

Touchgraph (2003). *Touchgraph LinkBrowser*. December, 2003. http://www.touchgraph.com. Date of last check.

W3 Consortium (2003a). *SOAP Version 1.2 Part 0: Primer*. December, 2003. http://www.w3.org/TR/2003/REC-soap12-part0-20030624. Date of last check 2003-12-19.

W3 Consortium (2003b). *XSL Transformations*. http://www.w3.org/TR/xslt. Date of last check 2003-12-19.

Instructional Design and Implementation of Interactive Learning Tools

Imke Sassen
Institute of Information Systems
University of Hamburg, Germany
sassen@econ.uni-hamburg.de

Torsten Reiners
Institute of Information Systems
University of Hamburg, Germany
t.reiners@tu-bs.de

Björn Paschilk
Institute of Information Systems
University of Hamburg, Germany
2paschilk@informatik.uni-hamburg.de

Stefan Voß
Institute of Information Systems
University of Hamburg, Germany
stefan.voss@uni-hamburg.de

Abstract: The usage of interactive learning methods can improve higher education by building modelling and problem solving skills. In this paper we focus on our experience in developing and implementing complex learning tools for teaching and learning in the field of Operations Research & Management Science (OR/MS). Here we want to show how teaching at university level in quantitative courses may be extended and enhanced. For this purpose we would like to present a field report including our pedagogical goals and instructional approach as well as the technological realisation of SimTool, a simulation tool especially designed for education.

Introduction

Only a close co-operation of media-didactical and program-technical developers ensures high quality and durable content. There are strong distinctions between the development of educational applets or learning environments and the development of "traditional" software in several aspects. In contrast with traditional software development, people with very different skills are involved in the development process, because there is a need of special technological and pedagogical resources like a consideration of aesthetic or cognitive aspects (see Koch 1999). In this paper we present some of the current advancements of SimTool. It is a simulation tool, specially designed for education within the classroom and the self-paced study. It demonstrates main principles as well as internal processes. SimTool is a source forge project and also part of the results of our research within the project "Virtual Operations Research/Management Science" – "VORMS". The main goal of this project is the implementation of virtual courses in OR/MS.

SimTool: Instructional & Technological Aspects

Simulation tools are used to transfer the reality into a (simple) model, which can be used for experiments like introducing different machines to produce a good, or evaluating the influence of defects (see, e.g., Eisenberg et al., 2003). There are several (commercial) simulation tools, but most of these tools are not designed to be used within e-learning. That is, special visualisations of internal processes, explanations, or adaptations to the student needs are not integrated. SimTool is a component oriented simulation tool, based on a discrete event oriented simulation engine.

It includes special features like feedback to performed actions, guidance by the simulation tool while solving a problem, or adaptation of the difficulty level regarding the student's background knowledge. The design allows a high degree of configurability defined in so-called lessons, which allows lesson specific configuration. Simulation objects can be turned on or off, depending on the aim of the lesson. The functionality as well as the interactivity might be restricted with respect to required options to solve the task. The main focus during the development of the prototype was put on modularity, extensibility and reusability (see Klie & Schalong, 2003). Furthermore, basic pedagogical support such as a good visualisation of internal processes and interactive components was designed.

The user should be introduced to the "simulation" subject in a stepwise fashion. The content is represented in a manner that is easy to understand. Furthermore, the learner should be motivated by the system using interactive components.

An additional goal was the integration into an existing virtual learning environment (see Reiners et al., 2003).

The tool has an easy to use, self-explanatory graphical user interface (GUI). It helps visualising simulation principles, which are usually hidden by simulation tools. Learners have the possibility to easily create basic simulation models. Furthermore, a demonstration facility is implemented. Therefore, a script language is used, which can be used by teachers for creating presentations for demonstration purposes.

The simulation tool also allows interaction of the learners with the system. The initial simulation model is built by executing a script in the beginning, whereas, subsequent to the building phase, the student has to solve the given problem formulation by interacting with the simulation tool.
Thus, the system is able to react to the learner's actions and informs the learner about possible mistakes. The usage of a script language enables course designers to implement tutorials including further exercises for the learner. The system is able to check whether the learners have solved the task. The simulation will give useful feedback depending on the difficulty level and defined events. Therefore, an exception handling is implemented. The initial simulation model is built by executing a script in the beginning, whereas, subsequent to the building phase, the student has to solve the given problem formulation by interacting with the simulation tool.

During the learning phase with SimTool the student is able to develop modeling and problem solving skills, based on one's own experiences. The student may be motivated to go through further simulation trials in order to improve solution procedures (the final report shows the differences between various runs). Furthermore, there is a so-called recording-modus integrated in SimTool. If recording is activated, all actions of building a simulation model are recorded step by step.

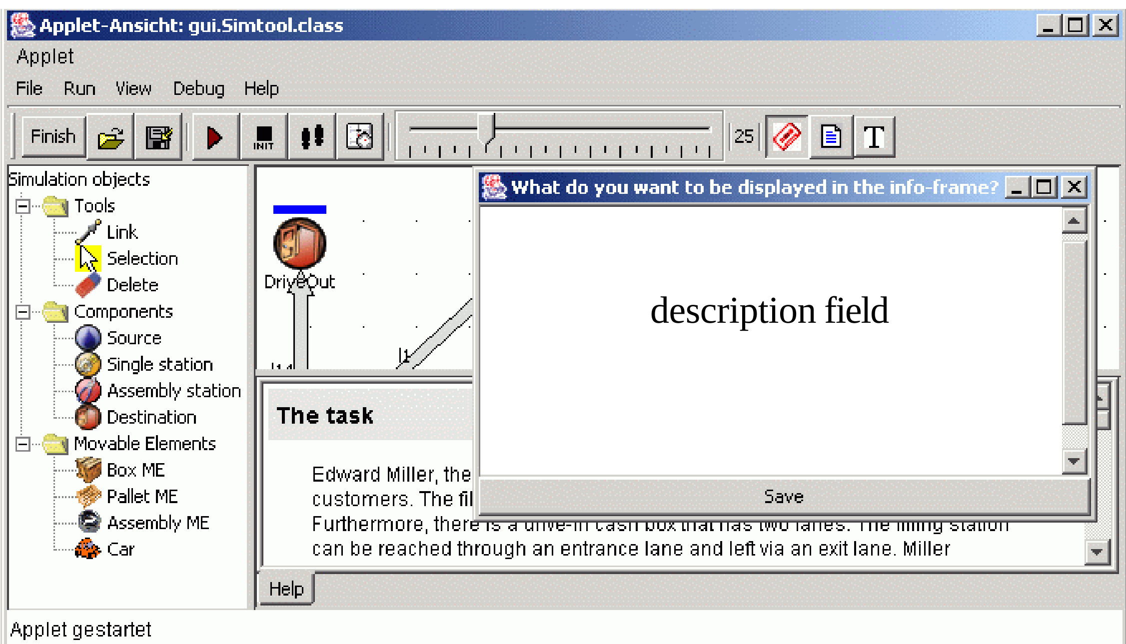


Figure 1: SimTool "Recording"

There is also an additional possibility to communicate to the learners by giving short text information in a small pop-up window or to show more complex information or tasks in the bottom frame of SimTool under conditions of chronological synchronism.

When loading a recorded simulation model, all recorded actions and information will be displayed in "real-time", which means in the order they were recorded and with the time intervals used at recording. This feature supports the learner in understanding the process of building a simulation model: there is an analogy to the teacher in the classroom creating and explaining the model. Furthermore, the recording process can either be applied on new models or be started anytime during the modeling process.

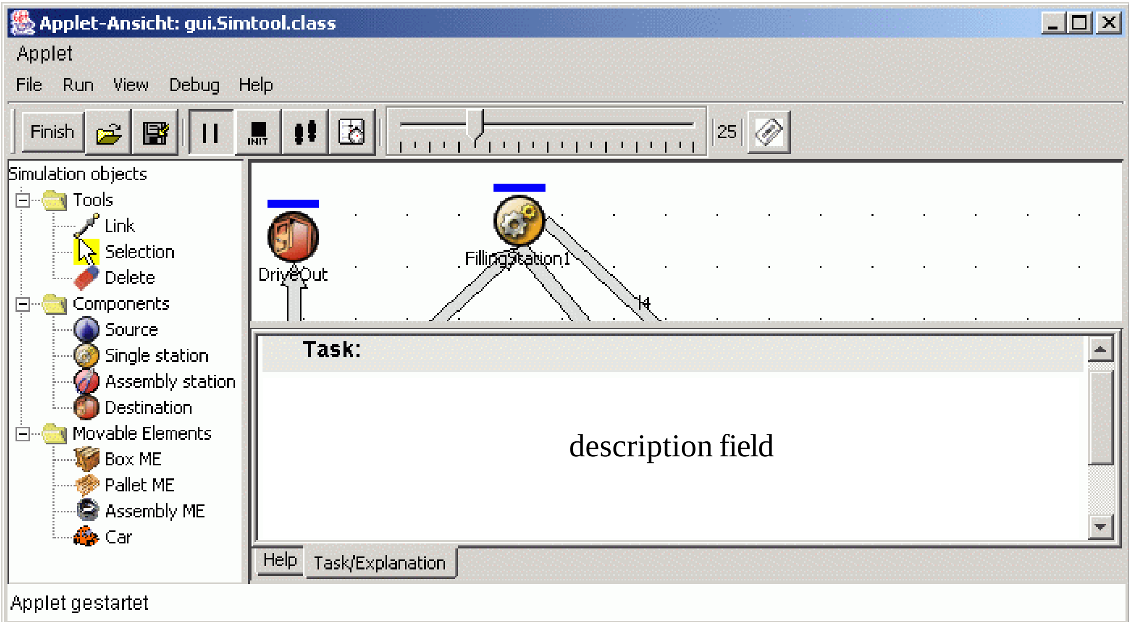


Figure 2: SimTool "Playing"

Java was chosen to be used for the implementation of SimTool, as Java is an objected-oriented programming language especially designed for platform independent applications including browser-based environments in form of applets. However, not only the engines are created with Java. Easy to use GUIs which enable the user to create simulation modules without programming are also available.

External files to keep information about the configuration, scripts to describe automatic sequences and events as well as the models themselves are stored incorporating a specially designed XML-schema. Thus, the files are readable and can be modified using modern editors.

Mathematical probability distributions, movable elements and simulation components are implemented in a modular way. It is possible to add new distributions, new movable elements and new components without having to recompile the system. Other important design goals were the visualisation of internal processes of simulation models and their relation with mathematical issues and the possibility for the users to perform statistical evaluations. A learner should not only see movable elements jumping from simulation component to simulation component. Statistical information should also be made available. For example, the actual state of the components should be indicated by the use of different colours (blocked = red, processing an element = green, preparing= yellow and idle = blue).

The aim of the system is to help teachers in teaching complex facts about simulation by using interactive examples. These examples can be built as tutorials where the student has to perform tasks. A deeper understanding of the subject "simulation" will require the learner to understand not only how the current example works. The learner must also understand the mathematical mechanisms that build the foundation of (successful) simulation. To support the learning processes, the system will work as an assistant for the user. Therefore, the system must be able to react on specific user actions. A possible way to implement this desired behaviour is allowing to attach conditions to instructions. An instruction will be performed only if its conditions are fulfilled. Settings of the components and model properties can be verified. If they are not correct, an instruction will be performed. Of course, an instruction can also be performed in the opposite case, if the settings are correct. In order to support the user in a tutorial, rule-based exceptions are used. An exception is a combination of an instruction and conditions. If the conditions of an exception are fulfilled, the associated instruction will be performed. Therefore, the exception handling mechanism has to be implemented in a modular way. Exceptions can be seen as modules, which themselves consist of smaller units (conditions and instruction). This will help to realise complex tutorials, because the teacher can choose from a variety of script interface methods which can be used in the condition tests.

Instead of defining internal Java API to write scripts via other Java classes, we use an XML-based script-language that comes with various methods to influence all components of SimTool. Thus, designer of simulation models including interactivity and event-driven components do not require knowing Java but can concentrate completely on the script creation process. Besides several advantages like no recompiling and the possibility of using any text editor, the complexity of writing XML-based scripts still requires work and knowledge, and, even though SimTool already allows basic recording of creation processes, it still requires some further development of advanced script editors.

Configuration and Modularity

The lesson file concept allows the teacher to disable distributions, components and routing algorithms. Furthermore, the language and the country can be specified to use a localised version of SimTool. To customize SimTool for a specific lesson, it would be better to have more possibilities. Both, more restrictions and a method to set reasonable default values are needed. In lessons intended for beginners, e.g., it is better to hide a lot of features in the property dialogues of components.
The configuration file which is called “lesson file” stores information about the available components, activated GUI elements, etc.

CONCLUSION

SimTool supports the individual needs of the learner as well as the needs of course designers in the field of simulation. As a complex application it is designed like a small but complete learning environment for teaching simulation as well as being integrated in a larger-scaled learning management system.
It allows a flexible, individualised, and experiential learning in higher education. Different from the traditional lecture and textbooks, the student can influence the processes and realises the direct consequences of his action. SimTool allows a flexible design of course units by writing scripts to guide the students. That is, certain milestones are defined using the XML-based script language, help texts can be integrated that are presented with respect to the difficulty level of the students, and parts of free exploration with given limits are specified. The graphical representation of the simulation model can be adapted by including images.
Simulation problems consist of movable as well as non-movable objects, which have to be related to one another over time. In addition to scheduling problems within production planning, other fields such as project planning and scheduling, the scheduling of threads of computer programs, or flight scheduling can be realised by reusing learning units within other context.
Based on our teaching experience we have come to the conviction that dynamical visualisations of complex aspects may support active learning processes based on one’s own experiences. Furthermore, the motivation of the learner may increase and lead to an active participation and a better internalisation of the material presented.

Acknowledgements

This work was supported by funding from BMBF (Bundesministerium für Bildung und Forschung) in the program “Neue Medien in der Bildung”(08NM094E).

Literature

Eisenberg, R., Reiners T., Sassen I., & Voß S. (2003). Instructional design of interactive learning modules by example of a special flow shop problem. *e-Society 2003, Proceedings, Volume 1*, Iadis, Lisbon, 520-527.

Klie, T., & Schalong, C. (2003). A Script-driven Simulation Tool as an Improvement to the Education in Web-based Virtual Learning Environments. *Diploma Thesis*, University of Technology Braunschweig.

Koch, N. (1999). A Comparative Study of Methods for Hypermedia Development. *Technical Report* 9905, University of Munich.

Reiners, T., Reiß, D., Schulze, H. & Voß, S. (2003). SMART FRAME: An integrated environment for XML-coded learning material. In: Uhr, W., Esswein, W., & Schoop, E. (Eds.) *Wirtschaftsinformatik 2003/Band I*, Physica, Heidelberg 613-632.

Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: von der Technik zur Anwendung

Torsten Reiners[1], Dirk Reiß[2], Imke Sassen[1], Stefan Voß[1]

[1] *Institut für Wirtschaftsinformatik*
Universität Hamburg
Von-Melle-Park 5, D-20146 Hamburg, Germany
{reiners,sassen}@econ.uni-hamburg.de, stefan.voss@uni-hamburg.de

[2] *Abteilung Allgemeine Betriebswirtschaftslehre, Wirtschaftsinformatik und Informationsmanagement*
Technische Universität Braunschweig
Abt-Jerusalem-Straße 7, D-38106 Braunschweig, Germany
d.reiss@tu-bs.de

Abstract

E-Learning, und die damit verbundene Entwicklung von virtuellen Lernumgebungen, kann als einer der Trends der letzten Jahre an den Hochschulen angesehen werden. Negative Beispiele, bei denen eine Gleichsetzung von virtuellen Lernumgebungen mit HTML-Seiten ohne weitere Extras vorgenommen wird, sind gegen die fortgeschrittenen Ansätze abzugrenzen, bei denen eine Motivationssteigerung durch (interaktive) Erweiterungen erzielt wird. Weiterführende Konzepte, die eine Internalisierung des Lernmaterials durch eine praktische Anwendung des Gelernten in realen Szenarien unterstützen, sind hingegen nur begrenzt anzufinden, für den Bereich Operations Research/Management Science aber unabdingbar.

In diesem Beitrag stellen wir SmartFrame vor, einen Prototypen für eine virtuelle Lernumgebung, bei der die Lernmaterialen mittels in XML kodierten, hierarchisch strukturierten Lernobjekten dargestellt werden. Im Unterschied zu anderen Projekten in diesem Forschungsgebiet werden die Meta-Daten der Lernobjekte in der virtuellen Lernumgebung direkt in den Transformationsprozess eingebunden und erlauben somit eine Adaption an individuelle Lernprozesse. Zusätzlich binden wir moderne innovative Komponenten zur Verbesserung der Navigation und Darstellung von Informationen innerhalb des Lernmaterials ein.

Wir präsentieren weiterhin Beispiele für die Gestaltung von Kursen im Operations Research/Management Science. Der Kurs Simulation verwendet ein für die Lehre entwickeltes, diskretes ereignisorientiertes Simulationswerkzeug, welches insbesondere durch die Einbeziehung einer XML-basierten Skriptsprache sowohl innerhalb des Selbststudiums als auch der Lehre Anwendung finden kann. Im Rahmen dieses Kurses wird der Einsatz des Simulationswerkzeuges innerhalb der Lernmethode Synchronized Blended Learning dargestellt. Das zweite Beispiel skizziert einen Kurs über Meta-Heuristiken, wobei hier neben einer Grundla-

gendarstellung insbesondere die weiterführende Anwendung der Theorie auf die Praxis im Vordergrund steht.

1 Einleitung

Angesichts steigender Studierendenzahlen und verringerter finanzieller Ressourcen der Ausbildungsinstitutionen wird die Bedeutung neuer Medien und Kommunikationstechnologien in der Lehre zunehmend größer. Virtuelle Universitäten bieten Möglichkeiten zur Förderung der Qualität der Lehre und werden aus diesem Grund von zahlreichen Ländern in ihr Bildungskonzept integriert; vgl. Hazemi et al. (1998). Die Einbindung von virtuellen Lernumgebungen in das universitäre Curriculum kann dabei neben den traditionellen Präsenzveranstaltungen vielfältige zusätzliche Nutzungsmöglichkeiten eröffnen. Allerdings ist grundsätzlich festzustellen, dass die Benutzer von virtuellen Lernumgebungen während ihres Lernprozesses in nur sehr geringem Maße durch kognitive und konstruktive Werkzeuge didaktisch unterstützt werden und dass ein großes Defizit von virtuellen Lernumgebungen noch immer in einem Mangel an interaktiven Anwendungen auf dynamischen Seiten besteht, mit denen Lernende selbstständig bestehende Daten und Objekte manipulieren beziehungsweise neue generieren können; vgl. Schulmeister (2002b).

Diese Entwicklung ist gerade im Bereich Operations Research und Management Science (OR/MS) von besonderer Wichtigkeit, da aufgrund der Anforderungen an eine praxisnahe Ausbildung ein erhöhter Bedarf an qualitativ hochwertigen Lernangeboten besteht. Der allgemeine Vorteil virtueller Lernumgebungen, Lernenden raum- und zeitunabhängig zur Verfügung zu stehen, fällt in diesem Zusammenhang besonders stark ins Gewicht. Aufgrund seiner Interdisziplinarität bedingt das Gebiet des OR/MS Wissen und Fertigkeiten in vielen Bereichen. Eine ausschließlich theoretische Ausbildung kann demzufolge den komplexen zukünftigen Anforderungen an die späteren Anwender von Methoden des OR/MS kaum gerecht werden. Für Lernende ist es sehr wichtig, sowohl vielfältiges Wissen als auch die insbesondere analytischen Fähigkeiten in Verbindung mit der eigenen Anwendung in den Bereichen Problemmodellierung und Algorithmen, aber auch Softwareentwicklung oder Projektmanagement zu erwerben. Eine wissenschaftliche Ausbildung mit starkem Praxisbezug ist daher unumgänglich. In der aktuellen universitären Lehre ist ein derart gestalteter Zugang jedoch zum größten Teil nicht möglich. Eine virtuelle Lernumgebung mit weiterentwickelten interaktiven Elementen im Bereich des OR/MS kann Lernprozesse durch interaktive Lehrmethoden auf besondere Weise unterstützen. Außer der Vermittlung problemerkennender und problemlösender Fähigkeiten können dadurch praxisbezogene Inhalte verdeutlicht und entsprechend trainiert werden.

Neben dem einfachen Angebot von Lernmaterial ohne eine weitere mediendidaktische Aufbereitung befinden sich im Internet bereits Lernumgebungen, die weiterführende Präsentationsmöglichkeiten neben Hypertext verwenden. Ein Beispiel für Offline-Software ist OR-World (vgl. OR-World (2003)); frei nutzbare Internet-basierte Sammlungen von Lernmaterialien sind beispielsweise tutOR von Sniedovich und Byrne (2003) und darauf aufbauend tutORial von der IFORS (vgl. International Federation of Operational Research Societies (2003)). Die Lernenden bekommen beispielsweise eine Einführung in das Operations Research und können mit interaktiven Darstellungen verschiedener Algorithmen experimentieren. Bereits am Markt etablierte kommerzielle Anwendungen hinsichtlich der Unterstützung der Lehre und der Verwaltung von Kursmaterialien sind unter anderem WebCT (2003) und Blackboard (2003).

In einigen Ländern – z.B. den USA oder Australien – ist es üblich, anstelle Web-basierter Lernumgebungen urheberrechtlich geschützte Software wie Microsoft Excel im Unterricht zu verwenden. Der Lehrende kann die dort vorhandenen Funktionalitäten nutzen, um relativ einfach komplexe Visualisierungen von Problemstellungen und ihren sich schrittweise entwickelnden Lösungen zu implementieren, während der Lernende mit der Software in der Regel bereits umzugehen weiß und sich auf die dargebotenen Inhalte und Übungen konzentrieren kann; vgl. hierzu z.B. die Beiträge von Bell (2000) oder Evans (2000) in der elektronischen Zeitschrift INFORMS Transactions on Education (2003).

Das Projekt VORMS (Virtual Operations Research/Management Science) ist ein Verbundprojekt im Förderprogramm Neue Medien in der Bildung aus dem Förderbereich Hochschulen des Bundesministeriums für Bildung und Forschung. Die wesentliche Zielsetzung des Projektes ist es, neue Organisationsformen basierend auf modernen Informations- und Kommunikationstechnologien zu entwickeln und zu evaluieren. Diese werden zur effektiven Vermittlung individualisierbarer Inhalte aus den quantitativen Wirtschaftswissenschaften eingesetzt, wobei der Schwerpunkt auf dem Fach OR/MS liegt. Die Projektteilnehmer entwickeln jeweils Lernangebote, deren hohe fachliche Qualität durch die einzelnen Kompetenzschwerpunkte gewährleistet wird (vgl. Abbildung 1). Auf diese Weise werden die Möglichkeiten eines einzelnen Lehrstuhls bei Weitem überschritten, da das Gesamtangebot an Lehrmaterialien über das Internet allen Projektteilnehmern sowie den Lernenden verschiedener Institutionen zur Verfügung steht. Obwohl die umgesetzten Inhalte aus einem bestimmten Fachgebiet stammen, ist das einheitliche Gesamtkonzept prinzipiell auch auf andere Fächer übertragbar. Ein Schwerpunkt innerhalb des Projektes ist auf das Prinzip der Wiederverwendbarkeit gelegt worden. Dieses begründet sich unter anderem darin, dass das Fach OR/MS integraler Bestandteil der meisten wirtschaftswissenschaftlichen Studiengänge inklusive Wirtschaftsingenieurwesen und Wirtschaftsinformatik ist.

VORMS
Virtual OR/MS

TP1 Projektkoordination, Entwicklungstools, Schulung
TP2 Grundlagen der Optimierung
TP3 Grundlagen zu Simulation und Heuristiken
TP4 Übungen, Fallstudien, Tests, Entscheidungsunterstützung
TP5 Aufbereitung von Fallstudien
TP6 Anwendungsbereich Logistik
TP7 Anwendungsbereich Industrielle Logistik- und Dienstleistungsmanagement inkl. Entscheidungsmodelle
TP8 Anwendungsbereich Optimierung in der Volkswirtschaftslehre
TP9 Verwertungs- und Transferplanung

Abbildung 1: Übersicht der Teilprojekte

Erst modulare, wiederverwendbare Lernobjekte erlauben Anbietern von unterschiedlichen Studiengängen die Möglichkeit, ihr Angebot mit den entsprechenden Lernmaterialien zu erweitern, diese den Anforderungen entsprechend anzupassen sowie nach Bedarf selbst zusammenzustellen. Inhaltliches Ziel des Projektes ist somit neben der Evaluation von didaktischen und technologischen Methoden die adäquate multimediale und interaktive Aufbereitung von Inhalten des OR/MS, um eine generelle Verbesserung der Qualität der Lehre zu erzielen. Darüber hinaus soll die virtuelle Lernumgebung den interdisziplinären Charakter des Faches betonen, das heißt algorithmisches und methodisches Wissen im Kontext diverser Anwendungen darstellen. Dieses Wissen soll neben den so genannten „soft skills", wie z.B. Kommunikations- und Teamfähigkeiten, durch die Lernmaterialien aufgebaut und gefördert werden. Als weiteres Projektziel ist eine Evaluation hinsichtlich der Eruierung von Gründen für den niedrigen Anteil an Frauen im Gebiet des OR/MS beziehungsweise Möglichkeiten der Förderung und Motivation von Frauen, diese Fachrichtung in ihrem Studium einzuschlagen, hervorzuheben; vgl. hierzu auch Frank et al. (2002) und Frank (2001). An dieser Stelle sei auch auf die sprachliche Problematik hingewiesen, dass sich zum Teil die männliche Form von Wörtern im Sprachgebrauch eingebürgert hat. Im Folgenden soll jedoch keine geschlechtsspezifische Gewichtung vorgenommen werden, das heißt, Personenbezeichnungen wie der Lernende beziehen sich sowohl auf die männliche als auch auf die weibliche Form.

In Abgrenzung zu vielen sowohl praxis- als auch forschungsorientierten Projekten, bei denen eine Realisierung der Lernmaterialien im Wesentlichen die Fokussierung hat, existierende virtuelle Lernumgebungen mit den dort zur Verfügung gestellten Technologien zu nutzen und sich auf die vorhandenen Möglichkeiten der didaktischen Gestaltung zu beschränken, sieht das Projekt VORMS einen relativ hohen Anteil an Forschung und Entwicklung von neuen innovativen didaktischen Komponenten vor. Hierzu zählt neben der Theorie auch eine zumindest prototypische Umsetzung, um so z.B. neben einer Machbarkeitsdarstellung auch entsprechende Evaluationsstudien bei den anvisierten Zielgruppen durchzuführen. Demzufolge können das hier beschriebene Konzept einer virtuellen Lernumgebung sowie die dazugehörigen

Lernmaterialien in der entsprechenden Umsetzung insgesamt nur einen vorläufigen Status besitzen, auch wenn die meisten im Projekt realisierten Komponenten vollständig abgeschlossen werden. Ziel ist es hierbei, dass die in Verbindung mit diesem Projekt entwickelte virtuelle Lernumgebung auf der einen Seite eine unterstützende Funktion in Praktika, Seminaren, Vorlesungen oder Übungen übernehmen kann. Das heißt, es wird den Lernenden die Möglichkeit gegeben, ein zusätzliches studienbegleitendes Lern- und Übungsangebot zu nutzen. Auf der anderen Seite kann die virtuelle Lernumgebung auch als Plattform für rein virtuelle Klassenräume und virtuelle Studienfächer fungieren. Eine wesentliche Zielsetzung ist dabei, den Lernenden durch die Nutzung von modernen Technologien eine einfach konfigurierbare und dennoch flexible Unterstützung für ein unabhängiges Lernen zu bieten. Gleichzeitig soll dabei die Annäherung an Kommunikationsoptionen von Präsenzveranstaltungen erfolgen.

Die vorliegende Arbeit gibt einen Überblick über wesentliche Aspekte der Arbeitspakete im Teilprojekt 3 des VORMS-Projektes sowie über die darüber hinausgehende Entwicklung der virtuellen Lernumgebung (Smart Technology for Research and Modern Education; vgl. auch SmartFrame (2003)). Der Einleitung folgt eine Übersicht der grundlegenden didaktischen Prinzipien. Anschließend wird die Verwendung der Spezifikation Learning Material Markup Language (LMML) erläutert und das Konzept der SmartBars vorgestellt, einer von Reiners et al. (2002) für die Präsentation von Informationen innerhalb von Lernmaterialien eingeführten Methodik. Der nächste Abschnitt beschreibt detailliert das Konzept der virtuellen Lernumgebung und dessen technische Komponenten (vgl. Abschnitt 2.3). Hierzu zählen neben der verwendeten Software die eigentliche Systemarchitektur einschließlich der Zusammenhänge der einzelnen Elemente. Weiterhin beschreiben wir die eigentliche Transformation des XML-basierten Lernmaterials in das vom Lernenden gewünschte Medium. Es wird aufgezeigt, inwieweit die zuvor beschriebenen didaktischen Methoden durch das hier vorgestellte technologische Modell (uneingeschränkt) unterstützt werden können. Im Kapitel 3 wird gezeigt, wie Lernmaterial verwendet werden kann, um die in Abschnitt 3.1 und Abschnitt 3.2 erläuterten speziellen Themenkomplexe des Operations Research, insbesondere Simulation und Meta-Heuristiken, didaktisch und mediengerecht aufbereitet darzustellen. Darüber hinaus werden in diesen Abschnitten didaktische Methoden beschrieben, die im Rahmen des Projektes erarbeitet wurden. Hierzu zählen unter anderem Synchronized Blended Learning zum direkten Abgleich unterschiedlicher Lehrmethoden (vgl. Abschnitt 3.1.2) sowie die durch die virtuelle Lernumgebung unterstützte Nachbearbeitung des Lernmaterials anhand realistischer Beispiele (vgl. Abschnitt 3.2.2). Der Beitrag schließt in Abschnitt 4 mit einer Zusammenfassung sowie einem Ausblick.

2 Didaktik

In diesem Abschnitt beschreiben wir die innerhalb des Projektes aufgestellten didaktischen Modelle sowie deren Übertragung auf den Prototypen einer virtuellen Lernumgebung. Neben einer auf den Lernenden spezifizierten Adaptivität der Präsentation gehen wir insbesondere auf die semantische Gestaltung der Lernmaterialien und die Projektion von zusätzlichen Informationen innerhalb des Lernmaterials in eine aggregierende Informationspräsentation - so genannte SmartBars - ein. Im Anschluss wird ein Konzept für eine virtuelle Lernumgebung vorgestellt, welches darüber hinaus in einem Prototyp zum Zwecke der Demonstration der didaktischen Modelle realisiert wurde.

2.1 Konfigurationsoptionen und Funktionen aus didaktischer Sicht

Grundsätzlich thematisieren mediendidaktische Ansätze die Erarbeitung beziehungsweise Verbesserung von Unterrichtsmedien sowie ihren geeigneten Einsatz im Lehr- und Lernprozess. Das didaktische Design beschäftigt sich in diesem Zusammenhang mit der Planung, Konzeption, Durchführung und Evaluation von Lehr- und Lernprozessen, wobei eine konkrete Gestaltung sowohl theoretisch als auch empirisch belegt sein sollte. Eine rezeptartige Anleitung für das didaktische Design einer virtuellen Lernumgebung kann und sollte es aufgrund der Komplexität der zu beachtenden Faktoren nicht geben. So ergibt sich die Auswahl einer geeigneten Präsentationsform innerhalb eines bestimmten Mediums immer aus der wechselseitigen Abhängigkeit von Lernzielen, Lehrinhalten, Lehr-Lernmethoden und Sozialformen. Die Wahl des Mediums sollte diese Wechselwirkungen stets berücksichtigen und keinesfalls im Vorhinein festgelegt werden, das heißt, die Medienwahl ist der didaktischen Konzeption unterzuordnen; vgl. Klein (2000).

Der Erfolg des Lernens mit einer virtuellen Lernumgebung ist unter anderem von der gegenseitigen Beeinflussung der verschiedenen eingesetzten Medien auf der einen Seite und den Merkmalen des Lernenden auf der anderen Seite abhängig. Auf der Lernerseite wird dabei zwischen aufgabenübergreifenden Eigenschaften (z.B. der Bevorzugung eines bestimmten Lernkanals beziehungsweise einer bestimmten Sinnesmodalität, einer bestimmten Symbolik sowie dem persönlichen kognitiven Stil) sowie aufgabenspezifischen Eigenschaften unterschieden. Parallel zu den Möglichkeiten traditioneller Unterrichtsmethoden, individuell auf Lernende einzugehen, sind innerhalb einer virtuellen Lernumgebung Methoden zu entwickeln, die persönliche Fähigkeiten, Motivationen, Interessen, Einstellungen und Ziele des Lernenden berücksichtigen. Die sich in der mediendidaktischen Forschung durchgesetzte kognitionspsychologische Sichtweise zeichnet sich dabei durch die Untersuchung von Wahrnehmung, Informationsinterpretation und -verarbeitung sowie der inhaltlichen und strukturellen Verknüpfung von Bild und Sprache aus.

Lernen wird als aktiver Prozess der Konstruktion von Wissen, basierend auf dem Vorwissen und der Entwicklung kognitiver Fähigkeiten verstanden, wobei individuelle mentale Modelle entstehen; vgl. van Merrienboër et al. (2002). Für das Design von virtuellen Lernumgebungen bedeutet dies, kognitive Lernprozesse in handelnder Auseinandersetzung mit der Umwelt zu ermöglichen (vgl. Schulmeister (2002a)) und dabei Wissen und Annahmen über die menschliche Kognition zu berücksichtigen. Das heißt, die Architektur einer virtuellen Lernumgebung sollte auf der kognitiven Struktur des lernenden Individuums basieren; vgl. Strittmatter (2000). Lernmaterial muss aus diesem Grund in Darbietungsmodalität, Struktur, Schwierigkeitsgrad und Sequenzlänge und -abfolge variieren, um möglichst vielen verschiedenen Lernenden einen positiven und effektiven Umgang mit der Lernumgebung zu ermöglichen.

Als Konsequenz ist bei der Entwicklung einer virtuellen Lernumgebung die Integration vielfältiger Konfigurationsmöglichkeiten vorzunehmen, um durch die Adaption an die Lernvoraussetzungen zum einen eine extreme Differenzierung nach Lernertypen zu ermöglichen und zum anderen eine Offenheit für diverse Lernstrategien zuzulassen; vgl. Schulmeister (2000). Den Lernenden wird auf diese Weise die Möglichkeit gegeben, sich eine Lernumgebung entsprechend eigener Ansprüche, Vorlieben und kognitiver Stile zu gestalten. Eher auf formaler Ebene sind dabei Konfigurationsmöglichkeiten betreffend der Farbgestaltung, Anordnung der Arbeitsleisten, Buttons und Fenster, Spracheinstellung, Kommunikationsformen sowie die verschiedenen Einstellungsmöglichkeiten der SmartBars (vgl. hierzu Abschnitt 2.2) zu nennen. Zusätzlich ergibt sich daraus die Möglichkeit, dass die Lernumgebung das bevorzugte Medium (Text, Audiosequenzen, Video, Abbildung, Animation) berücksichtigt, wenn die Inhalte dort redundant vorliegen, beziehungsweise dass bestimmte Medien, wie zum Beispiel Audiosequenzen, ausgeschlossen werden. Weitergehende Konfigurationsmöglichkeiten, die eher auf einer inhaltlichen Ebene angesiedelt werden können, werden aufgrund des modularen Aufbaus der Lernumgebung realisierbar.

Es erfolgt eine weitere Unterteilung des Lehrstoffs in semantische Module unter Verwendung von LMML; vgl. Abbildung 2 sowie Süß (2003). LMML wird zur Kodierung des Lernmaterials verwendet, wobei eine hierarchische Struktur benutzt wird; vgl. auch Abschnitt 2.3 für Analogien zu der weiteren Untergliederung des Lernmaterials in verschiedene Lernobjektarten. Dabei werden die Elemente von LMML in vier größere Kategorien eingeteilt, die im Wesentlichen einer Kategorisierung hinsichtlich ihrer semantischen Funktionalität innerhalb des Objektes entsprechen. Die Kombinierbarkeit der verschiedenen Elemente, besonders die Möglichkeit der gegenseitigen Aggregation, wird in Abbildung 2 verdeutlicht. Ausgehend von Strukturmodulen werden Inhaltsmodule verwendet, die insbesondere dem Objekt eine semantische Bedeutung geben und für den Aufbau von Lerneinheiten im Sinne einer didaktischen Gestaltung dienen. Wie in der Abbildung dargestellt, sollte eine Lerneinheit mit einer Motivation beginnen, gefolgt von einem Inhalt bestehend aus mehreren weiteren Inhaltsmodulen, und mit einer Zusammenfassung sowie gegebenenfalls Rückmeldungen und Aufgaben beendet werden. Darüber hinaus gibt es optionale Inhaltsmodule, die eine Lerneinheit im Hinblick auf weitergehende Informationen oder Interaktionsmöglichkeiten verbessern. Inhaltsmodule wiederum bestehen aus Strukturobjekten sowie Medienobjekten, mit denen der Inhalt an sich angegeben bzw. strukturiert wird.

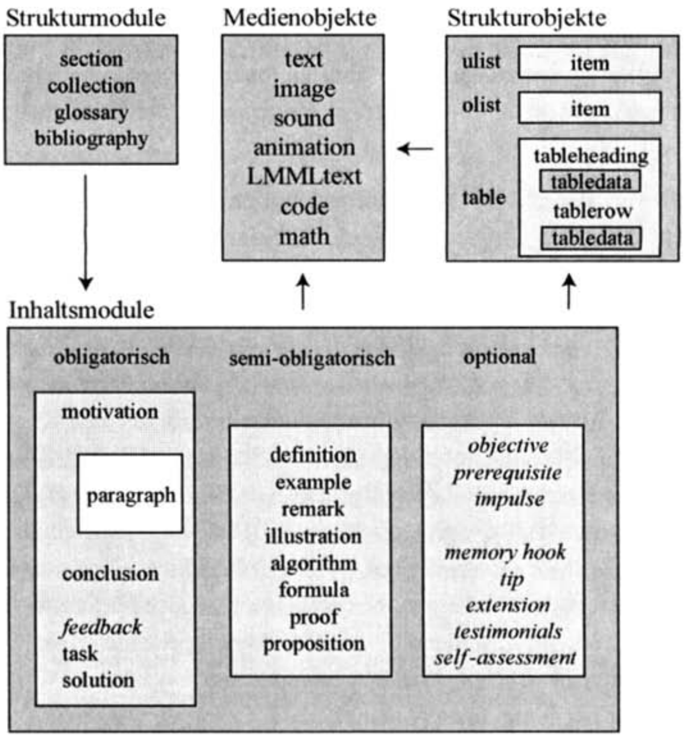


Abbildung 2: Struktur von LMML

Eine explizite Unterteilung in spezifische, semantische Module wird im Wesentlichen für den Lernenden vorgenommen, damit eine Kennzeichnung bei der Darstellung erfolgen kann, z.B. eine Hervorhebung von Hinweisen oder Impulsen. Darüber hinaus können Lernende aufgrund von individuellen Einstellungen festlegen, inwieweit bestimmte Module für sie relevant sind und in der Präsentation vorkommen sollen. So kann bei einem explorativen Durchlaufen des Lernmaterials die Darstellung von Modulen wie z.B. Impuls unterdrückt und entsprechend andere wie z.B. Zusammenfassung hervorgehoben werden. Diese Konfigurationsmöglichkeiten schließen beispielsweise auch eine semiautomatische Konfiguration unter Einfluss von Selbsteinschätzungstests im Vorab sowie die Integration von Übungen und Lernkontrollen im Nachhinein ein.

Das Ausmaß der (automatischen) Adaptivität der Lernumgebung an den individuellen Benutzer sollte jedoch vom Lernenden selbst bestimmt werden können. Bezüglich des Schwierigkeitsgrades ergibt sich daraus z.B. die Option, entweder eigene Festlegungen vorzunehmen oder auf Basis seiner Ergebnisse bei Übungsaufgaben eine adaptive Reaktion und Selektion des weiteren Lernweges durch die Lernumgebung auszuwählen. Den Lernenden sollen verschiedene festgelegte Lernpfade angeboten werden, die jedoch nach eigenem Ermessen verlassen und wieder betreten werden können. Darüber hinaus sind Möglichkeiten zur eigenständigen Festlegung von Lernpfaden - beispielsweise basierend auf dem Ergebnis eines Selbsteinschätzungstests - sowie die Modifikation von vorgegebenen Lernpfaden vorzusehen.

2.2 SmartBars

Das Lernmaterial innerhalb von Lerneinheiten ist häufig mit Marginalien und weiteren Markierungsvariationen innerhalb des Textes versehen. Generell sollen solche Markierungen die Lernenden bei der Organisation des Inhalts unterstützen, indem sie als Orientierungshilfen fungieren. Dabei können sie das längerfristige Behalten verbessern, da sie die Makrostrukturen des Inhalts verdeutlichen. Dennoch sollte eine „Überdidaktisierung" von Lehrinhalten vermieden werden (vgl. Ballstaedt (1997)), um den Lernenden geeignete Handlungsfreiräume zu geben. Weitere Informationen, welche in virtuellen Lernumgebungen innerhalb des Lernmaterials dargestellt werden, sind Verlinkungshinweise, Literaturverweise, Glossar-Einträge, Markierungen zur Hervorhebung von bestimmten Begriffen sowie Fußnoten zur Ergänzung der gegebenen Texte. Bei ungeeigneter Präsentation aller Information kann es zu einer Überflutung an Farben, Formatierungen und Sonderzeichen kommen, so dass eine Lesbarkeit nur eingeschränkt gegeben ist; z.B. werden Glossar-Einträge z.T. durch einen Pfeil angedeutet.

Um einer solchen Informationsproliferation in der virtuellen Lernumgebung vorzubeugen, wurde das Konzept der SmartBars entwickelt und in einem Prototyp für die Darstellung von verschiedenen Informationen umgesetzt (vgl. Abbildung 3). Differenzierte Nutzungsmöglichkeiten der SmartBars sollen den Lernenden einen flexiblen Umgang mit den Lerneinheiten ermöglichen (vgl. Reiners et al. (2002)). Im Lernmaterial vorhandene Informationen werden durch entsprechende Randmarkierungen gekennzeichnet und nur bei Bedarf, das heißt auf Wunsch des Lernenden, im Lernmaterial eingeblendet. Je nach Art der Information werden am Rand unterschiedliche Farben eingesetzt, so dass eine Unterscheidung hinsichtlich der Art der Information erfolgt. Dies trägt zur Verbesserung des Benutzerkomforts und der Übersichtlichkeit bei, da eine direkte Einblendung von Informationen in den Lernmaterialien lediglich aufgrund individueller Einstellungen erfolgt. Beispielsweise werden Verweise auf Literatur gegebenenfalls nicht direkt durch entsprechende textuelle Repräsentationen im laufenden Text angegeben, sondern am Rand durch eine farbliche Markierung.

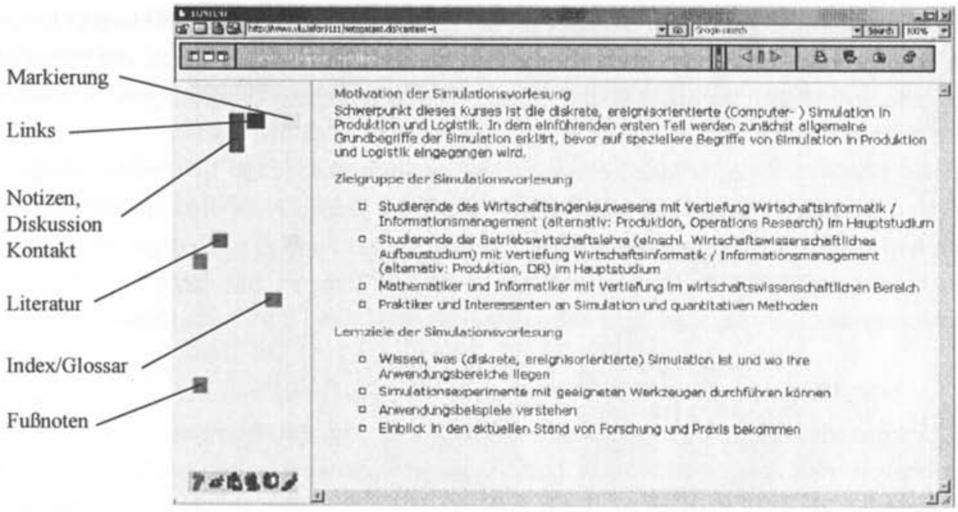


Abbildung 3: Darstellung der SmartBars innerhalb der Lernumgebung

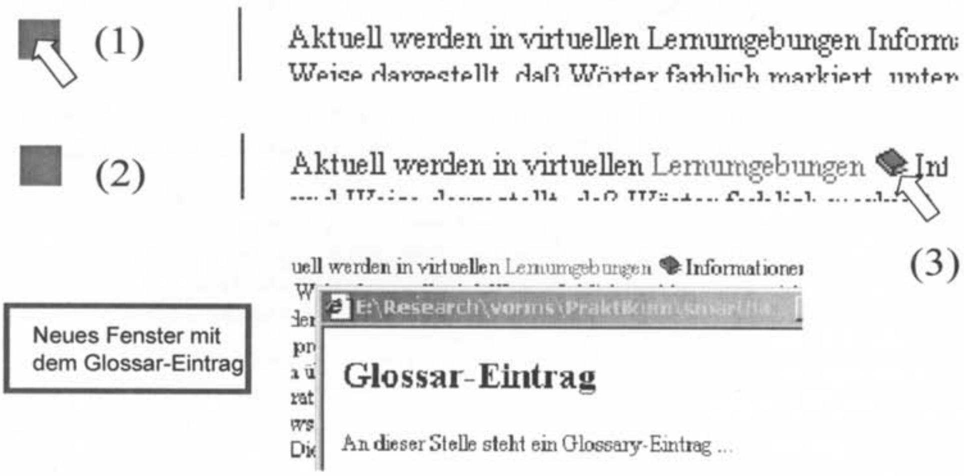


Abbildung 4: Verwendung der SmartBars für die Darstellung eines Glossar-Eintrags

Der Lernende kann durch eine Betätigung dieser Markierung mit der Maus – hierbei wird ein Überfahren (1) für eine kurzfristige und ein Anklicken (2) für eine dauerhafte Einblendung unterschieden – den eigentlichen Literaturverweis im Lernmaterial einblenden und verwenden; vgl. auch Abbildung 4, in der die Einblendung eines Glossar-Eintrags durch Anklicken des eingeblendeten Links in Form eines Buches (3) in einem neuen Fenster erfolgt.

Der Lernende kann weiterhin eigene Anmerkungen in die Lerneinheit einfügen und sich somit das Lernmaterial „zu eigen“ machen, indem er seinen eigenen Lernweg kommentiert oder eigene Schwerpunkte setzt. Beispielsweise können während und nach dem Prozess des Verstehens entstehende Verknüpfungen mit unterschiedlichen Arten und Inhalten des Vorwissens in diesen Anmerkungen festgehalten werden. Diese ganz individuellen Einfälle oder Vorstellungen, die auch als Elaborationen bezeichnet werden, sind aus didaktischer Sicht sehr bedeutsam, da sie Auswirkungen auf den Behaltensprozess haben.

Das Konzept der SmartBars sieht weitere Vorteile für das Lernen in der virtuellen Lernumgebung gegenüber der Verwendung herkömmlicher gedruckter Lernmaterialien vor. Bei den gedruckten Lernmaterialien ist nach dem Hinzufügen eigener Anmerkungen ein zweiter Arbeitsdurchlauf gegebenenfalls durch zu viele unübersichtliche Anmerkungen gestört. Zudem gewinnt der Lernende während einer zweiten Durcharbeitungsphase möglicherweise eine modifizierte Einstellung zum Text und möchte dieses zusätzlich festhalten. Die SmartBars können den Lernenden eine Funktionalität zur zeitlichen Protokollierung von Anmerkungen bieten, die dann hinsichtlich unterschiedlicher Sortierungskriterien angezeigt werden können. Auf diese Art kann ein individueller Überblick über den eigenen Lernfortschritt geboten werden.

Den Lernenden werden weiterhin verschiedene Kommunikationskanäle geboten, die innerhalb der SmartBars eingebunden werden. Hierzu zählen neben einem direkten Kontakt zu den Verantwortlichen der Lerneinheit – wie z.B. Autoren sowie vortragenden oder betreuenden Personen der Lerneinheit – auch die Eröffnung neuer beziehungsweise Teilnahme an existierenden Diskussionen in Form von Foren oder Messageboards. Kommunikationen beziehen sich auf ausgewählte Passagen der Lerneinheit und werden durch eine seitliche Markierung mittels der SmartBars entsprechend gekennzeichnet. Durch diese Methodik erfolgt eine Kommunikation zwischen verschiedenen Personengruppen und damit ein fachlicher Austausch sowie eine Förderung der „soft skills“, z.B. der Kommunikations- und Teamfähigkeit. Darüber hinaus kann der Inhalt der Diskussionen für eine Bewertung der Lerneinheit verwendet werden, so dass diese entsprechend der auftretenden Erfordernisse, das heißt Probleme der Lernenden beziehungsweise Wünsche hinsichtlich Ergänzungen oder Modifikationen des Lernmaterials, angepasst werden. Diese Form der Qualitätssicherung hat auf die Lernenden einen motivierenden Charakter, da ein direkter Zusammenhang zwischen der Rückmeldung der Lernenden und der Gestaltung der eigentlichen Lernmaterialien und der Lehre besteht.

2.3 Systemarchitektur

Verfolgt man die Evaluation neuer didaktischer Methoden, so bedarf es geeigneter technologischer Voraussetzungen und entsprechender Grundlagen zur Integration in virtuelle Lernumgebungen, die in bestehenden Umgebungen nicht beziehungsweise in nicht ausreichender Weise gegeben sind. Aufgrund dessen wurde alternativ zu einer extensiven Modifikation existierender Applikationen von den Autoren ein Prototyp für eine virtuelle Lernumgebung entwickelt, der neben der Verwendung von frei verfügbaren Softwarekomponenten eine Einbindung von innovativen Konzepten erlaubt. Hervorzuheben ist in diesem Zusammenhang insbesondere die Verwendung von Standards sowie Spezifikationen in einem einheitlichen Gesamtkontext, sowohl aus technologischer als auch aus didaktischer Sicht. Aufgrund inhaltlicher Inkonsistenzen, notwendiger und vorgesehener Erweiterungen von Spezifikationen sowie einer Vermeidung von Redundanzen wurden jedoch vereinzelt Ergänzungen bzw. Modifikationen vorgenommen; siehe hierzu auch Reiners et al. (2003).

2.3.1 Technische Realisierung

Die Systemarchitektur lehnt sich im Groben an die Konzepte aus dem Projekt OR- World (vgl. hierzu Kassanke und Suhl (2000)) an, wobei jedoch sowohl weitreichende Modifikationen durchgeführt als auch Erweiterungen in das Konzept integriert wurden; vgl. hierzu auch Reiners et al. (2002) und Reiners et al. (2003). Basierend auf einer der Umsetzung vorgelagerten Anforderungsevaluation (vgl. z.B. Frank et al. (2002) und Frank (2001)) wurden insbesondere Möglichkeiten einer vielfältigen Konfiguration entsprechend individueller Präferenzen mit dem Ziel einer positiven Beeinflussung des individuellen Lernerlebnisses berücksichtigt. Weitere für eine virtuelle Lernumgebung essentielle Komponenten, wie z.B. Navigation und Kommunikation, sollen innerhalb dieses Projektes hinsichtlich Funktionsumfang und Gestaltung evaluiert und entsprechend der dort erhaltenen Ergebnisse neben weiteren Ideen in die Umsetzung einbezogen werden. Hierbei ist zu beachten, dass die von uns entwickelte virtuelle Lernumgebung keinen Anspruch auf Vollständigkeit erhebt, sondern vielmehr als Prototyp zur Darstellung der von uns implementierten Konzepte dient, welche ansonsten nicht ohne z.T. sehr fundierte Kenntnisse anderer Systeme umzusetzen wären. Hiervon sind insbesondere kommerzielle Systeme ausgenommen, welche oftmals den Nachteil besitzen, dass man auf die von der Software bereitgestellte Funktionalität zumeist nur eingeschränkten und auch keinen modifizierenden Zugriff hat.

Das Lernmaterial wird in der Spezifikation XML (eXtensible Markup Language) kodiert, wobei zusätzliche Spezifikationen wie LMML, QTI (Questionnaire and Test Interoperability; vgl. IMS Global Learning Consortium (2003a)) sowie MathML (Mathematical Markup Language; vgl. W3 Consortium (2003a)) zur eigentlichen Darstellung der modularen, in eine hierarchische Struktur eingebetteten Lernobjekte verwendet werden. Lernobjekte — kleine semantische Einheiten des Lernmaterials — werden in XML-kodierten Dateien gespeichert. Jedem Lernobjekt werden in einer Datenbank gespeicherte zusätzliche Meta-Daten zugeordnet, wodurch neben einer reinen inhaltlichen Spezifikation auch Informationen bezüglich der Verwendung, der Rechte, der zeitlichen Entwicklung, der Beziehung zu anderen Lernobjekten auch technische Anforderungen angeführt werden können. Innerhalb des Projektes wird der IEEE Standard LOM (Learning Object Metadata; vgl. IEEE Learning Technology Standards

Committee (2003)) eingesetzt, der die Meta-Daten in mehrere Kategorien einordnet. Elemente in den einzelnen Kategorien klassifizieren das Lernobjekt entsprechend. So wird in der Kategorie Educational z.B. eine Unterscheidung hinsichtlich des Schwierigkeitsgrades (Anfänger, Fortgeschrittener, Experte) oder der Zielgruppe (Studierende im Grundstudium, Studierende im Hauptstudium) durch die Verwendung eines gegebenen Vokabulars angeführt. Gerade diese Meta-Daten unterstützen neben den Lernenden bei der (automatischen) Wahl der für sie relevanten Kurseinheiten — hierunter ist gegebenenfalls auch die systemseitige Adaptivität an das Lernverhalten zu verstehen — auch die Autoren von Lerneinheiten, da hierdurch eine gezielte Suche in der Menge der existierenden Lernobjekte ermöglicht und damit die Wiederverwendbarkeit an sich gefördert wird. Eine Kodierung der Meta-Daten kann neben LOM auch durch weitere Spezifikationen von wichtigen Organisationen kodiert werden; hierzu zählen insbesondere die Spezifikation ARIADNE (vgl. ARIADNE Foundation (2003)), SCORM (Advanced Distributed Learning Initiative (2003)) und IMS Meta-data (vgl. IMS Global Learning Consortium (2003b)). Die Meta-Daten können unabhängig von der gewählten Kodierung in Datenbanken mit unterschiedlichen Architekturparadigmen gespeichert werden. In dem Projekt werden relationale (z.B. MySQL von MySQL (2003)) sowie XML-basierte Datenbanken (z.B. Xindice von Apache Software Foundation (2003b)) eingesetzt. Durch die Verwendung von Datenbanken und entsprechende Verweise auf die in Dateien abgelegten Inhalte kann eine Verteilung der Lernmaterialien in Netzwerken erfolgen, wobei hierzu im Internet verfügbare Adressen in Form einer URI (Uniform Resource Identifier) erlaubt sind.

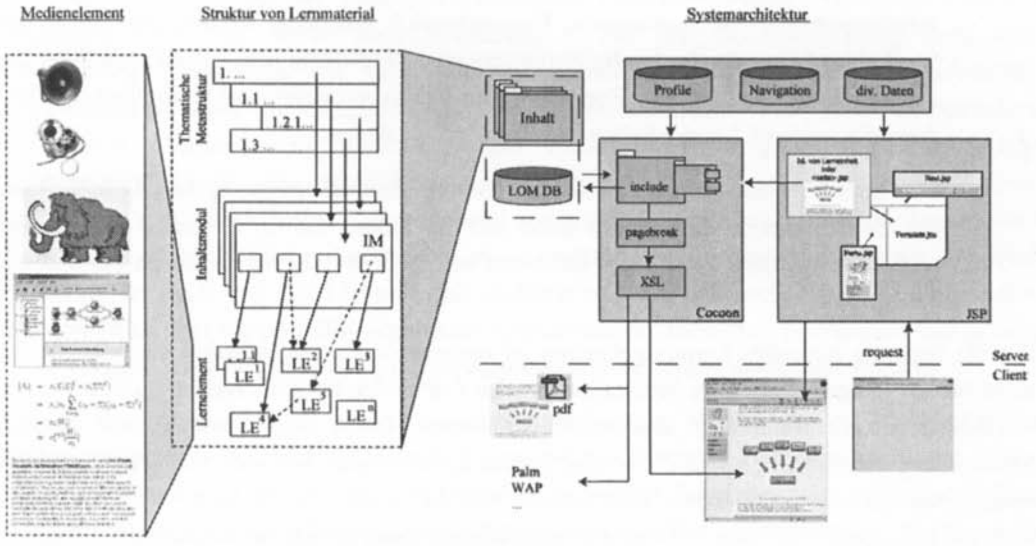


Abbildung 5: Systemarchitektur von SmartFrame

Abbildung 5 zeigt die Systemarchitektur von SmartFrame. Auf der linken Seite der Abbildung sind eine Darstellung unterschiedlicher Lernobjekte und deren Strukturierung gegeben. Dabei erfolgt eine Einstufung der Lernobjekte in vier verschiedene, durch spezifische Regeln im Vorfeld festgelegte, Granularitäten. Insbesondere sind durch eine geeignete Umsetzung der Lernobjekte Abhängigkeiten zwischen den einzelnen Lernobjekten zu verhindern, damit das Potential einer zukünftigen Wiederverwendung in unterschiedlichen Zusammenhängen erhöht wird. Es werden die folgenden Arten von Lernobjekten unterschieden, wobei diese jeweils zu größeren Einheiten miteinander kombiniert werden können.

- Medienelement, kleinste Einheit, die weitere Lernobjekte referenzieren, aber nicht einbinden kann. Hierzu zählen z.B. Textbausteine, Animationen, Bilder oder Audiosequenzen.
- Lernelement (LE), eine Komposition von mehreren Lernobjekten als semantische Einheit, wobei an dieser Stelle sowohl Medienelemente als auch Lernelemente als Lernobjekte bezeichnet werden. Unter einer semantischen Einheit ist der Inhalt eines Lernobjektes zu verstehen, welcher aus Sicht der Autoren auf einer „Seite" virtuell oder gedruckt dargestellt werden kann. Die Lernenden können jedoch zwischen verschiedenen Darstellungen wählen; so kann ein Lernobjekt auch auf mehrere Seiten aufgeteilt werden.
- Inhaltsmodul (IM) gruppiert die Lernelemente zu größeren Einheiten, wobei Inhaltsmodule einen bestimmten Themenbereich umfassen sollten. Dies kann der Stoff einer Lerneinheit oder der Foliensatz einer Vorlesung sein. Eine Darstellung für den Lernenden sollte in der Regel über ein Inhaltsverzeichnis erfolgen, über welches dann die weiteren Lerneinheiten erreicht werden können.
- Thematische Meta-Struktur stellt eine weitere Zusammenfassung von Lernobjekten dar. Semantisch handelt es sich dabei z.B. um Kurse, die der Lernende im Rahmen seines Curriculums zu belegen hat, oder um größere Arbeiten in einem bestimmten Fachgebiet.

Für die Nutzung virtueller Lernumgebungen ist insbesondere auch aufgrund der Vielfalt der angebotenen Umgebungen eine Motivation bei den Lernenden zu initiieren, sich mit einer zur Verfügung gestellten Plattform auseinander zu setzen. Hierzu besteht neben einer eigenen Entwicklung eine Option auf die Nutzung bereits existierender Software mit gegebenenfalls entsprechenden Einschränkungen. Eine weitere Option ist die Verwendung von Web-Browsern, die sich durch eine hohe Verbreitung auszeichnen. Beim Lernenden können grundlegende Kenntnisse im Umgang mit Web-Browsern, insbesondere im Umgang mit Browser-basierten Materialien, bereits vorausgesetzt werden, so dass die geringe Einarbeitungszeit des Lernenden für eine Integration von Web-Browsern spricht. Des Weiteren kann der Funktionsumfang des

Prototyps relativ einfach dadurch erweitert werden, dass im Browser existierende Schlüsselkomponenten, wie z.B. die Darstellung von HTML (Hypertext Markup Language) unter Einbeziehung verschiedener Medienarten wie Bilder, Töne und Filme, Kommunikationsformen (z.B. E-Mail) sowie Skriptsprachen (z.B. JavaScript) bereits vorliegen und ohne intensiven Entwicklungsaufwand genutzt werden können. Dies erlaubt innerhalb der Entwicklung eine Fokussierung auf neue und ergänzende Komponenten unter Nutzung der existierenden Standardkomponenten, wobei jedoch auf Kompatibilitäten zwischen den unterschiedlichen Browsern zu achten ist.

Bei der technischen Umsetzung einer modernen virtuellen Lernumgebung ist auf eine Verwendung von aktuellen Technologien sowie internationalen Standards Wert zu legen. Nur auf diesem Wege kann sich eine internationale Akzeptanz und Wiederverwendbarkeit unabhängig von der bestehenden Softwarearchitektur herauskristallisieren. Letzteres stellt sich als das weitaus größere Problem dar. Im Rahmen der Entwicklung von SmartFrame wird eine ausschließliche Verwendung von frei verfügbaren Software-Paketen favorisiert, um bereits zu Beginn eine mögliche Abhängigkeit von proprietären und damit herstellergebundenen Entscheidungen bei Softwaremodifikationen zu vermeiden. Die rechte Seite der Abbildung 5 zeigt in einer schematischen Darstellung alle hierzu relevanten Komponenten.

Der Server in dem Client-Server-Modell setzt sich aus zwei Technologien zusammen. Zum einen ist dies eine graphische Benutzerschnittstelle (GUI), welche mittels JavaServer Pages (JSP) unter Verwendung der Tomcat-Servlet-Engine (Apache Software Foundation (2003c)) umgesetzt wird; vgl. hierzu auch Abbildung 8. Bei der GUI-Entwicklung steht eine Priorisierung hinsichtlich der Flexibilität der Konfigurationsmöglichkeiten, Modularität, Erweiterbarkeit sowie Unabhängigkeit von den Lernmaterialien im Vordergrund. Insbesondere der zuletzt genannte Punkt ermöglicht eine Verbesserung der geforderten didaktischen Aufbereitung sowie eine individualisierte Darstellung von Lernmaterialien, da diese unabhängig von der virtuellen Lernumgebung angepasst werden können. Darüber hinaus wird dadurch die Möglichkeit des Austausches von Lernmaterialien gefördert. Technisch wird eine Trennung der GUI und des Lernmaterials dadurch erreicht, dass entsprechende Lernobjekte ausschließlich über die im LOM gegebenen Schlüsselwerte (Identifier) referenziert und in dem dafür vorgesehenen Bereich nach einer serverseitigen Generierung dargestellt werden.

Eine Aufbereitung des XML-basierten Lernmaterials erfolgt durch Apache Cocoon (Apache Software Foundation (2003a)), einer Software insbesondere zur Transformation von XML-Daten mittels XSL-Transformation (eXtensible Stylesheet Language; vgl. W3 Consortium (2003b)) unter Berücksichtigung von benutzerspezifischen Einstellungen sowie dem gewünschten Ausgabemedium. Zum Beispiel werden Folien innerhalb des Browsers mit den entsprechend in der Größe angepassten Schriften angezeigt, während eine Transformation in das Ausgabeformat PDF (Portable Document Format) eine zusätzliche Komposition von mehreren Folien auf einer Seite für ein Hand-out an die Lernenden vorsieht.

In Abgrenzung zu anderen Projekten – beispielhaft seien an dieser Stelle die Projekte wwr-Projekt (wwr wissenswerkstatt rechensysteme (2003)) sowie CMS-W3 (Fraunhofer Institut für Graphische Datenverarbeitung (2003)) genannt – erfolgt die Zusammensetzung und entsprechende Transformation direkt, das heißt, dass auf Anfrage die zugrunde liegenden XML-Daten umgewandelt werden und nicht bereits hinsichtlich bestimmter Kriterien vorliegen. Des Weiteren erfolgt eine direkte Einbeziehung der Meta-Daten für jedes untergeordnete Lernobjekt. Dadurch wird mit einer direkten Verlinkung der Lernobjekte eine Einbeziehung der in den Meta-Daten abgelegten Informationen vorgenommen. Dieses wird insbesondere am Beispiel von LMML mit einer direkten Verlinkung auf weitere Lernobjekte ohne Verwendung von anderen Meta-Daten deutlich. LMML umgeht diese Problematik durch die Verwendung von zusätzlichen in der Spezifikation integrierten Elementen zur Angabe von Meta-Daten, was jedoch mit dem Verständnis des XML-Prinzips der Autoren im Konflikt steht: Inhalte sind von den beschreibenden Meta-Daten getrennt zu halten, um mögliche Inkonsistenzen bzw. Redundanzen zu vermeiden.

Weitere Systemkomponenten sind auf einem Server laufende Datenbanken zur Speicherung der Meta-Daten, Navigationsinformationen beziehungsweise Lernpfade, Benutzerprofile sowie Informationen zu Diskussionsforen, Neuigkeiten beziehungsweise weiteren Interessengebieten; für eine detailliertere Beschreibung siehe Reiners et al. (2002).

Lernenden ist gegebenenfalls kein direkter ununterbrochener Zugriff auf den Server gewährt. Hierzu ist ein zusätzlicher so genannter Offline-Modus in das Konzept aufzunehmen. Dabei wird auf einer lokalen Installation mit den entsprechend notwendigen Komponenten, das heißt, neben den Datenbanken insbesondere der Technologie zur Transformation der XML-Inhalte, gearbeitet. Der Umfang der lokalen Installation kann im Gegensatz zu einem Server eingeschränkte Funktionalität besitzen, was sich positiv auf die Installationszeit sowie Anforderungen auswirken kann. Eine Synchronisation zwischen dem Server und dem Client erfolgt durch geeignete Replikationsmechanismen bei den Datenbanken. Zum Beispiel ist bei der Übertragung der Lernmaterialien auf eine Berücksichtigung der Bandbreite zu achten, und entsprechende mediale Komponenten sind entweder in einer schlechteren Qualität, und damit Größe, oder nur auf expliziten Wunsch des Lernenden zu transferieren. Trotz fehlender Funktionalität, wie z.B. Gruppenarbeit, E-Mail oder Chats, kann ein ortsunabhängiges Arbeiten realisiert werden, indem bei einem Wechsel eine Replikation zwischen Client und Server erfolgt. Die Replikation muss neben einer inhaltlichen Replikation von Lernmaterialien auch entsprechende Einstellungen in der GUI oder für das Lernmaterial einbeziehen. In diesem Zusammenhang ist auch der Autor beziehungsweise Lehrende aufgefordert, bei der Erstellung der Lernmaterialien eine Berücksichtigung von Faktoren wie Übertragungsmedien und Kompatibilitäten vorzunehmen. Zum Beispiel sollten alternative Lernobjekte realisiert werden, die anstelle von hochauflösenden — und damit großen — Multimedia-Dateien angezeigt werden und den Lernenden die Entscheidungen überlassen, ob aufgrund der ihnen gegebenen Informationen eine Übertragung als notwendig anzusehen ist. Hierdurch kann einer Demotivation durch lange Pausen entgegengewirkt und den Lernenden eine direkte Beeinflussung der Präsentation ermöglicht werden.

Zu den wesentlichen Komponenten gehören z.B. die SmartBars zur aggregierten Darstellung von inhaltlichen Informationen des Lernmaterials (vgl. auch Abschnitt 2.2), hyperbolische Graphen zur graphischen Darstellung von Zusammenhängen zwischen den einzelnen Lernobjekten sowie dem bisher durchlaufenen Lernpfad oder — im Fall des explorativen Lernens — der bereits betrachteten Lernobjekte, neue Formen der Kommunikation sowie individuelle Konfiguration der Oberfläche und der Präsentation des Lernmaterials. Eine Übersicht sowie weiterführende Beschreibungen der geplanten Innovationen für das virtuelle Lernen sind in Reiners et al. (2002) sowie Reiners et al. (2003) zu finden.

2.3.2 Aufbau und Transformation des Lernmaterials

Die wesentliche Technologie hinter SmartFrame ist Apache Cocoon, welches ein Servlet mit erheblicher Funktionalität im Bereich der XML-Transformation mittels XSL ist und dementsprechend prädestiniert für den Web-basierten Einsatz zur Darstellung von XML-Inhalten zu sein scheint. Der Transformationsprozess innerhalb von Cocoon basiert auf einem Pipelineprinzip, das erhebliche Flexibilität bei der Generierung von HTML, PDF oder WML (Wireless Markup Language) bietet. Das Pipelineprinzip in Bezug auf die Verwendung innerhalb von SmartFrame wird weiter unten näher erläutert.

Ein weiterer Vorzug von Cocoon ist die Einbindung der XSP-Technologie (eXtensible Server Pages), welche die direkte Integration von Java-Code in den Generierungsprozess der entsprechenden Ausgabeformate ermöglicht. Eine XSP-Seite zeichnet sich syntaktisch dadurch aus, dass eine Kapselung des Inhalts mittels eines <xsp:page>-Tags als Wurzelelement des enthaltenen XML-Dokuments erfolgt. Für die Generierung des Eingabestromes und die Ausführung des potentiell enthaltenen Java-Codes ist ein spezieller Generator erforderlich, der Serverpages-Generator. Für den Inhalt von XSP-Seiten besteht die Möglichkeit, Tag-Libraries zu definieren, die beim Generierungsprozess vorgegebene XSL-Transformationen anstoßen, beziehungsweise zur Ausführung von Java-Code genutzt werden können. In SmartFrame wird diese Technologie hauptsächlich zur Einbindung der Datenbank mit den Meta-Daten verwendet, so dass diese Meta-Daten direkt in den Transformationsprozess der Seiten mit einfließen können. Auf diese Weise ist es möglich, Lerneinheiten entsprechend eines bestimmten Anforderungsniveaus oder einer bestimmten Benutzerrolle zu generieren. Wie in Abschnitt 2.3 beschrieben, werden auch die Verlinkungen und Verweise von Lerneinheiten durch Extraktion der Datei-URL aus der Datenbank realisiert, wobei eine entsprechende Anfrage ebenfalls durch die Funktionalität einer Tag-Library durchgeführt wird.

Unter einer Pipeline ist ein Abschnitt innerhalb des Transformationsprozesses zu verstehen, der in sich abgeschlossen ist und in andere Pipelines integriert werden kann. Der folgende Auszug aus einer Cocoon-eigenen Konfigurationsdatei zeigt den von SmartFrame verwendeten Aufbau des XML-Bearbeitungsprozesses:

```
<!-- Hauptdokument -->

<map:match pattern="main.xsp">

  (1) <map:generate type="serverpages"

                     src="smartframe/website/main.xsp"/>

  (2)<map:transform type="cinclude"/>

  (3)<map:transform src="smartframe/website/pagebreak.xsl"/>

  (4)<map:transform src="smartframe/website/predesign.xsl"/>

  (5)<map:transform src="smartframe/website/design.xsl"/>

  (6)<map:serialize type="html">
  </map:match>

<!— Teildokumente —>

<map:match pattern="processed/**.xsp">

  (7)<map:generate type="serverpages" src="smartframe/{1}.xsp"/>

  (8)<map:transform src="smartframe/website/processed.xsl"/>

  (9)<map:transform type="cinclude"/>
  (10)<map:serialize type="xml"/>
  </map:match>
```

Der Vorgang der Cocoon-internen Transformation soll in Abbildung 6 verdeutlicht werden. Die in dem obigen Quelltext angegebenen Nummern korrespondieren mit den Ziffern in Abbildung 6.

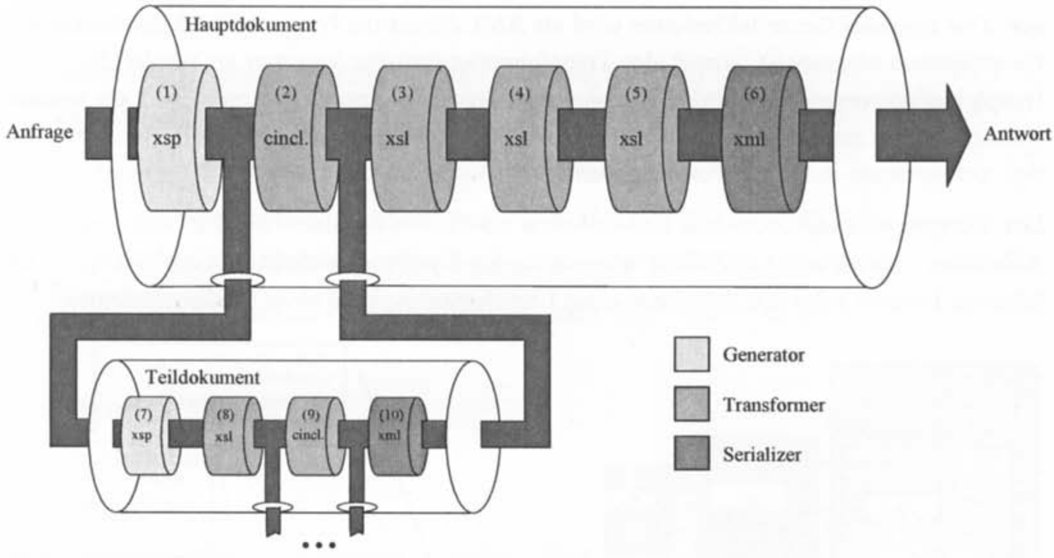


Abbildung 6: Cocoon-Pipelines innerhalb von SmartFrame

In der ersten Zeile wird lediglich der generelle Aufruf der im Folgenden beschriebenen Pipeline definiert. Der eigentliche Prozess beginnt stets mit der Generierung von SAX-Events[1] aus einer geeigneten Quelle (im Allgemeinen wird dies eine XML-Datei sein) mittels eines Generators (1) und endet mit der Serialisierung in eines der von Cocoon bereitgestellten Ausgabeformate (beispielsweise HTML oder XML) durch einen Serializer (6). In diesem Fall handelt es sich um einen Serverpages-Generator, der zum Ausführen von XSP-Seiten benötigt wird, und um einen HTML-Serializer, der aus dem bereits transformierten Dokument letztendlich eine HTML-Seite generiert. Zwischen diesen Initial- beziehungsweise Abschlusstransformationen kann eine beliebige Anzahl weiterer Umwandlungen mittels so genannter Transformer (2-5) erfolgen oder auch der Ausgabeteil einer anderen Pipeline eingebunden werden. In dem obigen Beispiel wird zunächst ein CInclude- Transformer (2) eingebunden, der für das Zusammenfügen mehrerer XML-Dokumente zuständig ist. An dieser Stelle kommt die zusätzliche Pipeline ins Spiel, die für das Bearbeiten von Teildokumenten verantwortlich ist. Auch diese beginnt mit einem Serverpages-Generator (7). Es folgt ein XSL-Transformer (8), der den Teil mit der gewünschten Sprache extrahiert. Für den Fall, dass noch weitere Dokumente eingebunden werden müssen, wurde ein CInclude-Transformer (9) in die Pipeline eingehängt. In diesem Fall würde die gleiche Pipeline für Teildokumente erneut rekursiv aufgerufen wer-

1 SAX (Simple API for XML) bietet eine Möglichkeit zum Parsen von XML-Dokumenten, welche auf Ereignissen basiert. Beispielsweise wird das Auftreten eines öffnenden oder schließenden XML-Elements durch ein entsprechendes Ereignis repräsentiert.

den. Das Ergebnis dieser Inklusionen wird als XML-Strom (in Form von SAX-Events) in die Hauptpipeline eingespeist, womit der Transformationsprozess von nun an wieder durch die Hauptpipeline vorgenommen wird. Es folgen mehrere XSL-Transformatoren (3-5), die weitere Umwandlungen mittels geeigneter XSL-Stylesheets durchführen und ein HTML-Serializer, der den Antwortstrom in HTML-Format generiert.

Der Transformationsprozess von Lerneinheiten innerhalb von SmartFrame ist schematisch in Abbildung 7 dargestellt. Die Ziffern entsprechen den Pipeline-Abschnitten in Abbildung 6. Das folgende Beispiel zeigt den XSP-Code eines Lernelementes sowie eines Medienelementes.[2]

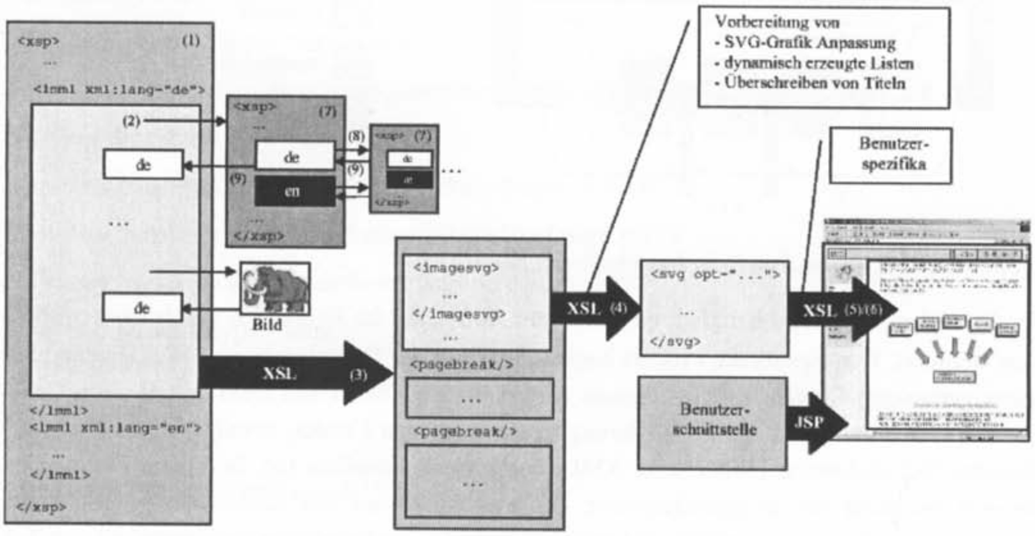


Abbildung 7: Transformationsprozess in SmartFrame

```
<xsp:page
  language="java"
  xmlns:xsp="http://apache.org/xsp"
  xmlns:smartframe="http://www.smartframe.de/xsl"
  xmlns:cinclude="http://apache.org/cocoon/include/1.0">
```

[2] An dieser Stelle soll auf eine ausführliche Beschreibung nicht näher eingegangen werden; vgl. hierzu Reiners et al. (2003). Allgemein ist der Aufbau wie folgt: Nach einem Header der XSP-Page mit entsprechenden Deklarationen von Namensräumen werden Funktionen z.B. zur Abfrage der Datenbank über den <smartframe:header/>-Tag eingebunden. Der Hauptteil besteht aus einer Festlegung des Typs (<smartframe:le>- bzw. <smartframe:me>-Tag) sowie deren Inhalt in LMML kodiert. Weitere Lernobjekte werden entsprechend mit dem <smartframe:link>-Tag referenziert.

```
 <smartframe:header/>
 <smartframe:le identifier="treiners_2002-10-25T09:42_01:00">
   <lmml xml:lang="de">
     <section title="Einleitung zur Simulationsvorlesung">
         <smartframe:link identifier="treiners_2002-10-25T09:35_01:00"/>
         <smartframe:link identifier="treiners_2002-10-25T09:25_01:00"/>
         <smartframe:link identifier="treiners_2002-10-25T09:37_01:00"/>
     </section>
   </lmml>
 </smartframe:le>
</xsp:page>
```

Der Aufruf einer Lerneinheit erfolgt über URL-Parameter (z.B. der Form http://www.smartframe.de/smartframe/main.xsp?identifier=treiners-2002-11-05T15: 50_01:00) beziehungsweise über ein <smartframe:include ref="…">-Tag innerhalb einer XSP-Seite. Im Falle des Aufrufs über einen Parameter wird der übergebene Wert innerhalb des XSP-Prozesses äquivalent zu einem ref-Attribut des include-Tags integriert. Zusätzlich enthält eine Lerneinheit in der Regel mehrere Sprachvarianten mit vergleichbarem Inhalt, welche wiederum in einem <lmml xml:lang="...">-Tag gekapselt sind. Für einen hohen Granularitätsgrad ist es notwendig, dass innerhalb des Transformationsprozesses die Möglichkeit besteht, andere Lerneinheiten oder auch Medienelemente in das aktuelle Dokument einzubinden. Eine derartige Inklusion wird über Taglib-Elemente angestoßen, die die Ausführung einer eigenen Pipeline veranlassen. Vor der eigentlichen Inklusion des Inhalts über einen CInclude-Transformer wird mittels eines XSL-Transformers der der voreingestellten Sprache entsprechende Teil der Datei extrahiert. Da inkludierte Lerneinheiten potentiell ebenfalls aus mehreren existierenden Lerneinheiten zusammengesetzt sein können, werden die entsprechenden Taglib-Elemente ebenfalls durch die referenzierten Einheiten ersetzt. Falls es sich bei den einzubindenden Dateien um Medienelemente, beispielsweise vom Typ SVG oder um statische Pixelgrafiken handelt, werden diese in entsprechende HTML-Elemente umgewandelt.

Nachdem die Lerneinheit auf diese Weise zusammengesetzt worden ist — wobei es sich nach dem ersten Schritt noch um ein Dokument in der Kodierung des LMML-Dialektes handelt — wird der generierte Strom an einen XSL-Transformer weiter gereicht. In diesem Schritt ist es möglich, das ganzheitliche Dokument vor der Darstellung noch in mehrere Teildokumente aufzusplitten, um die Lesbarkeit zu verbessern beziehungsweise eine Unübersichtlichkeit durch die Darstellung einer Lerneinheit über etliche Bildschirmseiten zu verhindern. Dazu ist es dem

Autor möglich, einzelne <pagebreak>-Tags zwischen semantisch abgeschlossenen Einheiten zu platzieren, wodurch an diesen Stellen jeweils ein Seitenwechsel impliziert wird. Beim ersten Aufruf einer Lerneinheit wird zunächst lediglich der Teil des Dokumentes bis zum ersten <pagebreak>-Tag dargestellt, wobei auch lediglich dieser Teil des Dokumentes an weitere XSL-Transformer weitergeleitet wird. Auf nachfolgende Seiten kann über Elemente der GUI zugegriffen werden.

Nach der Extraktion der relevanten Teile aus dem Gesamtstrom werden Umwandlungen vorgenommen, die als Voraussetzung für einen nächsten Transformationsprozess benötigt werden. Beispiele hierfür sind etwa die Sortierung von Literatureinträgen für eine spätere konsistente Darstellung oder das Überschreiben von ursprünglichen (direkt in das inkludierte Lernelement kodierten) Titeln. Möglich ist auch, dass über eine entsprechende Benutzer-konfiguration (gesetzt mittels der GUI-Einstellungsmöglichkeiten) gewisse Abschnitte nicht mehr an die eigentliche HTML-Transformation weitergereicht werden. So ist etwa denkbar, dass sich ein Lerner in der aktuellen Arbeitsphase lediglich einen Überblick über ein spezielles Kapitel verschaffen möchte und daraus resultierend keine Übungsaufgaben präsentiert bekommen möchte.

Abschließend wird die Transformation in HTML-Code vorgenommen. Dabei werden jeweils die LMML-Elemente (z.B. <motivation>) für den jeweiligen Anwender in konsistenter Weise in entsprechende HTML-Elemente überführt, was zu einer durchgängigen einheitlichen Darstellung führt.

2.3.3 SmartFrame: Prototyp für innovatives Lernen

SmartFrame stellt eine Umgebung dar, welche die zuvor beschriebenen Technologien zu einer virtuellen Lernumgebung vervollständigt. Hierzu zählen übliche Komponenten zur Verwaltung der Benutzer und deren Einstellungen, Navigationsmöglichkeiten sowie weiterführende Komponenten zur Kommunikation. Der aktuelle Prototyp setzt dabei auf einen Apache Tomcat-Server auf, bei dem mittels Servlets und JavaServer Pages entsprechend der Anforderungen die Oberfläche für eine Web-basierte Nutzung generiert werden. Dazu werden über ein Servlet Anweisungen ausgeführt und — je nach Konfiguration des Servlets — entsprechende JavaServer Pages ausgewählt beziehungsweise kombiniert, die wiederum Java-Code in Form von Java Beans oder Klassen enthalten können, um entsprechende Aktionen wie z.B. Datenbankabfragen zur individuellen Gestaltung der Seiten durchzuführen. Basierend auf einem Template mit Anweisungen hinsichtlich der Anordnung der Oberflächenelemente der virtuellen Lernumgebung werden auf diese Weise alle Komponenten inklusive der Interaktionsmöglichkeiten für den Lernenden dynamisch unter Berücksichtigung individueller

Einstellungen erzeugt. Ein schematischer Aufbau des Designs für diesen Prozess ist in Abbildung 8 gegeben.

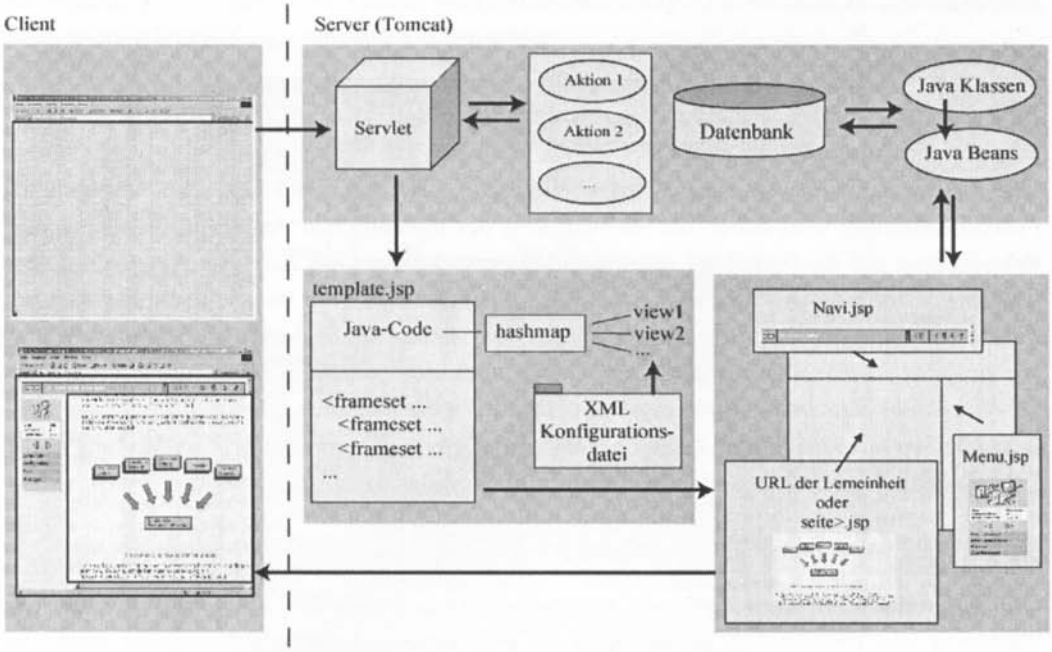


Abbildung 8: Servlet und JavaServer Pages zur Oberflächenerzeugung

Eine Entkopplung von dem Lernmaterial und den Inhalten an sich wird dadurch erreicht, dass zwar die Konfigurationseinstellungen des Lernenden mittels eines Session-Objektes an den Apache Cocoon-Prozess übergeben werden, aber ansonsten die Lernmaterialien nur mittels einer eindeutigen URI in einem Frame dargestellt werden. Dadurch wird eine Verwendung von externen Materialien weitestgehend unterstützt, ohne dass spezielle Anpassungen in der Logik der virtuellen Lernumgebung notwendig werden — insbesondere bei der Verwaltung von Kursen beziehungsweise Lernpfaden ist dies von hoher Relevanz.

3 Anwendungen

Eine der zentralen Forderungen im Bereich der wirtschaftswissenschaftlichen universitären Lehre ist die Schaffung einer Basis praxisgerechter Qualifikationen. Die Vermittlung von Handlungskompetenz darf jedoch keinesfalls mit der Vernachlässigung wissenschaftlicher Ansprüche an das Hochschulstudium einhergehen. Wesentliche Formen von Wissen, die erlangt werden sollten, sind das tätigkeitsspezifische Fachwissen im Sinne von Modellen und Theorien sowie das tätigkeitsfeldübergreifende Fachwissen, wie z.B. Problemlösefähigkeit und Transfervermögen. Letzteres kann differenziert werden in den Transfer von Fachwissen zur

Lösung von Problemen (Anwendungstransfer), den Transfer von theoretischem Wissen zur Lösung von Praxisproblemen (Praxistransfer) sowie den Transfer von Fachwissen auf andere Fragestellungen (Fachtransfer) (vgl. Schmitz (2001)).

3.1 Simulation

Im Rahmen der Umsetzung einer Lerneinheit zum Themenkomplex Simulation ist insbesondere auf die Ausgestaltung von didaktischen Ansätzen zur Förderung der Transferfähigkeit, hier im Speziellen Anwendungstransfer, Wert zu legen. Aufgrund der stark ausgeprägten Praxisorientierung wird neben der Aufbereitung des theoretischen Anteils insbesondere eine Fokussierung auf die Erstellung eines Simulationswerkzeuges mit Ausrichtung auf die Lehre und dem Vermitteln von Fachwissen sowie Zusammenhängen von unterschiedlichen Komponenten innerhalb der diskreten ereignisorientierten Simulation anvisiert. Im Folgenden gehen wir zunächst auf das von uns entwickelte Konzept sowie den danach entwickelten Prototypen für ein solches Simulationswerkzeug ein, präsentieren einen kurzen Überblick über eine mögliche Lektion mit interaktiven Komponenten und zeigen auf, inwiefern eine Einbindung in die Lehrmethodik Synchronized Blended Learning erfolgen kann.

3.1.1 Das Simulationswerkzeug SimTool

Die Integration des Themengebietes Simulation in eine virtuelle Lernumgebung dient der Förderung von forschendem beziehungsweise explorativem Lernen, insbesondere durch die interaktive und didaktische Aufbereitung. Dies wird durch die Verknüpfung einer Lerneinheit Simulation mit dem um didaktische Komponenten erweiterten Simulationswerkzeug SimTool ermöglicht. Hierdurch sollen bisherige Nachteile von existierenden Lernmaterialien, welche zumeist einen hohen theoretischen aber geringen praktischen Anteil haben, und von Simulationswerkzeugen ausgeglichen werden. Letztere sind in der Regel praxisorientiert und aufgrund der dort notwendigen Funktionsvielfalt und geringen didaktischen Ausrichtung zu kompliziert, als dass ein sinnvoller Einsatz innerhalb einer Lerneinheit angestrebt werden sollte. Beim Einsatz praxisorientierter Simulationswerkzeuge würde der Lernende einen wesentlichen Zeitaufwand allein dafür benötigen, die Software zu erlernen, statt sich mit der eigentlichen Funktionsweise von Simulationsmodellen auseinander zu setzen.

SimTool beinhaltet wesentliche Komponenten zur Unterstützung eines Einsatzes in der Lehre. Hierzu zählen die Konfigurationsmöglichkeiten der Benutzeroberfläche sowie auch die Selektion von Werkzeugen und Bausteinen in Abhängigkeit des aktuellen Lernkontextes. In Abbildung 9 ist die Benutzeroberfläche dargestellt; mittels Dateien zur Beschreibung von Lektionen können gezielt Elemente ausgeschaltet werden, um bestimmte Lernziele einer Lektion fokussiert zu vermitteln. Hierzu zählen neben dem Ausschalten von ganzen Komponenten wie der Buttonleiste auch einzelne Menüpunkte, z.B. Speichern, also auch

Felder innerhalb der Konfigurationsmenüs von Bausteinen. Beispielsweise könnte durch Ausschalten aller Komponenten außer der Buttons „Abspielen“ und „Pause“ sowie des Menüpunktes „Verlassen“ ein Modell geladen werden, welches der Lernende wie einen Film ohne weitere Interaktionsmöglichkeiten betrachtet.

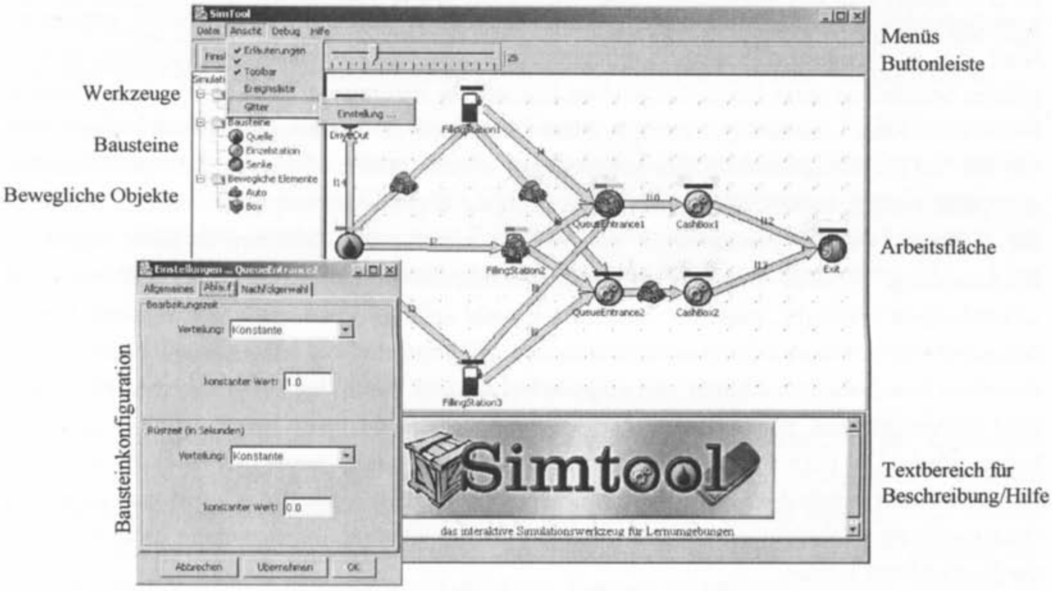


Abbildung 9: Oberfläche des Simulationswerkzeuges SimTooL

Ein weiterer wesentlicher Faktor bei der Unterstützung der Lehre ist die Möglichkeit, das Verhalten des Lernenden zu beobachten, zu analysieren und entsprechend darauf zu reagieren. Darüber hinaus sollte ein Mechanismus verwendet werden, der eine Unterstützung beziehungsweise Hilfestellung vornimmt. Hierzu wurde eine Skriptsprache sowie ein erweiterter Ereignis-Kontroll-Mechanismus entwickelt, mit denen sich Lektionen derart gestalten lassen, dass der Lernende auf der einen Seite frei an dem Modell arbeiten kann, auf der anderen Seite aber in Richtung der Zielsetzung geleitet wird. In Abhängigkeit des Schwierigkeitsgrades kann dies bedeuten, dass bei der falschen Eingabe von Parametern oder einer negativen Veränderung des Modells an sich sofort eine Rückmeldung in der Art „Dieser Parameter sollte nur zwischen den Werten 0 und 100 liegen“ erfolgt. Wie das System auf (falsche) Eingaben und — hier liegt der Schwerpunkt der didaktischen Unterstützung — auf Modellierungsfehler reagieren sollte, ist im Vorfeld in den Skripten und der Festlegung der Ereignisse zu bestimmen. Durch die Verknüpfung der Lerneinheit Simulation mit dem Simulationswerkzeug kann der Lernende daher sehr praxisnah und handlungsorientiert in die Logik der wissenschaftlichen Modellierung eingeführt werden. Anhand der Modelle können eigene Hypothesen evaluiert und unmittelbar die Konsequenz des eigenen Handelns beziehungsweise der Entscheidungsfindung erfahren werden. Darüber hinaus können eigenständige Modellierungen von gegebenen Problemsituationen zur Förderung der Abstraktionskompetenz des Lernenden beitragen.

Eine Lerneinheit kann derart gestaltet sein, dass den Lernenden per Skript zunächst ein grundlegendes Simulationsmodell mit entsprechenden Erläuterungen schrittweise präsentiert wird. Anschließend sind je nach Aufgabenstellung eventuell fehlende Komponenten zu integrieren beziehungsweise Einstellungen von Bausteinen anzupassen. In dem folgenden Beispiel sollen mit einem vollständig – bezüglich der im Modell verwendeten Bausteine – gegebenen Modell 500 Einheiten eines Produktes produziert werden, wobei einzig Attribute der Bausteine angepasst werden dürfen. Die Bausteinliste kann dabei entweder ausgeblendet sein, oder aber der Versuch, weitere Bausteine in das Modell einzubauen, resultiert in einer negativen Rückmeldung. Sollten durch den Ereignis-Kontroll-Mechanismus keine weiteren Einschränkungen vorliegen, kann der Lernende sowohl spielerisch als auch systematisch Veränderungen von Attributbelegungen durchführen. Dabei muss die Wertigkeit und Relevanz der einzelnen Komponenten erkannt und eingeordnet werden. Nach der Eingabe der Attributwerte wird veranschaulicht, wie sich das repräsentierte System unter den neuen Bedingungen verhalten würde. Die Lerneinheit gilt als erfolgreich abgeschlossen, wenn innerhalb der erlaubten Simulationszeit mindestens die geforderte Produktionsmenge erreicht wird, was zum einen mit einer positiven Rückmeldung und zum anderen mit Speicherung der erreichten Ziele innerhalb des Curriculums erfolgt.

Das Simulationswerkzeug ermöglicht eine Adaption an den Lernenden, sowohl hinsichtlich einer Senkung des Schwierigkeitsgrades bei Erkennen von Problemen, z.B. wenn der Lernende viele Fehler verursacht oder die Hilfe mehrfach benötigt, als auch eine Steigerung im Falle eines zu „schnellen" Durcharbeitens von Lerneinheiten ohne auftretende Probleme. Zum Beispiel wird in einer Lektion von SimTool die Modifikation von Attributwerten im Sinne der Aufgabenstellung verlangt. Nachdem der Lernende die Einstellungen vorgenommen hat, erfolgt eine Überprüfung auf Korrektheit und gegebenenfalls eine Freischaltung der nächsten Aufgabe. Bei Fehlern bekommt der Lernende die Möglichkeit zur Verbesserung; nach einer vorgegebenen Anzahl von Fehlversuchen wird automatisch eine weiterführende Hilfe angeboten, bzw. wenn es zu weiteren Fehlversuchen kommt, wird die Lösung textuell präsentiert, die dann in das Modell übertragen werden muss. Bei weiteren Fehlern wird darüber hinaus die Lösung automatisch generiert und die Aufgabe beendet. So bekommt der Lernende sukzessiv eine Vereinfachung der Aufgabe, wobei jedoch auch gleichzeitig darauf geachtet wird, dass ein Fortschritt bei der Bearbeitung der gesamten Aufgaben erfolgt. Sollte die Aufgabe als nicht gelöst gewertet werden, so ist systemseitig zu einem späteren Zeitpunkt eine strukturell ähnliche Aufgabe zwecks Wiederholung des Gelernten zu wählen. Aufgrund dieser adaptiven Fähigkeit ist das Simulationswerkzeug ein geeignetes Arbeitsmittel, um beim

Lernenden das Verständnis für die sich dynamisch verändernden Vorgänge und Werte zu fördern und gleichzeitig durch geeignete Rückmeldungen und Anpassungen den Lernenden direkt zu motivieren. Für die Autoren von Lerneinheiten bietet das Simulationswerkzeug zum einen hochgradige Konfigurationsmöglichkeiten, da sämtliche Interaktionsoptionen des Lernenden einzeln festgelegt werden können, sowie Rückmeldungen, Hilfestellungen und Adaptivität an den Lernenden modifizierbar sind, zum anderen aber auch eine Komplexität, die ohne die Verwendung von noch umzusetzenden Autorentools nur für kleinere Lerneinheiten geeignet ist.

Ebenso denkbar ist die eigenständige Modellierung von vorgegebenen Problemstellungen durch den Lernenden, wobei dieser durch intelligente Rückmeldungen der Lernumgebung unterstützt wird. Innerhalb von Klausuren oder Tests können Aufgaben mit dem Simulationswerkzeug bearbeitet werden, wie zum Beispiel die Ermittlung geeigneter Parameter zur Erzielung vorgegebener Ergebnisse oder dem Redesign von Simulationsmodellen.

3.1.2 Der Einsatz von Simulationsmodellen innerhalb von Synchronized Blended Learning

Virtuelle Lernumgebungen werden im Wesentlichen für das Selbststudium der Lernenden angeboten, bei dem sie auf sich allein gestellt die angebotenen Lernmaterialien bearbeiten können. Zusätzliche Komponenten unterstützen die Gruppenarbeit und fördern damit sowohl die Kommunikationsfähigkeit als auch die Teamfähigkeit, sind aber im Umfang deutlich unterhalb von Präsenzveranstaltungen mit direktem Kontakt zu anderen Personen anzusiedeln. Die von uns konzipierte virtuelle Lernumgebung SmartFrame sieht darüber hinaus weitergehende Verwendungsszenarien vor, bei denen z.B. ein Tutor die virtuelle Lernumgebung zur Präsentation einsetzt und somit den Lernenden ein einheitliches Medium innerhalb der Lehre bietet, wobei hier entsprechende Darstellungsmodi für die Lerneinheiten existieren müssen.

In diesem Kontext ist das von Frank et al. (2003) eingeführte Synchronized Blended Learning zu sehen, welches eine Erweiterung des didaktischen Konzeptes des Blended Learning darstellt. Blended Learning — zum größten Teil von Unternehmungen als reiner Marketingbegriff verwendet — ist die Kombination von verschiedenen Lehrformen, wie z.B. Präsenzveranstaltung oder virtuelles Seminar, und didaktischen Konzepten, die es den Lernenden erlaubt, die von ihnen bevorzugte Lernform selbst auszuwählen. Auf diese Weise ist es möglich, insgesamt einen höheren Lernerfolg zu erzielen. Synchronized Blended Learning integriert zusätzlich eine zeitnahe adaptive Synchronisation zwischen den verschiedenen Lehrformen und eröffnet den Lernenden auf diese Weise über die reine Auswahlmöglichkeit hinaus das Arbeiten in einer einheitlichen Lernumgebung ohne Medienwechsel. Das Prinzip des Synchronized Blended Learning soll im Folgenden anhand von zwei einfachen Beispielen aus dem Gebiet der Simulation unter Verwendung des Simulationswerkzeuges SimTool genauer erläutert werden; vgl. Abbildung 10.

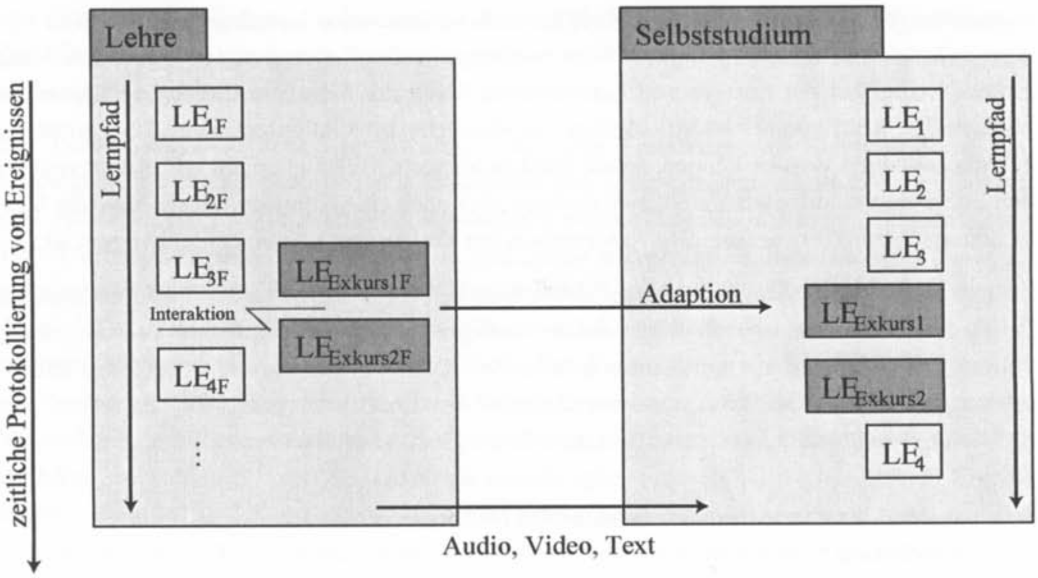


Abbildung 10: Synchronized Blended Learning am Beispiel des Abgleichs von Präsenzveranstaltung mit dem Selbststudium

Im ersten Beispiel wird die Verwendung der virtuellen Lernumgebung innerhalb der Präsenzveranstaltung angenommen, wobei die Lerneinheiten LE1F in einer Foliendarstellung vorliegen. Nach der Folie LE3F wird aus dem Auditorium eine Frage gestellt (Interaktion), die der Tutor durch die Präsentation eines Exkurses (Lerneinheiten Exkurs1F sowie Exkurs2F) beantwortet. Im Anschluss setzt er die Veranstaltung mit den zuvor geplanten Lerneinheiten fort. Lernende, die aus verschiedenen Gründen nicht an der Veranstaltung teilnehmen können und auf ein Selbststudium angewiesen sind, würden diesen Exkurs und die dazugehörigen Erklärungen nicht kennen und gegenüber den anderen Lernenden benachteiligt sein. Aufgrund dessen sieht das Synchronized Blended Learning vor, dass zeitnah die im Exkurs verwendeten Lerneinheiten in den Lernpfad des Selbststudiums eingepflegt werden und somit allen Lernenden zur Verfügung stehen. Dieses kann durch eine reine Wiedergabe der Folien, durch weiteres Lernmaterial, das im Vorfeld bereits zu den entsprechenden Folien definiert wurde oder durch Audio- beziehungsweise Videosequenzen aus der Präsenzveranstaltung ergänzt werden.

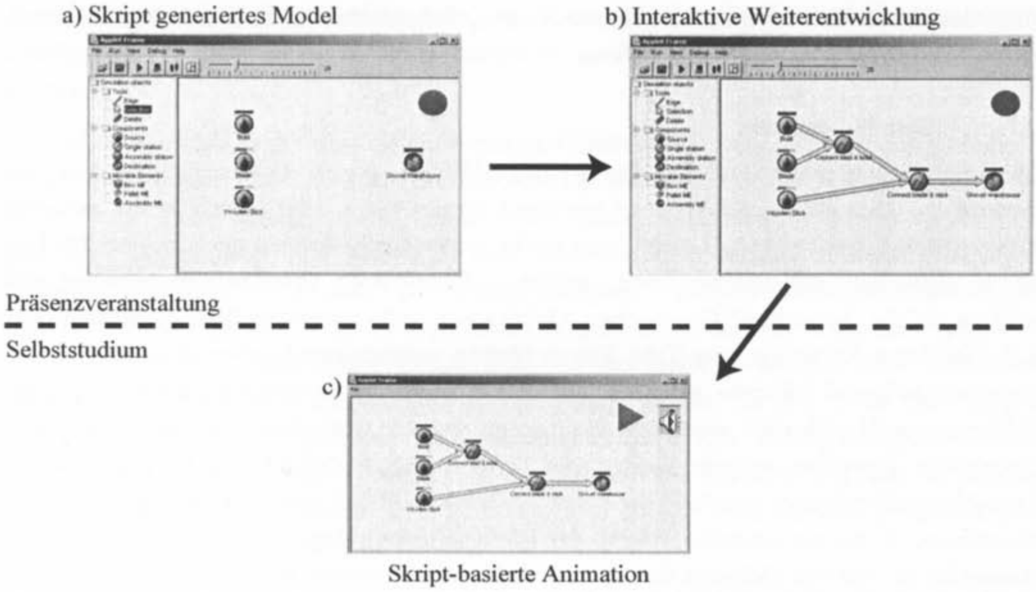


Abbildung 11: *Aufzeichnung der Modellierung eines Simulationsmodells in der Präsenzveranstaltung (a-b) und deren anschließende Übertragung in das Selbststudium mittels eines Skriptes (c)*

Eine weitere Form des Synchronized Blended Learning ist in Abbildung 11 dargestellt. Hier wird in einer traditionellen Präsenzveranstaltung das in die Lernumgebung integrierte Simulationswerkzeug SimTool genutzt, um das Design von Simulationsmodellen und die Analyse von Simulationsvorgängen zu demonstrieren. Dabei wird das Modell ausgehend von einigen vorgegebenen Bausteinen gemeinsam von Lehrenden und Lernenden interaktiv weiterentwickelt. Hierzu sind im Vorfeld entsprechende Skripte zu definieren, die Teile des Modells erstellen. Die vom Tutor durchgeführten Schritte sind zu protokollieren, wobei die Protokollierung ebenso die zeitliche Komponente wie auch durchgeführte Korrekturen zu unterstützen hat. Zusätzliche Audio- und Videoaufnahmen können als Ergänzung verwendet werden, um so z.B. die einzelnen Schritte ausführlich zu dokumentieren. Diese Materialien können anschließend in vielfältiger Art und Weise in andere Formen der Lehre übertragen werden. Zum Beispiel kann die gespeicherte Folge von Schritten in einer Art Abspielmodus zusammen mit Audiosequenzen im Selbststudium aufgezeigt werden. Durch dieses Vorgehen erscheint die Modellierung transparenter, da direkte Erklärungen des Tutors beziehungsweise Diskussionen mit den Lernenden innerhalb der Präsenzveranstaltung in das Selbststudium übernommen werden. Weiterhin können die Lernenden im Anschluss an die Präsenzveranstaltung die interaktiven Inhalte zum Zwecke eines Internalisierungsprozesses erneut ohne Informationsverlust betrachten. Weitere Ansätze sind z.B. zum Training und zur Vertiefung des Gelernten vorgegebene geführte Touren, die mit den in der Präsenzveranstaltung entwickelten Modellen synchronisiert sind.

3.2 Meta-Heuristiken

Meta-Heuristiken, als zeitgemäße und wichtige Werkzeuge zur Optimierung, stellen eine wesentliche Thematik innerhalb des Operations Research dar, sind jedoch in der aktuellen universitären Lehre sowie in Lehrbüchern weder ausreichend vertreten noch in einer für Lernende didaktisch aufbereiteten Form verfügbar. Existierende Umsetzungen beziehen sich zumeist auf die „bekannten“ Genetischen Algorithmen; andere wichtige Meta-Heuristiken wie z.B. Simulated Annealing oder Tabu Search bleiben weitestgehend unberücksichtigt. Daraus und auch aufgrund fehlender guter Fallstudien im Sinne von praktischen Anwendungsmöglichkeiten ergibt sich eine unzureichende Vermittlung der wesentlichen Vorteile von Meta-Heuristiken gegenüber anderen Formen der Optimierung. Fachdidaktisch betrachtet sollten Meta-Heuristiken jedoch innerhalb der Lehre im Bereich OR/MS eine weiterreichende Position einnehmen, da sie ein zentrales Prinzip der Lösungsfindung darstellen, in vielerlei Hinsicht Analogien zu weiteren Gebieten des Operations Research besitzen und in der Darstellung der Zusammenhänge einen hohen Grad an Veranschaulichung haben. Beispielsweise operiert der Simplex-Algorithmus (vgl. Domschke und Drexl (2002) sowie Domschke et al. (2002)) zur Lösung von linearen Optimierungsproblemen durch das gezielte Traversieren eines den Lösungsraum beschreibenden Polyeders. Analog gehen z.B. Verfahren zur Lösung von Transportproblemen vor, bei denen im Anschluss an eine Eröffnungsheuristik zur Ermittlung einer zulässigen Lösung Verbesserungsverfahren wie die MODI-Methode (vgl. z.B. Domschke (1995)) auf stark strukturierten Lösungsräumen zum Einsatz kommen. Für die Studierenden ergeben sich durch die Behandlung von Meta-Heuristiken aufgrund von Analogien der Prinzipien zur Lösung von Optimierungsverfahren aus fachdidaktischer Sicht weitreichende Transfer- und Anwendungsmöglichkeiten. Daher bietet sich eine inhaltliche Behandlung der lokalen Suche sowie von Meta-Heuristiken im Rahmen der Lehre an. In diesem Beitrag stellen wir einen einführenden Überblick zu unserem Konzept vor; für eine ausführlichere Darstellung vgl. Reiners und Voß (2003).

3.2.1 Vermittlung von Grundlagen

Obwohl Meta-Heuristiken unter methodischen Gesichtspunkten vielfältige Möglichkeiten der anschaulichen Darstellung auf verschiedenen Abstraktionsebenen bieten, werden didaktische Prinzipien und Konzepte bei ihrer Vermittlung bis jetzt nur wenig eingesetzt. Zurzeit wird noch auf statische untereinander verlinkte Texte oder auf Sammlungen interaktiver Applets ohne theoretische Ausführungen zurückgegriffen. Weitere Darstellungsformen von Optimierungsergebnissen sind z.B. bei Jones (1995) zu finden. Eine grundlegende Zielsetzung des dargestellten Lehrangebotes besteht darin, dass Lernende ohne Vorkenntnisse später eigenständig Meta-Heuristiken zum Lösen komplexer praktischer Problemstellungen einsetzen können.

Eine Meta-Heuristik ist definiert als ein iterativer Generationsprozess, der eine untergeordnete Heuristik steuert und eine Methode darstellt, ein Optimierungsproblem in adäquater Zeit zu lösen. Jedes Optimierungsproblem besitzt einen speziellen Lösungsraum, der hinsichtlich gegebener Restriktionen alle zulässigen (und gegebenenfalls darüber hinausgehende nicht zulässige Lösungen, wie sie sich z.B. vermöge geeigneter Problemrelaxationen ergeben) enthält. Da der Lösungsraum in den meisten Fällen zu groß ist, ist das Auffinden einer optimalen Lösung durch eine vollständige Enumeration — das heißt Evaluation aller möglichen Lösungen — häufig nicht möglich, insbesondere innerhalb der für die Praxis notwendigen Zeitrestriktionen. Daher grenzen Meta-Heuristiken durch die intelligente Kombination verschiedener Methoden den Lösungsraum auf bestimmte Bereiche ein und verwenden Lernstrategien, um Informationen zu strukturieren und effektiv optimale oder zumindest gute Lösungen zu finden; vgl. Voß et al. (1999).

Im Folgenden zeigen wir anhand von statischen Bildern eine Möglichkeit auf, wie eine interaktive Erklärung der lokalen Suche aussehen kann. Neben einer grundlegenden Erläuterung der Begriffe der lokalen Suche – hierzu zählen unter anderem Problem, Lösung, Nachbar, Zug sowie Nachbarschaft mit den jeweiligen Attributen – sind die Prinzipien anhand von interaktiven Beispielen zu verdeutlichen, in denen insbesondere die Transformationsschritte von der Problemstellung zur Lösung aufgezeigt werden. Abbildung 12 visualisiert das Handlungsreisendenproblem mit Hilfe eines Graphen.[3] Der Lernende kann mittels der Kontrollleiste Schritt für Schritt den Graphen in einen Lösungsvektor (hier einen Permutationsvektor) überführen, wobei zusätzliche Informationen und Erklärungen angezeigt werden. Abbildung 12 zeigt darüber hinaus, inwieweit der Lernende mittels der Funktionen  (für Legende) und  (für Informationen) Zusatzinformationen und Erklärungen einblenden kann, die wiederum interaktiv sein können. Hier hat der Lernende ein Element des Lösungsvektors gewählt, um weitere Informationen zu erhalten. In dem neuen Fenster wird beim Anwählen der Knotennummer eine zusätzliche Hilfe geöffnet, die weitere Erklärungen bietet, aber auch visuell einen Zusammenhang zur eigentlichen Problemstellung beinhaltet.

Die aus der Problemstellung erzeugten Lösungen werden unter anderem als Initiallösungen für lokale Suchverfahren verwendet. Bei der im Folgenden verwendeten Meta-Heuristik Steepest

3 Beim Problem des Handlungsreisenden ist eine kürzeste Tour (Rundreise oder geschlossener Weg; Minimierungsproblem) innerhalb eines gegebenen Graphen zu finden, wobei alle Städte bzw. Knoten genau einmal zu besuchen sind.

Descent wird ausgehend von der Initiallösung eine Nachbarschaft gebildet, aus der dann wiederum eine hinsichtlich der Zielfunktion beste Lösung für die nächste Iteration als Initiallösung ausgewählt wird.

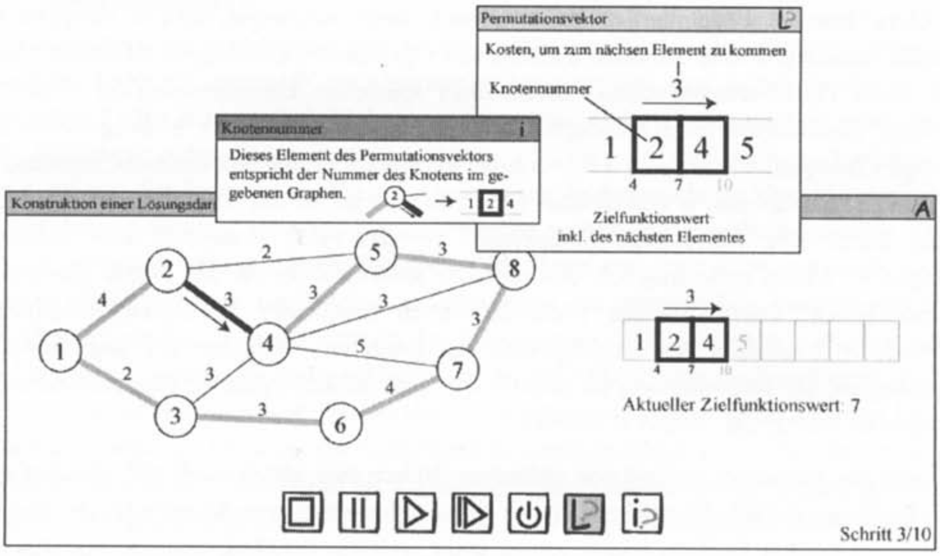


Abbildung 12: Konstruktion einer Lösungsdarstellung: Permutationsvektor

Eine Nachbarschaft ist durch die Lösungen definiert, welche von definierten Zügen aus der aktuellen Initiallösung gebildet werden können. Abbildung 13 zeigt an einem Beispiel die für die Durchführung eines Zuges entsprechende Modifikation eines Lösungsvektors, aus der eine neue Lösung (Nachbar) hervorgeht. Bei dem Zug handelt es sich um die Vertauschung von zwei Elementen des Vektors, das heißt, dass die Städte in dem Beispiel in vertauschter Reihenfolge besucht werden. Auf der rechten Seite in der Abbildung ist die (Teil-)Nachbarschaft zu sehen, die ausgehend von der Initiallösung durch die durchgeführten Züge aufgespannt wurde.

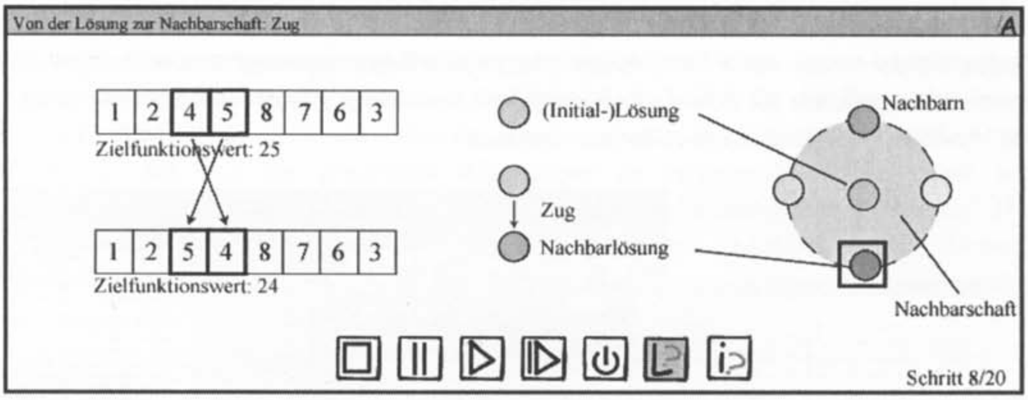


Abbildung 13: Konstruktion einer Nachbarschaft

Zuletzt wird die Traversierung des Lösungsraumes durch die Auswahl des – hinsichtlich eines Optimierungskriteriums – besten Nachbarn beschrieben, welcher in der folgenden Iteration als neue Ausgangslösung verwendet wird. Abbildung 14 zeigt neben der aus der vorherigen Abbildung bekannten Darstellung außerdem eine Abbildung des gesamten Lösungsraums sowie eine schematische Übersicht der Entwicklung in Form eines rollenden Balles im Lösungsraum. Der Farbwert der einzelnen Lösungen korrespondiert mit dem Zielfunktionswert: je kleiner der Zielfunktionswert, desto dunkler ist der Farbwert gewählt worden.

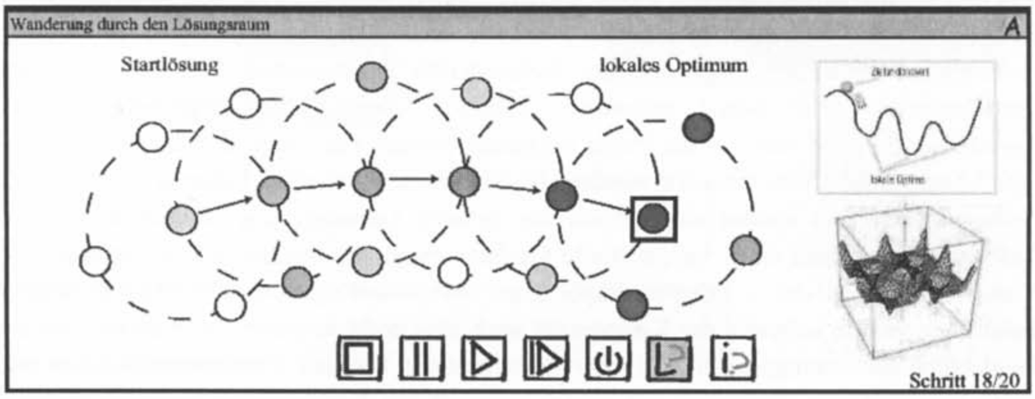


Abbildung 14: Durchlaufen des Lösungsraumes

Im Anschluss an die Vermittlung der Grundlagen und entsprechende Übung mit den interaktiven Lernmaterialien sind die eigentlichen Meta-Heuristiken zu erklären. Hierzu wird ein Applet entsprechend Abbildung 15 verwendet. Auf der linken Seite sind die aktuelle Lösung sowie eine Historie der Lösungen und eine Veranschaulichung der entsprechenden

Zielfunktionswerte zu sehen. In der Mitte ist eine Darstellung der aktuellen Nachbarschaft zu finden, auf der rechten Seite die Problemstellung sowie Erklärungen und Beschreibungen. Der Lernende kann Schritt für Schritt den Ablauf der Meta-Heuristik nachvollziehen und dadurch ein Verständnis für die Zusammenhänge bekommen.

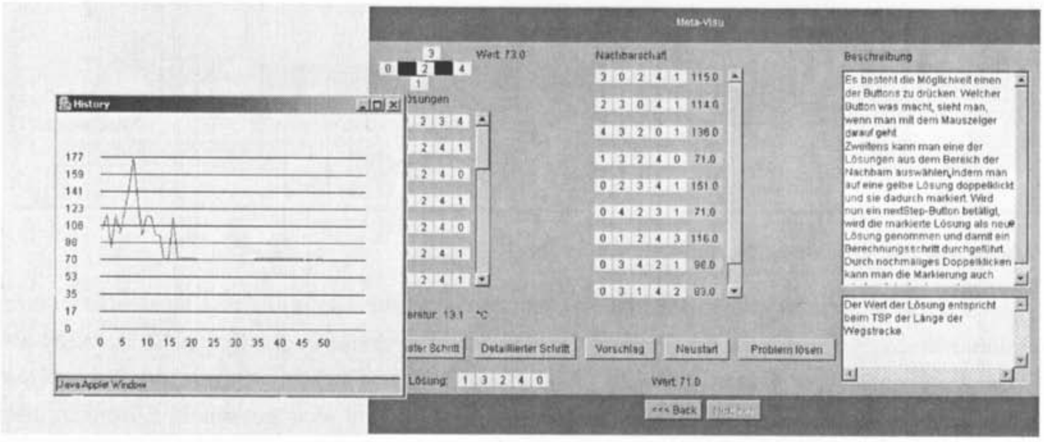


Abbildung 15: MetaVisu — Applet zur Darstellung von Abläufen bei Meta-Heuristiken

3.2.2 Meta-Heuristiken: Von der Technik zur Anwendung

Um eine Intensivierung der Materie zu erreichen, sollten Lernende erworbenes Wissen in einer komplexeren und in weiten Teilen realistischen Problemstellung – im Gegensatz zu den stark vereinfachten Beispielen innerhalb des Lernmaterials – anwenden und dabei auf den grundlegenden Prinzipien aufbauen. Hierzu können Problemstellungen von Forschungsprojekten oder aus der Praxis eingesetzt werden. Die Lernenden entwickeln dabei eine Lösungsmöglichkeit unter Verwendung bestehender und in weiten Teilen konfigurierbarer Softwarepakete und werden dabei durch die virtuelle Lernumgebung unterstützt. Bei der Software können zum einen kommerzielle Produkte eingesetzt werden, die im Umfang eine Unterstützung von der – zumeist graphischen – Problemeingabe bis zu dessen Lösung beinhalten, jedoch aufgrund der Komplexität auch eine problemspezifische beziehungsweise hinsichtlich der Lösungsqualität gegebene Einschränkung besitzen. Zum anderen können frei verfügbare Softwarebibliotheken mit wieder verwendbaren Komponenten eingesetzt werden, die jedoch mit dem Nachteil behaftet sind, dass gegebenenfalls Programmierkenntnisse vorauszusetzen sind. Um eine weitgehende Integration in unsere virtuelle Lernumgebung ohne Medien- oder Technologiebrüche zu gewährleisten, basiert das in dem Projekt umzusetzende Konzept auf dem zweiten Ansatz unter Verwendung von HotFrame (Heuristic OpTimization FRAMEwork); vgl. Fink und Voß (2002). Für die Konfiguration der wesentlichen

Komponenten der Meta-Heuristiken wird eine Browser-basierte Benutzerschnittstelle verwendet. Dort vorgenommene Einstellungen werden in einem (semi-) automatisch generierten Source-Code mit noch zu ergänzenden Todo-Teilen für die Lernenden transformiert. Schließlich stellt die virtuelle Lernumgebung ein Experimentierumfeld zur Verfügung, auf dem die generierten Algorithmen an diversen Problemstellungen mit verschiedenen Parametereinstellungen der Meta-Heuristiken evaluiert werden können. Die Resultate werden gesammelt und mittels statistischer Methoden und Darstellungsmöglichkeiten analysiert; vgl. auch Abbildung 16 für eine schematische Übersicht über den Prozess des eigenständigen Anwendens.

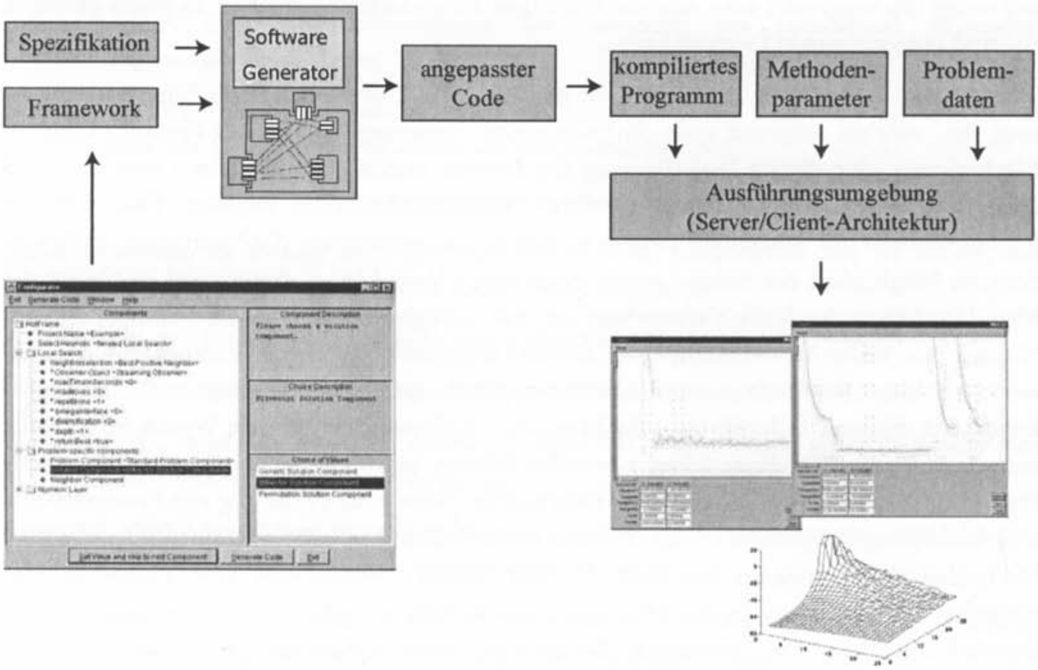


Abbildung 16: HotFrame - Anwendung mittels Software-Generator

4 Konklusion

In diesem Beitrag wurde ein Konzept für eine virtuelle Lernumgebung vorgestellt, bei dem neben einer Umsetzung von Lernmaterialien zu den Fachgebieten Simulation und — ein bisher stark vernachlässigter Bereich des Operations Research — Meta-Heuristiken insbesondere neue (innovative) Formen für die didaktische Darstellung erarbeitet werden. Hierzu zählen die starke Fokussierung auf die Berücksichtigung von individuellen Konfigurationsoptionen, so dass die Lernenden ihr Lernerlebnis in gewisser Weise selbst gestalten und verbessern können.

Dieses wird sowohl in der virtuellen Lernumgebung SmartFrame als auch in dem modularen Design der XML-basierten Lernmaterialien unter Verwendung der Spezifikationen LMML sowie LOM für die Meta-Daten in einem gesamtheitlichen Rahmen umgesetzt. Hierbei finden ausschließlich frei verfügbare Softwarepakete Anwendung. Auf der didaktischen Seite ist sowohl die Anpassung des Lernmaterials, das heißt die Möglichkeit des Lernenden darüber zu entscheiden, welche Komponenten dargestellt werden sollen, als auch der Einsatz von SmartBars zur Verbesserung der Informationsdarstellung von Lernmaterialien hervorzuheben. Als besonders vorteilhaft erweist sich in diesem Zusammenhang, dass auf diese Weise dem weit verbreiteten Problem, dass didaktische Konzepte der Technologie untergeordnet werden, indem nur das umgesetzt wird, was die bisherigen Möglichkeiten hergeben, in einem gewissen Ausmaß entgegengewirkt werden kann.

Das Simulationswerkzeug SimTool stellt ein interaktives sowie adaptives Simulationswerkzeug dar, welches aufgrund einer Skript-basierten Steuerung sowie eines Ereignis-Kontroll-Mechanismus eine direkte Beeinflussung des Lernens erzielt, das heißt den Lernenden wird ohne Einschränkung ein Rahmen für ihre Aktionsmöglichkeiten gegeben. Entsprechende Rückmeldungen zur Bewertung fördern in diesem Zusammenhang die Motivation. Insbesondere die Möglichkeit des freien Lernens ohne starke Vorgaben — dieses wird im Gebiet der Meta-Heuristiken durch die Verwendung von frei verfügbaren Software-Bibliotheken und der Nutzung von realen Problemstellungen ebenfalls umgesetzt — soll eine Verknüpfung des wissenschaftlichen Anspruchs eines Hochschulstudiums mit der Vermittlung von Handlungskompetenz erfolgen. Dahinter steht die Idee, dass Studierende erworbenes Wissen im Rahmen eines bestimmten Forschungsfeldes anwenden können, um nachgestellte Fragestellungen der Praxis oder der aktuellen Forschung zu untersuchen. Gerade die Erfahrung von Praxisrelevanz und Anwendungsmöglichkeiten des erlernten Wissens kann hochgradig zu einer Steigerung der Motivation der Lernenden beitragen. Darüber hinaus befindet sich eine Einbettung der virtuellen Lernumgebung in das lehr- und lernformübergreifende Konzept des Synchronized Blended Learning in der Umsetzung, das ebenfalls motivationssteigernde und damit lernförderliche Wirkung aufweisen kann.

Danksagung

Teile dieser Arbeit wurden durch das BMBF (Bundesministerium für Bildung und Forschung (2003)) in dem Programm „Neue Medien in der Bildung“ (08NM094D) unterstützt.

Literatur:

Advanced Distributed Learning Initiative (2003, Februar) SCORM. http://www.adlnet.org. Letzter Abruf: 01.02.2003.

Apache Software Foundation (2003a, Februar) Apache Cocoon. http://xml.apache.org/cocoon. Letzter Abruf: 01.02.2003.

Apache Software Foundation (2003b, Februar) Apache Xindice. http://xml.apache.org/xindice. Letzter Abruf: 01.02.2003.

Apache Software Foundation (2003c, Februar) Jakarta Tomcat. http://Jakarta.apache.org/tomcat/index.html. Letzter Abruf: 01.02.2003.

ARIADNE Foundation (2003, Februar) ARIADNE Project Home Page, http://ariadne.unil.eh. Letzter Abruf: 01.02.2003.

Ballstaedt, S.-P. (1997) Wissensvermittlung: Die Gestaltung von Lernmaterial. Weinheim: Beltz/Psychologie Verlags Union.

Bell, P.C. (2000) Teaching business statistics with Microsoft Excel. INFORMS Transactions on Education 1(1), S. 18-26. http://ite.informs.org/VollNol/bell/bell.html.

Blackboard (2003, Februar) Blackboard. http://www.blackboard.com. Letzter Abruf: 01.02.2003.

Bundesministerium für Bildung und Forschung (2003, Februar) Bundesministerium für Bildung und Forschung BmBF. http://www.bmbf.de. Letzter Abruf: 01.02.2003.

Domschke, W. (1995) Logistik: Transport (4. Aufl.). München: Oldenbourg.

Domschke, W.; Drexl, A. (2002) Einführung in Operations Research (5. Aufl.). Berlin: Springer.

Domschke, W.; Drexl, A.; Klein, R.; Scholl, A.; Voß, S. (2002) Übungen und Fallbeispiele zum Operations Research (4. Aufl.). Berlin: Springer.

Evans, J.R., (2000) Spreadsheets as a tool for teaching simulation. INFORMS Transactions on Education 1(1), S. 27-37. http://ite.informs.org/VollNol/evans/evans.html.

Fink, A.; Voß, S. (2002) HotFrame: A Heuristic Optimization Framework. In S. Voß, D.L. Woodruff (Hrsg.), Optimization Software Class Libraries, S. 81-154. Boston: Kluwer.

Frank, C. (2001) Studentische Erwartungen an computergestützte Lernumgebungen: Eine empirische Erhebung. Vorgetragen auf dem Symposium Operations Research, Duisburg.

Frank, C.; Kassanke, S.; Suhl, L. (2002) Meeting students' expectations and realizing pedagogical goals within the development of a virtual learning environment. In Proceedings of Elearn World Conference on E-Learning in Corporate Government, Healthcare & Higher Education, Montreal, Canada, S. 1124-1129.

Frank, C.; Reiners, T.; Suhl, L.; Voß, S. (2003) A Conceptual Framework for Synchronized Blended Learning on the Web. Arbeitspapier, Universität Paderborn.

Fraunhofer Institut für Graphische Datenverarbeitung (2003, Februar) CMS-W3-Kursverwaltungssystem. http://www.egd.igd.fhg.de/IGD/Abteilungen/ARl/Produkte_ARl/CMS-W3. Letzter Abruf: 01.02.2003.

Hazemi, R.; Halles, S.; Wilbur, S. (Hrsg.) (1998) The Digital University: Reinventing the Academy. London: Springer.

IEEE Learning Technology Standards Committee (2003, Februar) Standard for Information Technology: Education and Training Systems - Learning Objects and Metadata. http://ltsc.ieee.org/wgl2. Letzter Abruf: 01.02.2003.

IMS Global Learning Consortium (2003a, Februar) IMS Question & Test Interoperability Specification. http://www.imsproject.org/question. Letzter Abruf: 01.02.2003.

IMS Global Learning Consortium (2003b, Februar) IMS Implementation Handbooks. http://www.imsproject.org/implementationhandbook/index.html. Letzter Abruf: 01.02.2003.

INFORMS Transactions on Education (2003, Februar) http://ite.informs.org. Letzter Abruf: 01.02.2003.

International Federation of Operational Research Societies (2003, Februar) tutORial. http://www.ifors.ms.unimelb.edu.au/tutorial/index.html. Letzter Abruf: 01.02.2003.

Jones, C.V. (1995) Visualization and Optimization. Boston: Kluwer.

Kassanke, S.; Suhl, L. (2000) OR-World: Using Learning Objects in a Hypermedia Learning Environment. In Proceedings of EdMedia 2000 - World Conference on Educational Multimedia. Hypermedia & Telecommunications, Montreal, Canada, S. 914-920.

Klein, B. (2000) Didaktisches Design hypermedialer Lernumgebungen: Die adaptive Lernumgebung 'incorps' zur Einführung in die Kognitionspsychologie. Marburg: Tectum.

MySQL (2003, Februar) MySQL: The World's Most Popular Open Source Database. http://www.mysql.com. Letzter Abruf: 01.02.2003.

OR-World (2003, Februar) http://www.or-world.com. Letzter Abruf: 01.02.2003.

Reiners, T.; Reiß, D.; Voß, S. (2002) Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts. In Proceedings of the World Congress Networked Learning in a Global Environment. Challenges and Solutions for Virtual Education (NL 2002), Volume #100029-03-TR,-026, Millet Alberta, S. 1-7. ICSC-Naiso Academic Press.

Reiners, T.; Reiß, D.; Voß, S. (2003) XML-basierte Kodierung von Lernobjekten. Arbeitsbericht, Technische Universität Braunschweig.

Reiners, T.; Voß, S. (2003) Some Thoughts on how to Educate OR/MS. Arbeitsbericht, Universität Hamburg.

Schmitz, K. (2001) Virtualisierung von wirtschaftswissenschaftlichen Lehr- und Lernsituationen: Konzeption eines Application Frameworks. Wiesbaden: Deutscher Universitäts-Verlag.

Schulmeister, R. (2000) Didaktische Aspekte hypermedialer Lernsysteme: Lernvoraussetzungen, kognitive Re-Interpretation und Interaktion. In R. Kammerl (Hrsg.), Computerunterstütztes Lernen, S. 40-52. München: Oldenbourg.

Schulmeister, R. (2002a) Grundlagen hypermedialer Lernsysteme: Theorie-Didaktik-Design. München: Oldenbourg.

Schulmeister, R. (2002b) Virtuelle Universitäten und die Virtualisierung der Hochschulausbildung: Argumente und Konsequenzen. In L.J. Issing, G. Stärk (Hrsg.), Studieren mit Multimedia und Internet: Ende der traditionellen Hochschule oder Innovationsschub?, Volume 16 of Medien in der Wissenschaft, S. 129-145. Münster: Waxmann.

smartframe (2003, Februar) http://www.smartframe.de. Letzter Abruf: 01.02.2003.

Sniedovich, M.; Byrne, A. (2003, Februar) tutOR,. http://www.tutor.ms.unimelb.edu.au/frame.html. Letzter Abruf: 01.02.2003.

Strittmatter, P.; Niegemann, H. (2000) Lehren und Lernen mit Medien. Darmstadt: Wissenschaftliche Buchgesellschaft.

Süß, C. (2003, Februar) Learning Material Markup Language. http://www.lmml.de. Letzter Abruf: 01.02.2003.

van Merrienboër, J.J.G.; Seel, N.M.; Kirschner, P.A. (2002) Mental models as a new foundation for instructional design. Educational Technology: The Magazine for Managers of Change in Education 42(2), S. 60-66.

Voß, S.; Martello, S.; Osman, I.H.; Roucairol, C. (Hrsg.) (1999) Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Boston: Kluwer.

W3 Consortium (2003a, Februar) Mathematical Markup Language (MathML) 2.0 Recommendation. http://www.w3.org/Math. Letzter Abruf: 01.02.2003.

W3 Consortium (2003b, Februar) XSL transformations. http://www.w3.org/TR/xslt. Letzter Abruf: 01.02.2003.

WebCT (2003, Februar) WebCT.com. http://www.webct.com. Letzter Abruf: 01.02.2003.

wwr Wissenswerkstatt rechensysteme (2003, Februar) wwr-Projekt. http://wwr-project.de. Letzter Abruf: 01.02.2003.

INSTRUCTIONAL DESIGN OF INTERACTIVE LEARNING MODULES BY EXAMPLE OF A SPECIAL FLOW SHOP PROBLEM

René Eisenberg, Torsten Reiners, Imke Sassen, Stefan Voss
University of Hamburg
Institute of Information Systems
Von-Melle-Park 5
D-20146 Hamburg,
Germany
{eisenberg, reiners, sassen}@econ.uni-hamburg.de, stefan.voss@uni-hamburg.de

ABSTRACT

Modern education of scheduling can be improved by using interactive learning methods in order to build and develop modeling and problem solving skills. In this paper we discuss some possibilities of supporting the teaching in the field of production planning and the demands for a digitally published virtual training course in order to depict scheduling problems vividly. To provide students with a valuable introduction to a representative cross section of scheduling problems we present an interactive learning module for a flow shop problem using the application SIMTOOL. This is a discrete event-driven simulation tool, especially designed for education within the classroom and the self-paced study to demonstrate main principles as well as internal processes.

KEYWORDS

e-learning, education, flowshop, interactive virtual learning, production planning, scheduling, interactivity

1. INTRODUCTION

Research and development in the field of e-learning is leading to a variety of virtual learning environments and support of education using information technology. These systems range from the simplest form of electronically providing students with slides of a specific course to complete virtual universities. The main goal is the improvement of education by increasing the availability of learning material as well as allowing the enrollment in entire virtual courses. This establishes the option for students to learn independently (in terms of location and time) from the presence university.

Currently several research projects exist with the objective to analyze the potential of using modern and innovative information technologies and to improve the education within (presence) universities. Especially virtual learning environments are seen as media to enforce variation of traditional academic education and, therewith, the overall learning experience of students.

While teaching a course on production in our institute we were enabled to gain experience on teaching scheduling problems by means of different presentation techniques. Within our lectures both static slides as well as animations are used to present learning material on the same subject. Furthermore, several subjects may be observed and presented from different perspectives allowing a deeper understanding of the material. Based on reactions from students subsequent to the course, animated slides seem to stimulate more interest and even promote a better understanding of problems in the field of scheduling.

In this paper we want to show how teaching at university level in the field of machine scheduling may be extended and enhanced with a didactical focus on web-based interactive technology. For this purpose, we developed a java-based prototype of a discrete event-driven simulation tool including certain mechanisms for virtual learning environments. This application is used as a platform to develop interactive learning units by simulating the behavior of, e.g., a production system. The primary goal is the integration of simulation models for machine scheduling within self-paced studies allowing the students to interactively work with the learning material. Instead of showing static diagrams and presenting textual descriptions of applications and procedures, we want to give the opportunity to explore the treated problem and develop solution procedures in a self-guided fashion with detailed feedback.

Scheduling describes the problem of finding a feasible schedule for a given set of m machines M_i (i=1, …, m), which have to process a given set of n jobs J_j (j=1, …, n). A schedule consists of the assignments of time intervals for each job to one or more machines. Schedules may be represented by Gantt charts showing the usage of machines (job-oriented Gantt chart) or processing of jobs (machine-oriented Gantt chart) on a time-line, cf. Figure 1. The overall processing time (makespan) is determined by the finishing time of the last job or machine, respectively. Furthermore, a job J_i may be split into m_j operations $O_{j1}, \ldots, O_{j,m_j}$.

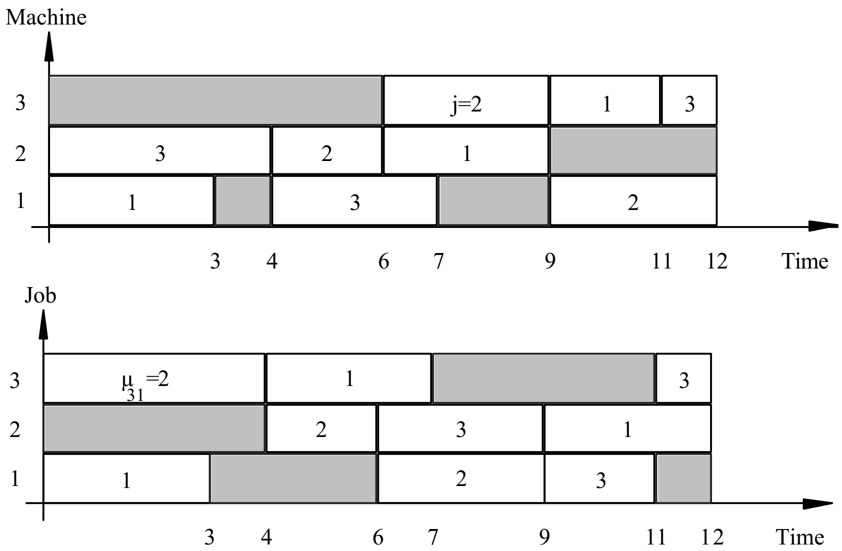


Figure 1. Gantt charts (j indicates a job and μ indicates a machine)

Schedules may be subject to certain restrictions. These restrictions describe the characteristics of the jobs, the machines, and their relations. Examples for job characteristics are preemption and precedence relations, examples for machine characteristics are, e.g., the number of machines or the kind of environment such as the flow shop. Within flow shops all jobs have to be processed on the machines in the same order. That is, each job J_j consists of m operations with operation O_{ji} being processed on machine i.

Finally, an objective function has to be defined for the scheduling problem, which is commonly the makespan. While there is a long debate about the usability of corresponding problems in the literature (see, e.g., Reisman et al. 1997) they may serve as excellent examples for teaching purposes. For a comprehensive discussion on production planning, machine scheduling, and especially flow shop scheduling the reader may be referred to Brucker (2001) or Domschke et al. (1997). The classification and characterization of scheduling problems goes back to the work of Graham et al. (1979).

The paper demonstrates how a learning unit on machine scheduling can be designed. We focus on a special case of the flow shop problem where the jobs are transferred from one machine to another—as long as not all operations are performed—including the consideration of transportation times; cf. Stern and Vitner (1990), Panwalkar (1991), and Lee et al. (1997). The example in Figure 2 shows a rather realistic model of a flow shop problem with two machines (M_1, M_2), one warehouse, and three jobs (J_1, …, J_3), which are transported by an automated guided vehicle (AGV). The objective is to determine a feasible schedule, which minimizes the makespan.

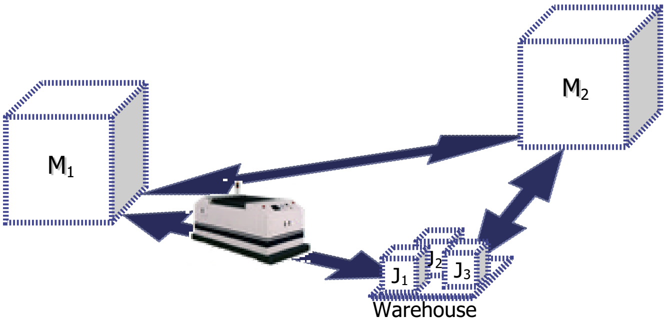


Figure 2. Two-machine flow shop problem with transportation times

This example demonstrates the production system, but is not very useful to show exactly how processing, transportation, and waiting times add up to the makespan of the whole problem. Therefore, we present an overview of different applets that can be used by students to understand the problem formulation and to interactively explore possible hazards in finding good (or even optimal) solutions, e.g., the order of the jobs, blocking machines, or transport units that are mainly moved empty.

In the next section we discuss some didactical aspects. In Section 3 we introduce the simulation tool SIMTOOL integrating the time dimension into the above model, which allows students to influence the production process and, therefore, the objective function value (makespan). Section 4 concludes the paper with a critical discussion on our approach.

2. DIDACTICAL ASPECTS

Advances in information and communication technologies lead to an increasing popularity of technology-supported learning. That is, most universities are involved in projects to transfer their learning material at least to electronically available documents. This fact poses the question how education can profit significantly from hypermedia and e-technologies. Within this context virtual and interactive learning modules have become crucial. Experiments are an important aspect of traditional scientific methods whereas a virtualization of such experimental exploration possibilities is difficult to realize in a way that allows the student to gain new insights of internal processes as well as to validate theories and methods. A primary goal of interactive learning material is to support the student to achieve better understanding for given problems by developing and testing own hypotheses.

Here, we demonstrate how to proceed from a classroom presentation to a learning unit for a virtual learning environment within a self-paced study. In general, the classroom presentation may contain slides, which are static—not counting special effects—like text zooming in or exploding screens. Therefore, a slide could contain the visualization of a flow shop problem as shown in Figure 3. Here, the transportation unit is in the focus transporting the job from the warehouse to the corresponding machine and vice versa. The lengths of the boxes for each machine represent the processing times of the jobs.

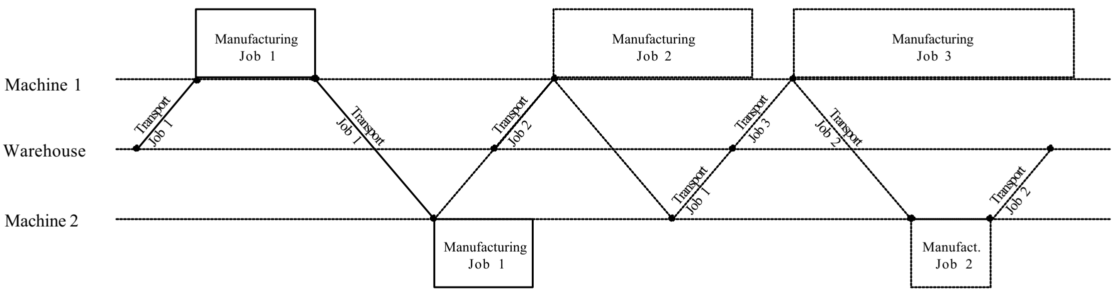


Figure 3. Flow shop problem, cf. Domschke et al, 1997, p. 394

The presentation of the example can slightly be improved by adding animations. Figure 4 shows an example where the transportation unit is moving between the warehouse and the machines while the time is visualized by an ongoing filling process of the jobs. Anyhow, the animation still misses interactive features such that a student is restricted to passively observe the presentation. Even simple features like moving forward and backward as well as changing the composition of jobs are not given.

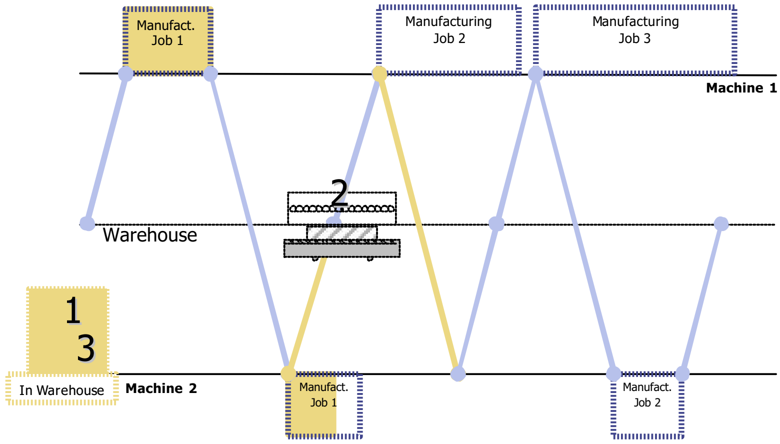


Figure 4. PowerPoint presentation with integrated animation paths

Even though the process-oriented presentation is probably an adequate way for education, another form is common and closely related to the previous design. So-called Gantt charts present a coordinate system with time-machine axes in which the jobs are drawn according to their properties. Here, the movements of the transportation unit are not given but instead the interdependences of the jobs are easier to recognize, in particular if each job is shown using a specific color.

Figure 5 demonstrates two examples of Gantt charts for interactive learning; cf. Hochbaum (2003) for the example on the right-hand side, and Hartmann et al. (2003) for the one on the left-hand side. Both applets demonstrate the usage of virtual and interactive learning but also miss certain important components. The student has to find the order of the jobs for a short time span by moving the jobs to a new location on the time line (using drag-and-drop). Errors and bad solutions are indicated but the user interface, first, lacks certain information, e.g., explanations, help, and traceability, and, second, at least of the applet by Hartmann et al. (2003), is not intuitive such that the student needs a large period of vocational adjustment.

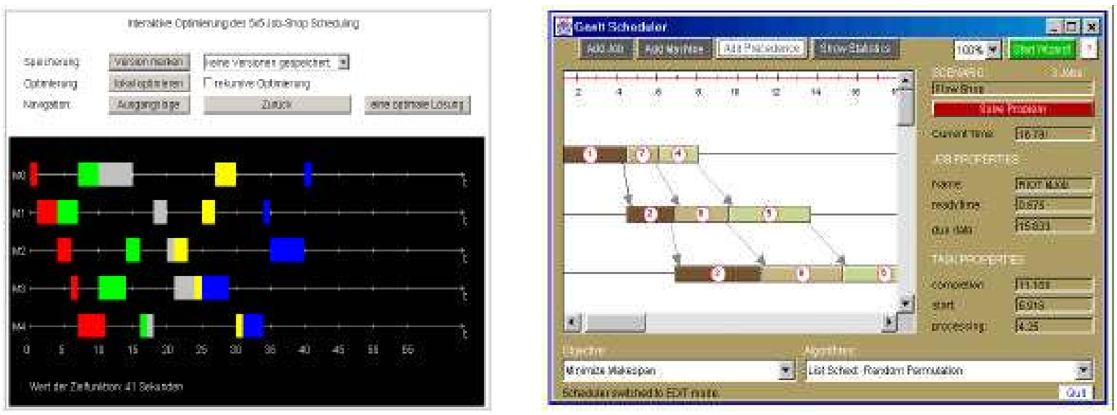


Figure 5. Examples for interactive flow shop applets within virtual learning material

The rapid growth in demand and usage of sophisticated technology does not necessarily result in a higher learning efficacy. For virtual and interactive learning material to be successful both its instructional as well as the didactical design has to reflect the characteristics and possibilities of multimedia. Furthermore, the research publication in the field of cognition science should be considered for the design of the learning material, always with regard to the student. Particularly missing communication channels have to be considered. Teamwork or discussions like in traditional presence courses are not given to the same extent in virtual learning. In most virtual learning environments the student goes through the course by himself such that all required information has to be considered in advance. That is, all interactive components should be self-explanatory, a glossary should explain the most important terms, and prerequisites should be introduced, at least by short abstracts.

The use of complex technology without the consideration of the didactical structure of certain content may require a higher level of total participant effort. According to the cognitive effort perspective (cf., e.g., Todd and Bensabat, 1991) individuals will adapt their strategies in such a way that they limit their overall expenditures of effort, as the subcomponents of cognitive tasks are made more or less laborious (cf. Alavi et al. 2002). The cognitive effort perspective may provide some insights how well designed interactive learning modules may influence learning in a positive way. The easiest possibility to reach a qualitative improvement of virtual and interactive learning modules is the consideration of experiences from teaching traditional courses. On account of this we would like to present a more illustrative interactive learning module with further self-explanatory and motivating features in the following section.

3. INTERACTIVE VIRTUAL LEARNING USING A SIMULATION TOOL

Simulation tools are used to transfer the reality into a (simple) model, which can be used for experiments like introducing new scheduling algorithms, different machines to produce a good, or evaluating the influence of defects. There are several (commercial) simulation tools but most of these tools are not designed to be used within e-learning. That is, special visualization of internal processes, explanations, or adaptation to the student is not integrated. Therefore, we developed the application SIMTOOL, a discrete event-driven simulation tool, specially designed for education within the classroom and the self-paced study to demonstrate main principles as well as internal processes. Especially features like lesson specific configuration, feedback to performed actions, guidance by the simulation tool while solving a problem, and adaptation of the difficulty level regarding the student's background knowledge.

The design allows a high degree of configurability defined in so-called lessons. Depending on the aim of the lesson, simulation objects can be turned on or off and the functionality as well as the interactivity might be restricted with respect to required options to solve the task. Furthermore, XML-based scripts can be used to define macros as well as events that control the learning process; cf. Klie and Schalong, 2003. A sample lesson might be constructed as shown in the following example. The initial simulation model is built by executing a script in the beginning, whereas, subsequent to the building phase, the student has to solve the given problem formulation by interacting with the simulation tool. Depending on the difficulty level and defined events, the student is observed such that appropriate feedback is displayed. For example, if the distribution of a source has to be set to a normal distribution with a given mean and deviation value the event-action mechanism can be triggered in a way that all inputs that diverge by more than 100% from the expected values are rejected including a negative feedback in form of a textual explanation. This allows as much free exploration as possible without losing the control to the student.

SIMTOOL is still a prototype and, therefore, does not have a large library of components that can be used. The author of lessons might have to implement new components or extend the script language, which is intensively supported by the modular design of SIMTOOL. In the following example, some components need to be extended by further attributes and the control center has to be created. Even though this is a larger component and requires a skilled Java-programmer, several interfaces exist to access required information for the display and to implement the interactivity.

In the following, we sketch a learning unit allowing the student to explore the processes regarding a flow shop problem. Here, we mainly focus on the students' view about the design of components as well as the complete model. Incoming orders or jobs, respectively, (generator) are stored in a warehouse and then transported to the machine according to their properties. Jobs are represented by immobile units (here a box), which contain a list (attribute) of the processes to be accomplished (which machines and, if essential, in which order) before being delivered to a customer (order/job termination) as well as the status of each process. In the lower part of the screen the jobs are shown including their status (a list of processes including the expected processing time in time units whereas a process can be not started yet (-), currently processed (p), waiting to be picked up by the transportation unit (w), or being finished) and their current location. The links between the warehouse and machines are non-movable. That is, the jobs can only be moved by a (automated) transportation unit (here emblematized by a palette, it can also be an AGV). Depending on the difficulty the amount of feedback as well as interactivity is set; the following description assumes a beginner.

After starting the simulation, the transportation unit has to transport jobs between the warehouse and the corresponding machines (as defined in their processing list). Here, the student has to decide on the sequence of the jobs as well as the assignment to a machine. This is accomplished by pausing the simulation whenever an event occurs; e.g., there are no further events in the event list, the transportation unit reaches its destination, or the status of a machine is changing. Whenever the model is paused, the student has the following options: (1) Load a job to the transportation unit, (2) send the transportation unit to a new location, (3) ask for hints, which can also be executed, (4) wait for the next event, or (5) just continue with the simulation. Assuming the situation as shown in Figure 6 where the transportation unit is empty on its way to the warehouse, machine 2 finished the processing of job 4 and waits for pick-up, and machine 1 is processing job 1. Note that the order of the jobs being processed is defined by the student and might not correspond with the best possible solution. Here, the transportation unit performed the following transports: job 4 from the warehouse to machine 1, back to the warehouse (empty), job 1 to machine 1, where job 4 is waiting to be brought to machine 2, back to the warehouse (empty). Furthermore, the transportation unit waited at the machines for several periods. In the moment where the transportation unit arrives at the warehouse, the student has to load the next job according to his considerations. In case of loading job 2, the transportation unit is send to machine 2. Afterwards, the event “job 1 at machine 1 is ready for pick-up” is triggered (the student just continues the simulation) the transportation unit arrives at machine 2. Here, job 2 is automatically placed on the machine, while job 4 is loaded onto the transportation unit. Then, the student has to define the next action, here the transport back to the warehouse to deliver the completely processed job.

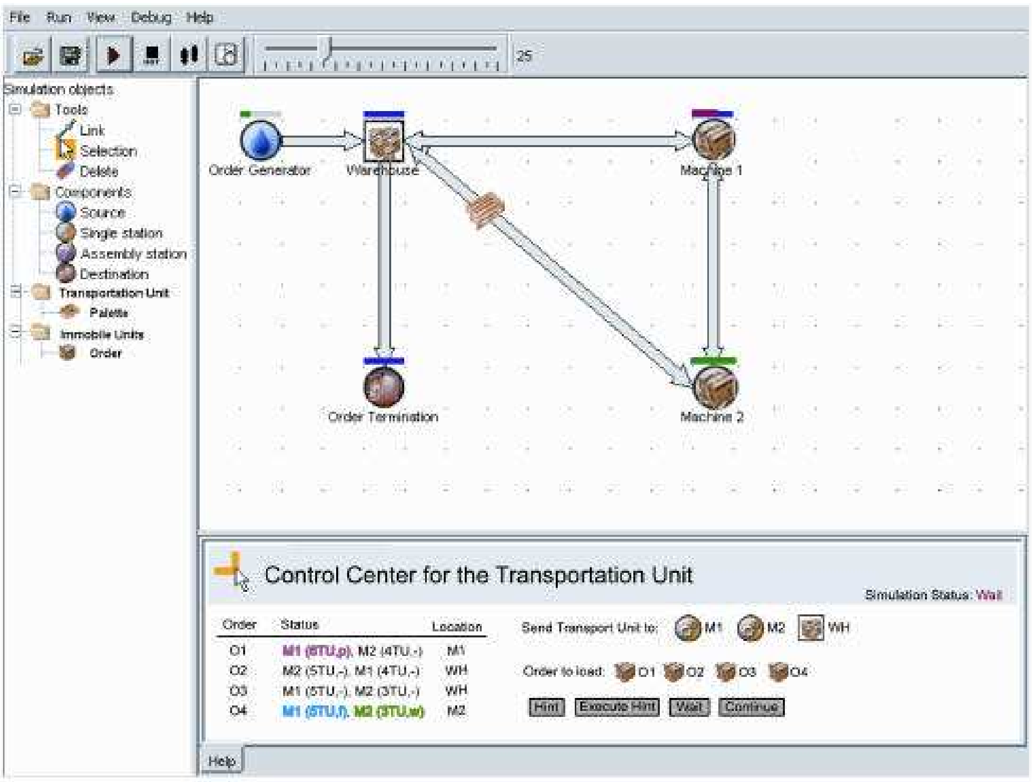


Figure 6. Screenshot of the SIMTOOL applet

The simulation is finished as soon as all jobs reach the destination (order termination). All actions are written to a protocol, which can be analyzed by the student, compared to an existing Gantt chart, or stored for a later comparison with another run of the simulation model.

During the learning phase with SIMTOOL the student is able to develop production planning skills, especially machine scheduling, based on own experiences. Additionally, the student may be motivated to go through further simulation trials in order to improve solution procedures (the final report shows the differences between various runs and the optimal solution). Over and above, within the implementation of a game-like approach solutions of different students may be compared in a high-score list, which should act as an incentive. The quality of the decisions made by the student is quantified by a score so the students can compete with each other. Hence, the motivation to deal with a complex subject may increase because the students participate more actively and absorb more of the presented material.

4. CONCLUSION

SIMTOOL as a sophisticated planning tool for teaching simulation allows a flexible, individualized, and experiential learning in higher education. The applet implements a learning unit that lets students interactively experience a two-machine flow shop problem with transportation times. Different from the traditional lecture and textbooks that are more or less forms of a one-way communication, the student can influence the production process and realizes the direct consequences of his action. Once the student manages to pass the task completely this leads to a deeper understanding and internalisation of the subject. Furthermore, learning is independent from time and location, i.e., one may choose where, when, and how fast to study.

Of course, interaction is limited due to the existence of trade-offs between interactivity/feedback and implementation costs. The teaching effort will not decrease through this new technology at short notice, since a website cannot replace the human lecturer. But new information technology surely can enhance teaching in the field of production planning.

Hence, there are compelling reasons to incorporate SIMTOOL into virtual learning environments as well as traditional classrooms. SIMTOOL simulates realistic problems, e.g., a flow shop problem, allowing the students to gain a better understanding of the problem itself and to design and test various solution strategies in an engaging environment. This allows the instructor to illustrate learning material using realistic instances that reflect the true complexity of diverse planning problems. SIMTOOL allows a flexible design of course units by writing scripts to guide the students. That is, certain milestones are defined using the XML-based script language, help texts have to be written that are presented with respect to the difficulty level of the students, and parts of free exploration with given limits are specified. The graphical representation of the simulation model can be adapted by including images.

Since scheduling problems usually consist of movable as well as non-moveable objects, which have to be related to one another over time in order to generate feasible solutions, the interactive simulation of the production process using SIMTOOL seems to be highly suitable. In addition to scheduling problems within production planning other fields such as project planning and scheduling, the scheduling of threads of computer programs, or flight scheduling can be realized by reusing learning units within other context.

ACKNOWLEDGEMENT

This work was supported by funding from the BMBF (Bundesministerium für Bildung und Forschung) in the program “Neue Medien in der Bildung” (08NM094D).

REFERENCES

Alavi, M., G.M. Marakas, and Y. Yoo, 2002. A Comparative Study of Distributed Learning Environments on Learning Outcomes. *Information Systems Research*, 13:404–415.

Brucker, P., 2001. *Scheduling Algorithms*, 3rd Ed., Springer, Berlin.

Domschke, W., A. Scholl, and S. Voß, 1997. *Produktionsplanung: Ablauforganisatorische Aspekte*, 2nd Ed, Springer, Berlin.

Graham R.L., E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan, 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326.

Hartmann, W., R. Reichert, and D. Hornung, 2003, February. Informatik auf EducETH. http://www.educeth.ch/informatik/interaktiv/scheduling/einleitung.html#ANIM. Date of last check.

Hochbaum, D.S., 2003, February. Scheduler Demo. http://riot.ieor.berkeley.edu/riot/Applications/Scheduling/demo/flow1.html. Date of last check.

Klie, T. and C. Schalong, 2003. A Script-driven Simulation Tool as an Improvement to the Education in Web-based Virtual Learning Environments. Diploma Thesis, University of Technology Braunschweig.

Lee, C.-Y., L. Lei, and M. Pinedo, 1997. Current trends in deterministic scheduling. *Annals of Operations Research*, 70:1–41.

Panwalkar, S.S., 1991. Scheduling of a two-machine flow-shop with travel time between machines. *Journal of the Operational Research Society*, 42:609–613.

Reiners, T. and S. Voß, 2003. Some Thoughts on how to Educate OR/MS. Working paper, University of Hamburg, Submitted to *European Journal of Operational Research*.

Reisman, A., A. Kumar and J. Motwani, 1997. *Flowshop scheduling/sequencing research: A statistical review of the literature, 1952–1994. IEEE Transactions on Engineering Management*, 44:316–329.

Schulmeister, R., 2003. *Lernplattformen für das virtuelle Lernen: Evaluation und Didaktik.* Oldenbourg, Munich.

Stern, H.I. and G. Vitner, 1990. Scheduling parts in a combined production-transportation work cell. *Journal of the Operational Research Society*, 41:625–632.

Todd, P. and I. Bensabat, 1991. An experimental investigation of the impact of computer based decision aids on decision making strategies. *Information Systems Research,* 2:87–115.

SMARTFRAME: An Integrated Environment for XML-Coded Learning Material

Torsten Reiners, Stefan Voß
University of Hamburg

Dirk Reiß, Henrike Schulze
University of Technology Braunschweig

Abstract: Universities and lecturers can improve the quality of education by offering virtual course material in addition to the classroom presentation. In most cases, this is either done through downloadable pdf-files or static html-pages even though there is a shift to the application of virtual learning environments. Here, it is important that the added-value of more technology increases the learning experience rather than surrounding the (static) html-pages with some gimmicks. In this paper we present SMARTFRAME, a prototype for a virtual learning environment incorporating a hierarchically structured design of XML-coded learning material including meta-data information. A demarcation to other products is based on the direct application of meta-data during the transformation process of the learning material to the desired output format as well as the integration of innovative concepts and components.

Keywords: e-learning, virtual learning environment, LOM, LMML

1 Introduction

Analyzing actual developments within the sector of virtual learning environments (VLEs), one realizes that a wide-spread variety of different VLEs exist, either freely available or proprietary ones. Some of them offer an integrated learning environment including features for collaborative work like chat- and forum-functionality or an integrated whiteboard (see, e.g., WebCT [WebCT03] and Blackboard [Black03]); others are a loose collection of interactive Java-applets or Macromedia Flash-animations (see, e.g., [SnBy03] and [Bal⁺03]). These VLEs have in common that they predominantly aggregate the learning material in a more or less static form, i.e., it is only available in standard html-format and lacks the ability to be (re-)used in other contexts. More differentiated approaches use XML for encoding and meta-data for describing the learning objects (see, e.g., [Met03], [Orw03], and [ReVo03]).

Being interested in developing learning material for different courses at university level that are enriched by modern and innovative didactical concepts to improve the learning experience, we develop a concept for a web-based VLE. Main aspects of our architectural concept SMARTFRAME (Smart Technology for Research and Modern Education; see [Smar03]) are its flexibility and configurability for the learner, the structured design of the XML-based learning material, and the exclusive usage of modern (standardized) technologies that are available without charges, i.e. due to open-source projects. In contrast to most VLEs, the content is stored in hierarchically structured semantic units, so-called learning objects, which are directly composed and transformed according to user-specific configurations and meta-data descriptions to the requested output format. Depending on the learning style and goal, the learner can select the form and density of the presentation. That is, if the learner repeats course units to memorize the most important facts, the learning material can be (automatically) condensed to the most relevant parts regarding the learners' needs by hiding "non-relevant" and optional learning objects.

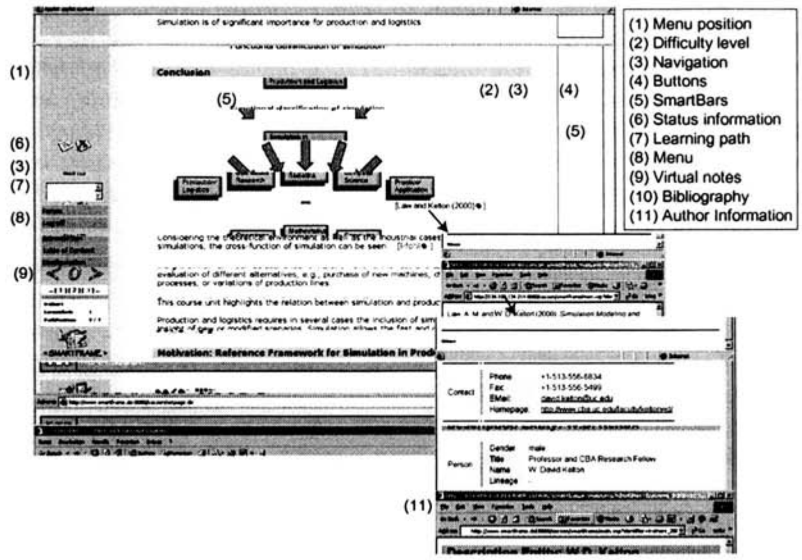


Figure 1: Screenshot of SMARTFRAME

Figure 1 shows a screenshot of the current version, whereas the development is part of an ongoing research project. The VLE consists of a control bar at the upper part containing buttons for superior functionality like navigation, activation of SmartBars, and communication, as well as a menu bar on the left-hand side. The menu bar includes status information, navigation, learning path, menu, and virtual notes. Note that the position of components is not fixed but can be on either side of the browser or completely turned off. In the following this and the underlying technology is called the graphical user interface. Figure 1 shows also the Smart-

Bars on the right-hand side, which is an innovative technology to visualize aggregated information from the learning material; see [Rei+02] for a detailed description of SmartBars and other concepts for improving the presentation of learning material. The learning material itself is displayed in the main area of the browser. The two small additional windows show a bibliographic reference and author information, which can be opened by the learner clicking on the bibliographic reference in the learning material.

The design of a course can be obtained either by a technically driven or by a didactically driven approach; see [ReVo03]. The first approach is (usually) limited regarding the incorporation of new didactical methods. Therefore, we decided to follow the second one. Subsequent to a preliminary evaluation and, therewith related definitions of the most relevant didactical methods—for the learners as well as the authors—that have to be at least part of the VLE, we designed a specific concept being realized within SMARTFRAME; see [Rei+03a] as well as [Fra+02] for the didactical background of this paper. In this paper we focus on the technical view, which allows a high degree of modularity, flexibility, and reusability due to its modular and open design.

The following technical description of the system architecture as well as the learning material will be explained in such a way that we show how the example in Figure 1 is produced. That is, we describe the structure of the learning material (Section 2.1) and give a brief overview of the content encoding (Section 2.2). In Section 2.3 we describe the meta-data concept being a further specification of the content and learning objects, respectively. In our concept we focus on the Learning Material Markup Language (LMML; see [Sues03]) and Learning Object Metadata (LOM; see [IEEE03]) and, therefore, show how these specifications can be combined and modified with respect to insufficiencies and redundancies. The technical realization of how the data is stored and the author can be supported using an authoring tool is discussed in Section 2.4. The section about the learning material is concluded by a description of the content aggregation and transformation into the user-specific presentation (Section 2.5).

In Section 3 the technical architecture of the graphical user interface of SMARTFRAME is described including the configuration and navigation. Besides presenting a concept, the technical realization is explained, i.e. how Servlets, JavaBeans, and JSPs are used to compose a VLE. The paper is concluded in Section 4 by presenting further components being integrated in SMARTFRAME as well as an outlook on future developments.

2 Learning Material

There are several choices of how to encode learning material. In this paper, we focus on different factors that allow a later reusability in various contexts, incorporation of modern technologies as well as practicability. That is, XML-based learning material is chosen due to its popularity, future perspective, and readability but also from a technological point of view, i.e. simple transformation to various output formats by using XSLT (eXtensible Stylesheet Language Transformations; see [W3C03b]), processing by freely available tools, and standard operations like full textual search. Furthermore, several organizations work on standards specifying the encoding of learning material aiming at interchangeability between different VLEs. Our concept is closely related to LMML, whereas several components had to be adopted, especially in terms of refining the underlying didactical model as well as combining LMML with the meta-data specification LOM. This section describes the learning objects and their encoding in LMML, sketches the LOM specification, our modifications as well as the concept for an authoring tool, and outlines the transformation.

2.1 Learning Objects

The reusability of learning material is improved by defining a hierarchical and modular structure of different learning objects. Each learning object is encoded as an XSP-file (eXtensible Server Pages) using the following structure:

```
(a1)  <xsp:page language="java"
(a2)    xmlns:xsp="http://apache.org/xsp"
(a3)    xmlns:smartframe="http://www.smartframe.de/xsl"
(a4)    xmlns:cinclude="http://apache.org/cocoon/include/1.0">
(a5)
(a6)    <smartframe:header/>
(a7)
(a8)    <smartframe:me identifier="id_1">
(a9)      <!-- CONTENT -->
(a10)   </smartframe:me>
(a11) </xsp:page>
```

Lines (a1)-(a4) define the type of the file as well as the used namespaces, line (a6) will force the inclusion of code, e.g., to resolve links to other learning objects, and line (a8) identifies the learning object by its granularity level (*smartframe:me* corresponds to a media element) and a unique identifier (*id_1*).

Learning objects are composed to larger units using four different types of links. The tag *<smartframe:link type="type" ref="id">* is used with *type* being chosen from the following list:

- **referenceinternal/-external**: Another learning object, either internal (part of the local learning material) or external (all other material like web-sites in the Internet), is only referenced. That is, the object is not completely integrated but displayed as a link using descriptive information from the object, e.g., the caption of a figure is shown as a reference instead of the figure itself.

- **includeinternal/-external**: The learning object is completely integrated, e.g., the figure is included and shown.

Depending on the content, size, and relation to other objects, we distinguish different levels of granularity of learning objects with the following meaning. Note that variants of the learning material in terms of the language are stored within the same learning object file.

- **media element (ME)** is a small unit without further partitioning, whereas it is important that other learning objects are not included but referenced as described above. Examples are text fragments, tables, figures, glossary entries, formula systems, or media objects (e.g., sounds, videos, or photos). The following example defines an object of the type image with the title (or caption, respectively) specified in line (b3) and a link to the image itself (b4). Different versions of the learning object regarding the language are distinguished using the attribute *xml:lang* identifying English (b2-b6) and German (b7-b9).

```
(b1)  <smartframe:me identifier="id_2">
(b2)   <lmml xml:lang="en">
(b3)    <image title="Areas of Simulation">
(b4)     <smartframe:link type="referenceinternal" ref="id_3"/>
(b5)    </image>
(b6)   </lmml>
(b7)   <lmml xml:lang="de">
(b8)    <!-- content in german -->
(b9)   </lmml>
(b10) </smartframe:me>
```

- **learning element (LE)** is a composition of objects (media elements as well as learning elements) to produce a semantic unit. That is, the content of a "webpage" or section in a printed document. So-called pagebreaks can be used to split up the LE in smaller parts being displayed separately using standard navigation.

```
(c1)  <smartframe:le identifier="id_le">
(c2)   <lmml xml:lang="en" >
(c3)    <section title="Course in Simulation">
(c4)     <motivation title="A Motivation for the Course">
(c5)      <smartframe:link type="includeinternal" ref="id_1"/>
(c6)     </motivation>
(c7)     <smartframe:pagebreak/>
(c8)     <!-- text component not shown -->
(c9)     <image title="Functional Classification of Simulation">
(c10)     <smartframe:link type="includeinternal" ref="id_2"/>
(c11)    </image>
(c12)    <!-- etc -->
(c13)   </section>
(c14)  </lmml>
(c15) </smartframe:me>
```

This example shows the inclusion of two other objects as well as its adaptation. The attribute *title* of image (c9) is used instead of the originally defined title of the included object (see previous listing) and, therefore, supports the reusability in terms of adaptation to the context.

- **content module (CM)** is superior to learning elements grouping these to larger units. The content module should cover a certain content of the lecture or a special field, or include the slides of a course or seminar. The presentation to the learner should be realized as a table of contents or hyperbolic tree or graph, respectively.
- **thematic metastructure (TM)** covers a whole thematic field in the VLE and could either be the work of a department in a specific field or the courses a learner has to take in order to obtain a certain degree.

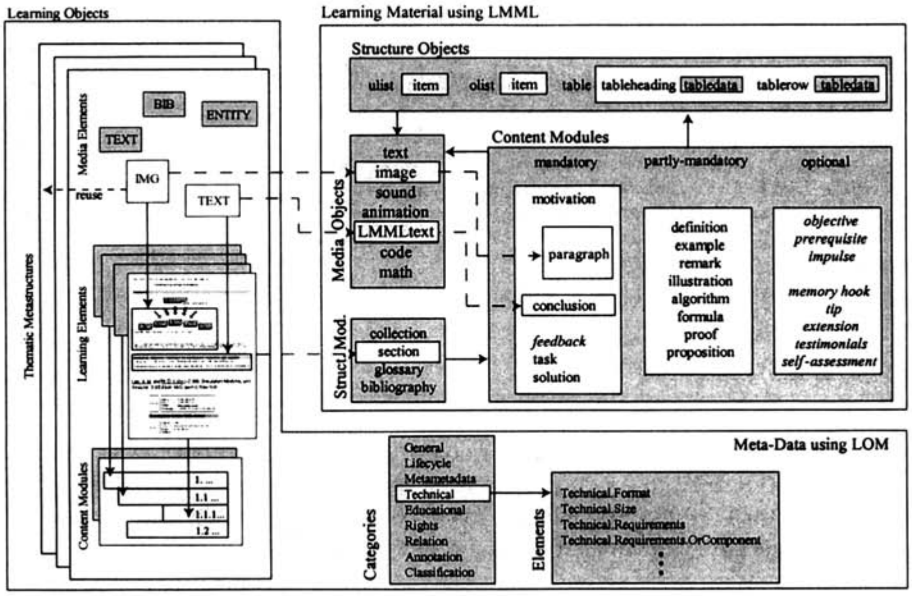


Figure 2: Structure of learning material

Figure 2 gives an overview of the learning material. On the left-hand side, the different kinds of learning objects are shown, which are used to construct the sample course unit. Note that the media element IMG is reused within another thematic metastructure. The MEs are combined within a LE, whereas the bibliographic entry is part of the LE but only shown to the user by the SMARTBARS. The learning objects are mainly built using LMML as shown for the media elements IMG and TEXT. In addition to LMML, objects like BIB and ENTITY use further specifications, e.g., DocBook [WaMu99], QTI (Question and Test Interoperability, [IMS03b]), and SVG (Scalable Vector Graphics [W3C03a]). For a detailed description on how to encode learning objects see [Rei[+]03b].

2.2 Standards for the Encoding

The majority of the learning objects is coded using LMML; only non-supported types like bibliographic entries have to be based on other approaches. LMML introduces further semantical classification of the learning material by distinguishing four categories. Figure 2 shows on the upper right-hand side structure modules (used to enclose and structure the content), content modules (to compose the learning material), structure objects (to build tables and lists), and media objects to include and classify the content itself.

Our sample course unit first has to motivate the learner (*motivation*) before presenting the content in form of a textual and pictorial block (*paragraph*) that uses the media objects *LMMLtext* and *image* to include the respective content. The course unit is concluded by a content module *conclusion*. The course can be enriched by using further elements, whereas we also define new elements (presented in italics) not considered in the LMML-specification. The modularization allows a user-specific presentation according to his own configuration as well as difficulty level by selecting required objects. A learner of a beginner level might require the learning objects *motivation*, *objective*, and *prerequisite* to follow the subject, whereas the expert might only want the main definitions or theses.

In the following we describe the transformation process on the learning object *motivation*, whereas the textual content is shortened and the header of the XSP-file has been left out; see also the examples in the previous section.

```
(d1)    <motivation title="A Motivation for the Course">
(d2)     <paragraph>
(d3)      <LMMLtext>
(d4)       Production and logistics requires ...
(d5)      </LMMLtext>
(d6)     </paragraph>
(d7)    </motivation>
```

In addition to text-based content in Unicode, MathML, or SVG, binary learning objects have to be used for, e.g., sounds in mp3 or pictures in gif. These objects are wrapped, specified, and referenced by another object coded in LMML using the tag *<smartframe:link>*. During the transformation process the reference is resolved into the real location of the object (LOM-database) and inserted into the output according to the mime type.

```
(e1)    <image title="Functional classification of simulation">
(e2)     <smartframe:link type="includeinternal" ref="id_3"/>
(e3)    </image>
```

2.3 Meta-Data

Learning objects have to provide mechanisms for authors as well as learners, allowing reusability, classification with respect to several criteria, and recovery in a large pool of objects. Therefore, each learning object is specified by meta-data as shown in Figure 2 on the lower right-hand side. Currently, several organizations develop specifications for meta-data to be used within VLEs; see, e.g., ARIADNE [Aria03], IMS [IMS03a], LOM [IEEE03], or SCORM [ADL03]. In our approach we decided to use LOM, which basically arose from the other standards and is on the verge of being published as an IEEE-standard.

Even though LMML itself also incorporates meta-data for the learning objects, it is to be preferred to use external specifications. That is, the meta-data in LMML is designed as a part of the content and, therefore, can only be used if a full textual search over all objects is initiated. Hence, we modified LMML in a way that meta-data elements are either eliminated in case of redundancy or moved to LOM by being added as new elements in the appropriate category; see the following list for some examples and [Rei⁺03b] for a detailed description.

Group	LMML-meta-data	LOM element
general	author	lifecycle.contribute
time	creationTime	lifecycle.contribute.date
content	language	general.language
pedagogical	difficulty	educational.difficulty
structure	type	educational.learning resource type

Table 1: Examples

Additionally, further attributes in LMML, e.g., for linking learning objects or referencing bibliographic entries, were changed due to our design of the XML-based learning material. The linkage of other learning objects using direct URLs (*target_module)* does not allow the incorporation of the meta-data during the transformation process and bibliographic entries are coded as separate learning objects and, therefore, have to be referenced in the same way as other objects (*<smartframe:link ref="identifier">* instead of *bibkey*).

Even though the current version of the LOM-specification reached a final status, there are still required adaptations with respect to allowing an appropriate application. Especially a major error in the category *technical* has to be corrected. In the specification, several requirements are conjunct with *and*, whereas in the requirements itself further *or*-conjunctions are allowed. Unfortunately, this does not allow the definition of terms following the Boolean algebra and, therefore, had to be extended by further elements that allow the adequate definition of technical requirements for learning object. The category *technical,* as we amended it, has the following elements:

Requirements

Req.{Or,And,Not}Composite

Req.{Or,And,Not}Comp.Type

Req.{Or,And,Not}Comp.Name

Req.{Or,And,Not}Comp.Min Version

Req.{Or,And,Not}Comp.Max Version

This allows the definition of requirements T with the following properties:

$$t_i=(r_{i1} \wedge \ldots \wedge r_{ik}) \wedge (r_{i(k+1)} \vee \ldots \vee r_{il}) \wedge (\neg r_{i(l+1)} \wedge \ldots \wedge \neg r_{im}) \quad i=1,\ldots,n$$

$$T = t_1 \vee \ldots \vee t_n$$

Furthermore, the meta-data should correspond with the type of the learning objects. That is, not all meta-data seems to be useful to be set—if possible at all—to a default value. For example, the duration is necessary for movies but can and should be left empty with pictures, the interactivity of bitmap-graphics is generally low, and the semantic density of bibliographic entries is high. This can be considered by changing the cardinality of elements in respect of the learning object type; see [Rei+03b] for an extensive description.

2.4 Authoring Tool and Storage in Databases

Besides the great advantages of modularity, reusability, and the superior specification of learning objects with meta-data, the necessary labor for an author who has to determine and input all the required information exceeds the tasks for composing simple html-pages by far. Therefore, we implemented an advanced authoring tool that can be used to enter both the learning objects as well as meta-data in an intelligent way. That is, the authoring tool—configured by XML-schema-specifications of the LOM-categories and the learning objects as well as further XML-specifications about relations of elements—automatically sets all elements to a default value according to the type of each learning object that can also be based on the input of other elements. This allows a flexible and effective development of learning material that includes the necessary information for an improved learning experience; see Figure 3 for a screenshot showing the current authoring of meta-elements in the category *technical*.

Figure 3: Screenshoot of the authoring tool

Most of the data is encoded in XML suggesting the usage of XML-databases. Nevertheless, there are currently major disadvantages: XML-database-systems are either very expensive (and, therefore, contradict our concept of using freely available software) or not comparable to relational database systems regarding speed and stability. Therefore, we store the meta-data in a relational database using a data model based on a transformation from the XML-structures; see, e.g., [Schö03]. Due to the time consuming SQL-queries that have to be done for each learning object involved in the transformation process we adapted the resulting model with respect to efficiency. The learning objects themselves are not stored in the database but as files using the native file system. The location of the files is part of the meta-data such that every access to learning objects includes the meta-data information. Basically, the learning material can be distributed on the network, but in terms of speed a local storage from the learners' point of view is preferred and also supported by replication mechanisms.

2.5 Transformation

The XML-based learning material is transformed to, e.g., traditional printed scripts or web-pages. The transformation process of XML- and, especially, XSP-files is done with Cocoon [Apa03a], a specialized software running as a web-application under the Tomcat Servlet Engine [Apa03b], using XSLT (eXtensible Stylesheet Language for Transformation). In this section we demonstrate the transformation of the described learning elements into a web-based presentation; see Figure 4, especially the numbers in brackets. Based on the users' request, e.g., *http://www.smartframe.de:8080/cocoon/main.xsp?identifier=id_1,* the XSP-page *main.xsp* (see the following listing) is called with the identifier for the requested learning object as a parameter (1). The file *main.xsp* is used to initiate the transformation to a html-page by executing the link-tags *<smartframe:link .../>* with the identifier as a parameter, which then will be further processed by logicsheets.

```
<!-- XSP-header -->
<smartframe:header/>

<html>
  <body onload="initialize()">
     <smartframe:link>
        <smartframe:identifier>
          <xsp-request:get-parameter name="identifier"/>
        </smartframe:identifier>
     </smartframe:link>
  </body>
</html>
```

Logicsheets are used to transform specific tags (e.g., *<smartframe:header/>*) into markup language code-embedding directives, thus allowing the execution of Java-code while keeping the logic and the content separated. The following extract from the logicsheet shows the template that is matched with and used instead of the tag *<smartframe:header/>*.

```
<xsl:template match="smartframe:header">
  <xsp:structure>
     <xsp:include>smartframe.DBInterface</xsp:include>
  </xsp:structure>

  <!-- code to read meta-data information from DB like location-->

  <xsp:logic>
    public String elementName;
  ...
```

Another template is applied for the tag *<smartframe:link ...>*, which reads the required meta-data from the database to perform at least a transformation, i.e. the location as well as mime-type of the referenced object. Depending on the type and context, further information has to be requested from different categories, e.g., the requirements from *technical*, the difficulty from *educational*, or the kind of relation from *relation*. Herewith, we are capable to directly react on the user and, therefore, influence the learning experience, e.g., by presenting learning material according to a specific learning level and qualification.

The corresponding XML-files as specified by the location are included; see the pipeline in Figure 4 on the upper right-hand side. Further links to other objects are processed recursively (2, 9) subsequent to the extraction of the sub-documents in the user-specific language (7, e.g., the content of the *<lmml xml:lang="en">*-tags); if not available a default language is used. The extracted sub-documents are serialized (10), i.e. converted to XML, and combined to a single document containing all requested learning objects of the same language.

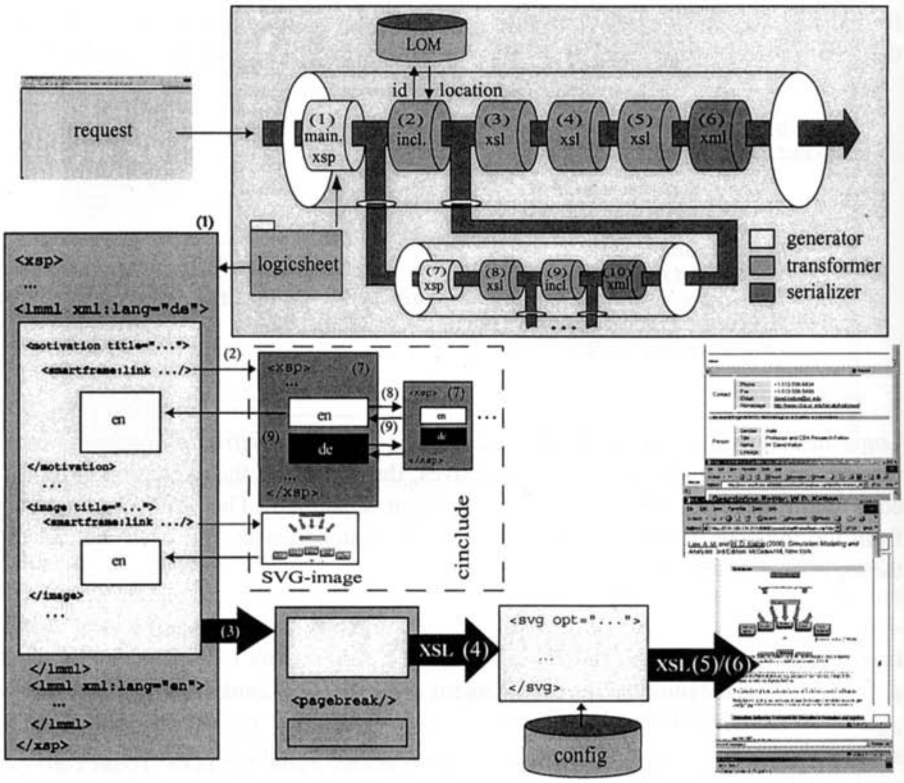


Figure 4: Transformation process of learning material

In the next step the document is analyzed for pagebreaks, and, if needed, decomposed in sub-documents—part of the semantic unit to be displayed on one page—whereas the corresponding part is specified with another parameter (3). Within two further steps the XML-document is transformed to the output-format. The predesign- (4) and design-phase (5) are used for a two-step process, whereas the first one prepares certain structures for the next phase, e.g., inserting Javascripts to be used by the learning objects. In the second phase, the settings of the user are accounted. That is, the user decides, which learning object is shown and in which ways. For example, the object of the type bibliographic entry can either be displayed within a table or as a list of references. The result of the transformation pipeline is serialized according to the requested and from Cocoon supported output format, e.g., for a pdf-document or a web-page.

The learning object *motivation* of our example is transformed to the following html-code:

```
<h3 class="me_text_motivation_titlebar">
```

```
    Motivation: Reference Framework for Simulation in Production
    and Logistics
</h3>
<p class="me_text_motivation_content">
    This course unit highlights the relation between
    simulation and production and logistics.
</p>
```

The design regarding font-type, color of the text, or formatting is defined in user-specific cascading stylesheets (css) that may be edited by the user through an interface and referenced in the html-page by

```
<link rel="stylesheet" type="text/css"
      href="http://www.smartframe.de:8080/cocoon/smartframe/
            static_content/css/treiners.css"/>
```

The following code shows an extract for the title of the motivation being shown with a greenish background and bold black-colored verdana font slightly larger than standard. Further design features are realizable, e.g., a colored frame if a table for the motivation with the corresponding coloring of fields is used.

```
h3.me_text_motivation_titlebar
{
   background-color:#dbe8ac;
   color:#000000;
   font-size:110%;
   font-family:verdana;
   font-weight:bolder;
}
```

Referenced learning objects that are not completely included in the web-page require a special processing. At first, the displayed name for the reference is constructed from the learning object depending on its type as well as the user-specific configuration. In our example (see Figure 1) the bibliographic entry is shown on the web-page as *[Law, A.M. and W.D. Kelton (2000)]* with a small picture of a book. To increase the flexibility of such references, *<span>*-tags with different attributes are used. Depending on the stylefile the display can be changed from showing to hiding the references. Furthermore, these tags can be read by the SmartBars to aggregate the information. For simplification, the Javascript-code for *showBibliography* is not quoted in the following example.

```
<span id="bib" class="hideBib">
   [Law, A.M. and W.D. Kelton (2000)<img
      src="/cocoon/static_content/images/bib.gif"
      class="hideBib"
      onClick="showBibliography('main.xsp?identifier=ident_4')"/>]
</span>
```

3 Graphical User Interface

The graphical user interface (GUI) is built using JavaServer Pages (JSP) for the presentation of the user interface in a browser and Servlets, simple Beans (no Enterprise JavaBeans), and Java classes to implement the application logic. Furthermore, we use a JDBC interface to a relational database (here, we use the free of charge database MySQL), a connection pool as an interface between the database and the user, and mechanisms to handle XML-files. These XML-files are on the one hand used to configure the GUI as well as the underlying logic, on the other hand they contain user-specific settings, e.g., for the transformation process of the learning material.

In the following sections the architecture of the GUI, the configuration of all components, the interface to the learning material as well as the navigation within the selected learning units are described.

3.1 Composition of SMARTFRAME

SMARTFRAME is designed in a 3-tier-architecture (see Figure 5) with a presentation tier, a business tier with business objects (mostly beans) running on a Tomcat Servlet-engine, and the data storage tier, currently a relational database and XML-files using the native file system. The concept considers thoughts about separating business logic from presentation logic, but also extensibility and scalability were of particular importance. Furthermore, SMARTFRAME is executed in the context of Cocoon such that a communication with the transformation process of the learning material is enabled; see Section 3.2.

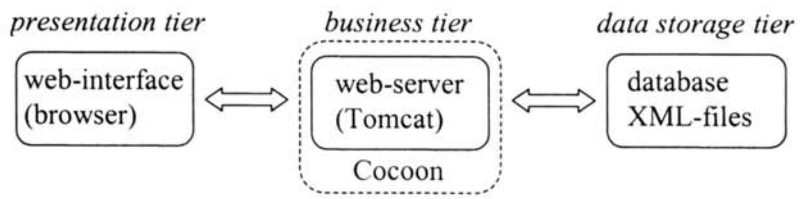


Figure 5: Architecture of the web application

The visualization is done by JSPs, and to guarantee a consistent visualization of SMARTFRAME, they all use the same user-specific stylesheet and are composed using a frameset. That is, the four components as shown in Figure 6 (i.e. title, control bar, menu bar, and content definition) are generated by a specific JSP. This JSP is configured by an XML-file containing *views*, i.e. mappings between the four components and corresponding JSP-files. For example, if the content-page is requested, a menu with all items, a control bar, and a content page with the selected learning unit are shown. The advantage of this design lies in the independence between the JSPs such that each can easily be replaced or renamed by modifying the configuration file; which can also be easily changed to create new mappings.

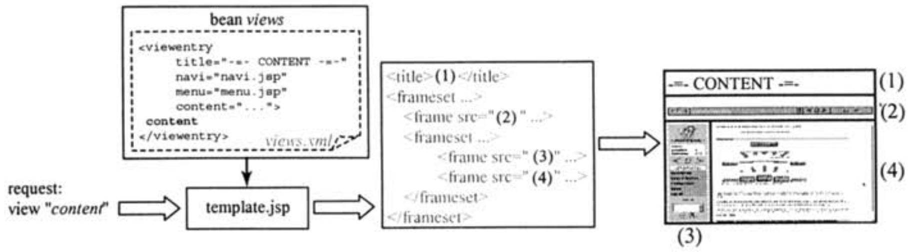


Figure 6: Dynamic frameset generation using *template.jsp*

Within SMARTFRAME certain actions, e.g., identifying the user during the login process (*CheckLoginAction*), initialization of user-specific settings (*InitAction*), or navigation on the learning path (*next* or *previous),* have to be performed. This is done by encoding a URL with the specific action (i.e. the action name with an appended *.do*, e.g. *next.do)* calling a Servlet where the action is mapped to the corresponding method; see also the Jakarta Struts Framework for a similar concept [Apa03c]. Each action either returns a view being displayed as the result of the action within the browser or returns a further action to be executed. The latter one is repeated until the return value is a view. Like views, actions are mapped to Java-classes within an XML-configuration file. Thus, new actions can easily be integrated into SMARTFRAME by adding a combination of a unique name for the action and class reference into the configuration file.

Several beans are used to store data for the period of a session, e.g., user-specific data required to generate the GUI. This concept allows reacting on the user. That is, a bean is triggered to transfer the current data to a database or XML-file whenever the user applies a modified setting of SMARTFRAME.

3.2 Configuration

We have to distinguish several configuration settings. First, global settings for the GUI include the previously mentioned mappings for action-classes and views, color schemes, language, menu structure as well as information about the database access and location of files. Second, user-specific configurations for the GUI cover the preferred language, the position of the menu, the chosen color scheme, and user-data (email-address, username, and password). Furthermore, the user specific configuration concerning the learning material has to be specified and stored in an XML-file being accessible by the transformation process described in Section 2.5. This involves information about, e.g., the incorporation of the Smart-

Bars, the kind of display for bibliographic entries, or how SVG-images are displayed—either directly or being converted into static binary image like gif.

A Servlet initiates beans with *"scope=application"* (i.e. the bean is available application-wide) after the Tomcat Servlet-engine is started and the first request submitted. That is, XML-based configurations for the SMARTFRAME are read and applied to the interface. In Figure 7 the configuration and communication process from the login to the presentation of the learning unit is shown. The user gets the login view being asked for the login name and password. After providing the information, user-specific settings are read either from the database or XML-files and verified with the password (1-3). During the initialization process the temporary file *userconfigfile* to pass configuration information to the transformation process within the Cocoon process is generated. Furthermore, two attributes are stored in the session object: the location of the *userconfigfile* and the CSS-file for the final presentation of the learning material (4). Afterwards, the welcome screen is shown including the menu and control bar (5).

Selecting the menu item *Table of Content* presents a list of available course units (6) from which the user can select the desired course (7). Based on the identifier associated with the first learning element in the learning path, the transformation process as described in Section 2.5 is initiated (8). The resulting web-page—generated according to the user-specific preferences in the configuration file *userconfigfile* and incorporating a reference to the individual CSS-file—is shown in the content frame (9).

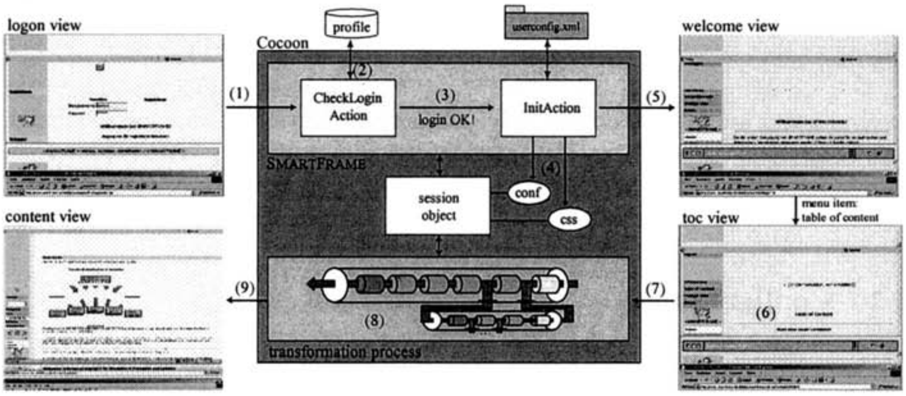


Figure 7: Configuration and communication between the GUI and the learning material transformation within Cocoon

3.3 Navigation and Browsing

The navigation within the learning material is insofar crucial that most VLEs only support linear learning paths and no explorative browsing. Our approach incorporating learning objects allows a larger degree of freedom in designing different views of the learning material. Besides the inclusion of different types of learning objects, the user can flexibly decide on the learning path, i.e. selecting excursions or changing the learning paths where these intersect. Intersections of learning paths especially highlight the relation between different courses supporting the interdisciplinary teaching at universities.

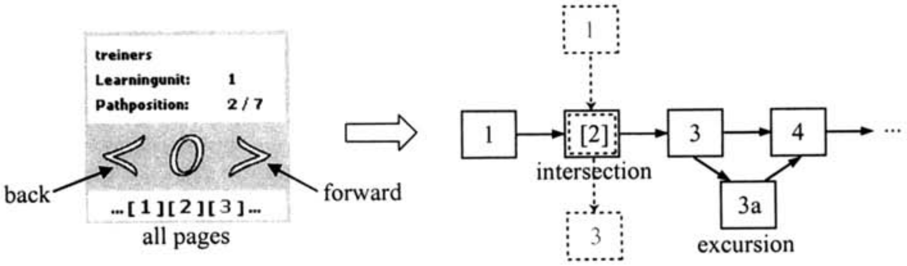


Figure 8:Navigation and learning path

Figure 8 shows the navigation components of SMARTFRAME. Besides going backward and forward on the linear learning path using the arrows, the user can directly select a particular page on the numerical view of the learning path. Visited pages are shown in a different color allowing the user to receive an overview of already visited pages. The button in-between the arrows is used for reload, return from the configuration menu or other views to the current page, and to exit an excursion path to continue on the original learning path.

The navigation is improved using the action mechanism as described before. Depending on the users profile the next button can either show the next learning element on the learning path independent from the status (visited or unknown) or jump to the next so far unvisited page. Furthermore, learning elements might consist of several parts being separated by *<smartframe:pagebreak/>*-tags. In this context, the next button could induce the display of the next part of the learning element. Note that this implies a consideration of the pagebreaks compared to a complete display of the learning element.

4 Conclusions and Outlook

In this paper we have described SMARTFRAME as a prototype of a virtual learning environment, which consists of several components, namely the transformation process of the learning material into different user-specified formats like html and pdf, the graphical user interface as well as an authoring tool. Here, we should emphasize that the transformation process itself could easily be incorporated in various environments like proprietary learning environments. That is, the transformation process requires information about the user-configuration passed in XML-files but could also use default settings resulting in static learning material presentation. On the other hand, the communication between the user interface and Cocoon could also be established using parameters appended to the invoking URL but would be fairly limited due to restrictions concerning the ability and allowance to adapt a proprietary learning environment.

These environments would have to be capable of managing some kind of linear (or arborescent) learning path structure to invoke the appropriate learning element and display the result within a separate frame. This is, e.g., given in WebCT [WebCT03] where the learning elements generated by Cocoon can be referenced by URLs like *http://www.smartframe.de:8080/cocoon/main.xsp?identifier=id_1*. Besides its more general functionalities, the user interface of SMARTFRAME can also be used to display simple static html-pages. Both alternatives are significantly lacking the advanced features that result from the collaboration of the GUI and Cocoon.

Having implemented both, the GUI as well as the transformation of the learning material, within the same technological context using the Apache Tomcat server allows a much greater flexibility. Both components share the same session object so that the user interface is able to assign attributes to it, which can be read out by Cocoon and vice versa. Having the graphical user interface handling administrative tasks like user management and preferences concerning, e.g., technical and didactical aspects, the learning material can be displayed in a user-specific way. The user is able to choose his favorite color-scheme, select his preferred language, and determine whether and where menu bars and components like SmartBars are displayed. Furthermore, the composition of the displayed learning material with respect to LMML-elements can be adapted.

The concept of SMARTFRAME allows the integration of innovative technologies, especially SmartBars and hyperbolic graphs [Rei+02]. This is due to the modular design of both technologies providing several interfaces for integration. In particular the transformation process, which might have to be adapted according to the requirements of new components, is extensible by integrating new transformation rules regarding the configuration of the user.

SMARTFRAME is part of an ongoing research project and, therefore, has to be seen as a prototype missing several essential components to be a fully featured virtual learning environment according to various notions. Mainly communication features including email, message board, or whiteboard, are currently under development. That is, we do not seek for a rudimental implementation just supporting the basic features but develop innovative functionality that, first, increases the learning experience and, second, enables an individual support of the learner during the course being comparable to traditional classroom courses.

On the other hand, the modular design of the learning material should be used to implement different scenarios like the application in the classroom, the self-paced study as well as composing of printed material. Here, synchronized blended learning can be seen as a potential application of SMARTFRAME where several didactical methods are combined within one environment; see also [Fra+03].

References

[ADL03] Advanced Distributed Learning Initiative: SCORM. http://www.adlnet.org. Date of last check 2003-05-10.

[Apa03a] Apache Software Foundation. Cocoon software. http://xml.apache.org/cocoon. Date of last check 2003-05-10.

[Apa03b] Apache Software Foundation. The Jakarta Site – Jakarta Tomcat. http://jakarta.apache.org/tomcat/index.html. Date of last check 2003-05-10.

[Apa03c] Apache Software Foundation. The Jakarta Site – Struts. http://jakarta.apache.org/struts. Date of last check 2003-05-10.

[Aria03] ARIADNE Foundation for the European Knowledge Pool: ARIADNE Educational Metadata Recommendation - V3.2. http://www.ariadne-eu.org/en/publications/metadata/ams_v32.html. Date of last check 2003-05-10.

[Bal+03] Balci, O.; Gilley, W.S.; Adams, R.J.; Tunar, E.; Barnette, N.D.: Online Interactive Modules for Teaching Computer Science. http://courses.cs.vt.edu/~csonline. Date of last check 2003-05-10.

[Black03] Blackboard: Blackboard. http://www.blackboard.com. Date of last check 2003-05-10.

[Fra+02] Frank, C.; Kassanke, S.; Suhl, L.: Meeting students expectations and realizing pedagogical goals within the development of a virtual learning environment. In Proceedings of Elearn, World Conference on E-learning in Corporate, Government, Healthcare, & Higher Education, pp. 1124-1129, Montreal, Canada, 2002.

[Fra+03] Frank, C.; Reiners, T.; Suhl. L.; Voß, S.: Conceptual Framework for Synchronized Blended Learning on the Web. Technical Report, University of Paderborn. 2003.

[IEEE03] IEEE Learning Technology Standards Committee: Standard for information technology: Education and training systems – learning objects and metadata. http://ltsc.ieee.org/wg12. Date of last check 2003-05-10.

[IMS03a] IMS Global Learning Consortium: IMS Learning Resource Meta-data Specification. http://www.imsproject.org/metadata/index.cfm. Date of last check 2003-05-10.

[IMS03b] IMS Global Learning Consortium: IMS Question & Test Interoperability Specification. http://www.imsproject.org/question. Date of last check 2003-05-10.

[Met03] Methodenlehre-Baukasten. http://www.izhd.uni-hamburg.de/mlbk/home.html. Date of last check 2003-05-10.

[Orw03] OR-World. http://www.or-world.com. Date of last check 2003-05-10.

[ReVo03] Reiners, T.; Voß, S.: Some Thoughts on how to Teach OR/MS. Technical Report, University of Technology Braunschweig. 2003.

[Rei⁺02] Reiners, T.; Reiß, D.; Voß, S.: Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts. In Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002), Volume #100029-03-TR-026, Millet Alberta, pp.1–7. ICSC-Naiso Academic Press. [ISBN: 3-906454-31-2], 2002.

[Rei⁺03a] Reiners, T.; Reiß, D.; Sassen, I.; Voß, S.: Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: Von der Technik zur Anwendung. Accepted for publication in Proceedings of the Workshop E-Learning in Wirtschaftsinformatik und Operations Research, 2003.

[Rei⁺03b] Reiners, T.; Reiß, D.; Voß, S.: XML-basierte Kodierung von Lernobjekten. Technical Report, University of Technology Braunschweig. 2003.

[Schö03] Schöning, H.: XML und Datenbanken: Konzepte und Systeme. Hanser, München, 2003.

[Smar03] SMARTFRAME. http://www.smartframe.de. Date of last check 2003-05-10.

[SnBy03] Sniedovich, M.; Byrne, A.: tutOR. http://www.tutor.ms.unimelb.edu.au/frame.html. Date of last check 2003-02-01.

[Sues03] Süß, C.: Learning Material Markup Language. http://www.lmml.de. Date of last check 2003-05-10.

[W3C03a] W3 Consortium: XSL Transformations, http://www.w3.org/TR/xslt. Date of last check 2003-05-10.

[W3C03b] W3 Consortium: Scalable Vector Graphics, http://www.w3.org/TR/SVG. Date of last check 2003-05-10.

[WaMu99] Walsh, N.; Muellner, L.: DocBook: The Definitive Guide. O'Reilly, Beijing, 1999.

[WebCT03] WebCT: WebCT.com. http://www.webct.com. Date of last check 2003-05-10.

Meta-Heuristiken in virtuellen Lernumgebungen

Torsten Reiners, Imke Sassen und Stefan Voß

Technische Universität Braunschweig, Institut für Wirtschaftswissenschaften, Abteilung ABWL, Wirtschaftsinformatik und Informationsmanagement, Abt-Jerusalem-Str. 7, D-38106 Braunschweig, {t.reiners, i.sassen, stefan.voss}@tu-bs.de.

Zusammenfassung. In der Lehre gewinnt der Einsatz von virtuellen Lernumgebungen an Bedeutung, wobei für das Operations Research aufgrund der Anforderungen an eine praxisorientierte Ausbildung ein erhöhter Bedarf an qualitativ hochwertigen Lernangeboten besteht. Anhand von konkreten Lerninhalten soll aufgezeigt werden, auf welche Weise eine Verknüpfung der theoretischen Grundlagen mit praktischen Anwendungen hergestellt werden kann. Hierzu erarbeiten die Lernenden grundlegende Begrifflichkeiten und Zusammenhänge, die in einem weiteren Schritt anhand von realen Problemstellungen aus einer neuen realitätsbezogenen Perspektive betrachtet und dadurch vertieft und gefestigt werden.

Einleitung

Angesichts einer steigenden Anzahl von Studierenden, verringerten finanziellen Ressourcen der Ausbildungsinstitutionen und darüber hinaus der Forderung nach der Bereitstellung von Angeboten zum lebenslangen Lernen, bieten virtuelle Universitäten weitere Möglichkeiten zur Förderung der Qualität in der Lehre und werden aus diesem Grund von zahlreichen Ländern in ihr Bildungskonzept integriert (siehe Hazemi et al. (1998)). In diesem Beitrag soll der Bereich Operations Research in Bezug auf die Darstellung und Präsentation in virtuellen Lernumgebungen (VLU) am Beispiel der Meta-Heuristiken näher betrachtet werden. Da gerade im Operations Research eine praxisorientierte Lehre zur Aneignung von Kenntnissen und Fähigkeiten in der Modellierung und Lösung von (mathematischen) Problemstellungen, Algorithmen, Softwareentwicklung sowie Projektmanagement notwendig ist, sind die Entwicklungen der Forschung hinsichtlich neuer didaktischer Methoden und virtueller Lernumgebungen zu verfolgen und umzusetzen. Insbesondere Meta-Heuristiken, als zeitgemäße und wichtige Werkzeuge zur Optimierung, haben in der Lehre sowohl in Lehrbüchern als auch in praktischen Umsetzungen ein Schattendasein und sind nur mangelhaft vertreten, so dass traditionelle Präsenzveranstaltungen im Bereich OR/MS sich nur unzureichend mit Meta-Heuristiken auseinandersetzen. Daraus und auch aufgrund fehlender guter Beispiele aus der Praxis ergibt sich eine unzureichende Vermittlung der wesentlichen Vorteile von Meta-Heuristiken gegenüber anderen Methoden der Optimierung, wobei hier bekannte Algorithmen wie z.B. die genetischen Algorithmen ausgenommen sind.

Meta-Heuristiken besitzen in vielerlei Hinsicht Analogien zu weiteren Gebieten des Operations Research. Beispielsweise operiert der Simplex-Algorithmus (siehe u.a. Domschke und Drexl (2002)) zur Lösung von linearen Optimierungsproblemen durch das gezielte Traversieren eines den Lösungsraum beschreibenden Polyeders. Analog gehen z.B. Verfahren zur Lösung von Transportproblemen vor, bei denen im Anschluss an eine Eröffnungsheuristik zur Ermittlung einer zulässigen Lösung Verbesserungsverfahren wie die MODI-Methode (siehe z.B. Domschke (1995)) auf stark strukturierten Lösungsräumen zum Einsatz kommen. Für die Studierenden ergeben sich durch die Behandlung von Meta-Heuristiken aufgrund von Analogien der Prinzipien zur Lösung von Optimierungsverfahren aus fachdidaktischer Sicht weitreichende Transfer- und Anwendungsmöglichkeiten. Daher bietet sich eine inhaltliche Behandlung der lokalen Suche sowie Meta-Heuristiken im Rahmen der Lehre an. In diesem Beitrag stellen wir einen einführenden Überblick zu unserem Konzept vor, für eine ausführlichere Darstellung siehe Reiners und Voß (2002).

Neben dem einfachen Angebot von Lehrmaterialien ohne eine weitere mediendidaktische Aufbereitung existieren bereits Lernumgebungen im Internet, die weiterführende Präsentationsformen als Hypertexte verwenden. Ein Beispiel für Offline-Software ist OR Welt (2002); frei nutzbare Lernumgebungen sind tutOR von Sniedovich und Byrne (2002) und darauf basierend tutORial von der IFORS (2002). Kommerzielle Systeme, im Wesentlichen zur Unterstützung der Lehre und der Verwaltung von Kursmaterialien, sind u.a. WebCT (2002). In einigen Ländern wie den USA oder Australien ist es auch üblich, anstelle web-basierter Lernumgebungen urheberrechtlich geschützte Software wie Microsoft Excel im Unterricht zu verwenden; siehe z.B. Bell (2000). Gemein ist den Ansätzen, dass das Angebot im Bereich Meta-Heuristiken zurzeit höchstens auf statische untereinander verlinkte Texte oder auf unstrukturierte Sammlungen interaktiver Applets ohne weitere theoretische Ausführungen begrenzt ist. Weitere Darstellungsformen von Optimierungsergebnissen sind z.B. bei Jones (1995) zu finden.

Darüber hinaus fehlt in aktuellen Ansätzen eine ausreichende Integration von Problemstellungen aus der Praxis unter Einbindung interaktiver Darstellungsmöglichkeiten; siehe Wolsey (1979) oder Reisman (1997): "This profession currently has more algorithms than applications".

Aufbau der virtuellen Lernumgebung

Für die interne Repräsentation eines Kurses über Meta-Heuristiken soll ein weiterführendes Konzept für eine VLU verwendet werden; insbesondere die Unterteilung in semantische Module unterschiedlicher Größe soll eine dynamische Darstellung bezüglich der Konfiguration durch Studierende ermöglichen, um Bedürfnisse der Lernenden zu erfüllen und dadurch insgesamt eine Motivationssteigerung zu erzielen. Der Einsatz weiterer Komponenten soll der Schwierigkeit Rechnung tragen, im Bereich Operations Research dauerhaft Interesse und Motivation der Lernenden zu erzielen und die Teilnahme an web-basierten Kursen zu steigern, wobei keine Unterscheidung zwischen der Art der Lernenden (z.B. „Distance-Lerner“ oder „Lifelong-Lerner“) gemacht werden soll. Hierzu erfolgt die Einbindung einer relationalen Datenbank zur Verlinkung der Lernobjekte unter

Ausnutzung der dort gespeicherten Meta-Informationen. Dies erlaubt eine Adaption an Studierende und ihre Bedürfnisse hinsichtlich Lerngewohnheiten sowie den Einbezug von verschiedenen Kommunikationsformen. Eine detaillierte Beschreibung der konzipierten Architektur ist in Reiners et al. (2002a), ein Aufbau der Lehrmaterialien in Reiners et al. (2002b) gegeben.

Das dargestellte Lehrangebot besitzt den Anspruch, dass Lernende ohne Vorkenntnisse später eigenständig Meta-Heuristiken zum Lösen komplexer praktischer Problemstellungen einsetzen können. Hierzu werden grundlegende Prinzipien in einer vorgelagerten Kurseinheit durch interaktive Beispiele vermittelt. Eine Meta-Heuristik ist definiert als ein iterativer Generationsprozess, der eine untergeordnete Heuristik steuert und eine Methode darstellt, ein Optimierungsproblem in adäquater Zeit zu lösen. Jedes Optimierungsproblem besitzt einen speziellen Lösungsraum, der hinsichtlich der Restriktionen alle zulässigen (und ggf. darüber hinausgehende nicht zulässige Lösungen, wie sie sich z.B. vermöge geeigneter Problemrelaxationen ergeben) enthält. Der Lösungsraum ist jedoch in den meisten Fällen zu groß, um die optimale Lösung durch eine vollständige Enumeration zu finden bzw. eine derartige Vorgehensweise in der vorgegebenen Zeit zu realisieren. Daher grenzen Meta-Heuristiken durch die intelligente Kombination verschiedener Methoden den Lösungsraum auf bestimmte Bereiche ein und verwenden Lernstrategien, um Informationen zu strukturieren und effektiv optimale oder gute Lösungen zu finden, siehe Voß et al. (1999).

Die grundlegenden Komponenten von Meta-Heuristiken werden zur Erklärung von speziellen Algorithmen wie Steepest Descent, Simulated Annealing oder Tabu Search genutzt. Der Lernende wird hierbei durch einfache interaktive Beispiele unterstützt, welche die zugrunde liegenden Prinzipien veranschaulichen und das Verhalten der verschiedenen Methoden verdeutlichen. Abbildung 1 zeigt die Schritte einer interaktiven Animation zur Verdeutlichung einer Nachbarschaft. Die Interaktion für die Studierenden liegt z.B. darin, dass der nächste „beste" Nachbar gewählt werden muss, wobei eine Bewertung durch die VLU erfolgt. Über die Berücksichtigung von Analogien lassen sich hier wiederum Kenntnisse z.B. über den Simplex-Algorithmus einbinden (sowie umgekehrt die Vorgehensweise des Simplex-Algorithmus als „einfache" lokale Suche auf einem gut strukturierten Lösungsraum erläutern). Betrachtet man die Vorgehensweise des Bestimmens einer ersten zulässigen Lösung und die daraufhin folgende Anwendung eines Verbesserungsverfahrens, wie sie im Bereich der Meta-Heuristiken i.d.R. angewendet wird, so besitzt dieses gleichermaßen eine Korrespondenz in der so genannten Zweiphasen- oder M-Methode.

Abb. 1 Darstellung einer Nachbarschaft für eine vorgegebene Lösung

Aufbauend auf den grundlegenden Prinzipien sollten Studierende erworbenes Wissen in einer komplexeren und in weiten Teilen realistischen Problemstellung anwenden, um eine Intensivierung der Materie zu erreichen. Hierzu können Problemstellungen von Forschungsprojekten oder aus der Praxis eingesetzt werden. Unterstützt durch die virtuelle Lernumgebung entwickeln die Studierenden eine Lösungsmöglichkeit unter der Verwendung bestehender und konfigurierbarer Softwarepakete. Diese Software kann entweder ein kommerzielles Produkt mit einer graphisch aufbereiteten Benutzeroberfläche zur Eingabe des Problems und mit Parametern für die Lösungsalgorithmen sein, oder aber eine Softwarebibliothek, die z.T. wieder verwendbare Codes enthält, aber dennoch auch einige Programmierkenntnisse voraussetzt. Um eine weitgehenden Integration in unsere virtuelle Lernumgebung ohne Medien- oder Technologiebrüche zu gewährleisten, verwenden wir HOTFRAME (Heuristic OpTimization FRAMEwork) von Fink und Voß (2002). Ein Benutzerinterface ermöglicht die Konfigurierung der wesentlichen Komponenten der Meta-Heuristiken. Außerdem wird ein automatisch generierter Source-Code ausgegeben, mit Todo-Teilen für die Lernenden, die in ihrem Schwierigkeits- und Komplexitätsgrad variabel sind. Schließlich stellt die VLU ein Experimentierfeld zur Verfügung, in dem das erreichte Ergebnis an diversen Problemstellungen mit verschiedenen Parametereinstellungen der Meta-Heuristiken getestet werden kann. Die Resultate werden gesammelt und können von den Lernenden durch den Einsatz statistischer Methoden und Darstellungsmöglichkeiten eigenständig evaluiert werden (siehe auch Abbildung 2).

Ausblick

Auch hochgradig interaktive Lernumgebungen unterstützen zumeist nicht die Idee, dass Studierende erworbenes Wissen im Rahmen eines bestimmten Forschungsfeldes anwenden können, um Fragestellungen der Praxis oder aktuellen Forschung zu untersuchen. Gerade die Erfahrung von Praxisrelevanz und Anwendungsmöglichkeiten des erlernten Wissens kann jedoch hochgradig zu einer Steigerung der Motivation der Lernenden beitragen. Darüber hinaus kann aufgrund der Möglichkeit der eigenständigen Evaluation des erreichten Ergebnisses bei Studierenden der Anspruch entstehen, ihr Ergebnis noch weiter zu verbessern, was ebenfalls zu einer weiteren Auseinandersetzung mit den Lehrinhalten führt. So beinhaltet das Generieren und Testen eigener Lösungen unter lernpsychologischen Gesichtspunkten lernförderliche Momente, da durch den Wechsel von Hypothesengenerierung und -testen das entdeckende und explorierende Lernen unterstützt wird und Lösungen beliebig oft konstruiert werden können. Im Vergleich zu anderen Methoden werden Selbstlernprozesse besonders gefördert, da auch eigenständige Fehlersuche und deren Beseitigung in schwierigen Programmteilen durch die Möglichkeit der Selbstevaluation motiviert werden, aber optional auch Hilfestellungen von Seiten der Lernumgebung gegeben werden können. Gerade bei den Anfängen des Transfers von erlerntem Wissen auf das Lösen komplexer Problemstellungen sollten die Lernenden sich nicht allein gelassen fühlen, sondern im gewünschten Maße Unterstützung in Anspruch nehmen können. Vor allem die Erfahrung, die bereitgestellte Software in richtiger Weise zum Lösen komplexer Problemstellungen eingesetzt zu haben, gibt den Studierenden auf der einen Seite

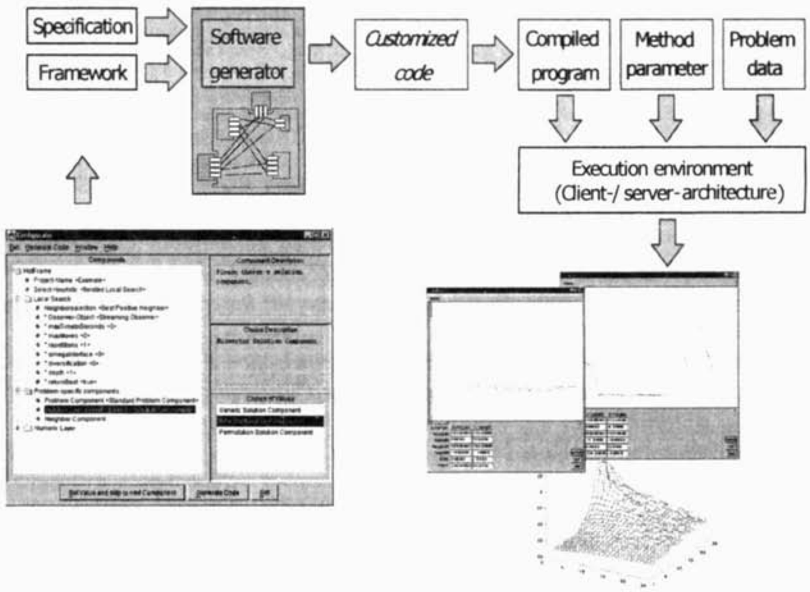


Abb. 2 Anwendungssequenz einer Meta-Heuristik

Aufschluss über ihren Lernerfolg und versichert ihnen auf der anderen Seite auch den Wert des Gelernten, sowie dessen interdisziplinären Verwendungsmöglichkeiten, entgegen dem Vorurteil, dass sich mit Methoden des Operations Research nur in theoretischen Anwendungen gute Ergebnisse erzielen ließen.

Danksagung

Diese Arbeit wurde von dem BMBF (Bundesministerium für Bildung und Forschung) in dem Programm „Neue Medien in der Bildung" (08NM094D) unterstützt.

Literatur

Bell, P.C., 2000. *Teaching Business Statistics with Microsoft Excel.* INFORMS Transactions on Education. 1(1). http://ite.informs.org/Vol1No1/ bell/bell.html.

Björck, U., July 2002. *List of Virtual Universities.* http://www.ped.gu.se/ulric/vus.html.

Domschke, W., 1995. *Logistik: Transport.* 4. Auflage. Oldenbourg, München.

Domschke, W. und A. Drexl, 2002. *Einführung in Operations Research.* 5.Auflage. Springer, Berlin.

Fink, A. und S. Voß, 2002. *HotFrame: A Heuristic Optimization Framework.* In: S. Voß und D.L. Woodruff (Hrsg.). *Optimization Software Class Libraries.* Kluwer, Boston. 81–154.

Hazemi, R., S. Hailes und S. Wilbur (Hrsg.), 1998. *The Digital University: Reinventing the Academy.* Springer, London.

IFORS, July 2002. *tutORial.* http://www.ifors.ms.unimelb.edu.au/tutorial.

Jones, C.V., 1995. *Visualization and Optimization.* Kluwer, Boston.

ORWelt, July 2002. *ORWelt.* http://dsor.uni-paderborn.de/de/forschung/wbs/orwelt.

Reisman, A., 1997. *Flowshop Scheduling/Sequencing Research: A Statistical Review of the Literature, 1952–1994.* IEEE Transactions on Engineering Management. 44, 316–329.

Reiners, T. und S. Voß, 2002. *Teaching Meta-Heuristics.* Working Paper, Technische Universität Braunschweig.

Reiners, T., D. Reiß und S. Voß, 2002a. *Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts.* Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002). ICSC-Naiso Academic Press, Millet Alberta [ISBN: 3-906454-31-2], #100029-03-TR-026, 1–7.

Reiners, T., D. Reiß und S. Voß, 2002b. *XML-basierte Kodierung von Lernobjekten: Dokumentation zu VORMS.* Arbeitspapier. Technische Universität Braunschweig, in Vorbereitung.

Sniedovich, M. und A. Byrne, July 2002. *tutOR.* http://www.tutor.ms.unimelb.edu.au/ frame.html.

VORMS, July 2002. *Virtuelles Studienfach Operations Research/Management Science.* http://www.vorms.org.

Voß, S., 2001. *Meta-Heuristics: The State of the Art.* In: A. Nareyek (Hrsg.). *Local Search for Planning and Scheduling.* Lecture Notes in Artificial Intelligence 2148. Springer, Berlin, 1–23.

Voß, S., S. Martello, I.H. Osman und C. Roucairol (Hrsg.), 1999. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization.* Kluwer, Boston.

WebCT, Inc., July 2002. *WebCT.com.* http://www.webct.com.

Wolsey, R.E.D., 1979. *Pragmatism Triumphant or Past Sophistication and Future Elegance.* In: K.B. Haley (Hrsg.). *Operational Research '78.* North Holland, Amsterdam. 80–86.

Using Hyperbolic Trees and SmartBars within Virtual Learning Environment Concepts

Torsten Reiners, Dirk Reiß, Stefan Voß

Department of Business Administration, Information Systems, and Information Management
Braunschweig University of Technology
Abt-Jerusalem-Str. 7, 38106 Braunschweig, Germany
{t.reiners, d.reiss, stefan.voss}@tu-bs.de

Within the last years several projects were initiated to use modern information technology to integrate virtual learning environments in the existing so-called presence or new virtual universities. While virtual learning environments are developed in projects at various universities new and innovative ideas are still necessary. In this paper we suggest a browser-based concept for location independent learning with highly configurable components by demonstrating how existing technologies like XML encoded learning material or hyperbolic trees can be combined and endorsed with an innovative way of information representation using SmartBars. Our concept demonstrates how to support an individually and flexibly configurable learning environment with concentration on factors like usability and navigation whilst keeping the social aspect of inter-personal communication.

1. Introduction

Modern information technology allows further variations of education besides the traditional presence university. Especially in the context of growing numbers of students, fewer funds for education, and the desire to have a lifelong learning support, virtual universities might be one opportunity to support the education and, therefore, are currently opened in many countries using different strategies for their educational program (Hazemi et al. (1998)). These programs might provide only online-access to all course materials (e.g., Massachusetts Institute of Technology (2001)), but can also realize a full digital or virtual university where the learner is regularly enrolled as a student (e.g., University of Phoenix (2001), see also Björck (2001) for a list of virtual universities). The main goal of virtual universities is the provision of education with the same or better quality as in presence universities using virtual learning environments where the learner is navigating through the content while having access to services like communication with lecturers or teaching assistants or participating at tests.

Even a superficial search in the Internet can reveal that almost every university is working on larger projects, mostly together with further partners from educational institutions or companies. Even if the search is concentrated on projects that are developing solutions being available for the appliance in education without further financial investments for the students or lecturer, the number is still enormous; examples at German universities are Virtueller Hochschulbund Karlsruhe (2001) and Learning Lab Lower Saxony (2001). Most of these learning environments use an interface to access revised lecture notes in HTML (Hypertext Markup Language) using hyperlinks to improve the usability as well as embedded Java-Applets (e.g., University of Melbourne (2001)) whereas other systems like ORWorld (2001) or Multibook (2001) have an advanced foundation due to the usage of XML (Extensible Markup Language) for the coding of the content but still do not provide enough flexibility to the user interface.

One major problem is to obtain the interest as well as develop and support the motivation of the learner – in this context we do not distinguish between the kind of learner, e.g., the distant learner, the enrolled learner, or the lifelong learner (Wedemeyer (1981)) – and to find a demarcation to other projects implementing a virtual learning environment (VLE). Especially factors like location independent learning, navigation, communication, information presentation, and configuration are not extensively integrated in most system architectures.

The focus of this paper is to describe a system architecture together with special components with the objective to improve the learning process of learners as well as the development of new learning units by authors or lecturers. The authors have to code learning objects of the VLE in XML and, therefore, are independent from the later visualization and can mainly concentrate on the content. The only structural elements they have to provide are information about (cross-) references within the learning material (e.g., literature, glossary entries, or other learning materials) as well as the combination of the different learning objects to larger learning units or structured documents. For the visualization of learning material and the learning process itself, we are going to demonstrate a concept for a browser-based framework that supports an individually and flexibly configurable learning environment with concentration on factors like usability and navigation whilst keeping the social aspect of inter-personal communication. In this context, we introduce two innovative ideas for VLE to support the learner: hyperbolic trees and SmartBars. The former allows the navigation in the learning space including a history function as well as the visualization of relations between learning units, the latter are used to display information within the text in a concentrated graphical way.

Furthermore, the possibility to configure the learning environment to the learner's requirements for a pleasant learning experience has to be integrated in the VLE in a way that the configuration can be accessed independently from the location, time, and the connection mode. This is not given in most VLE due to the fact that they are either based on a client/server model where the learner has to be online while using the system, or distributed on a transportable device like CD or DVD, having the disadvantage of being bonded to a location. We suggest a consolidation of both modes by using a replication mechanism for the learning content, changes and annotations to the learning material as well as the learner profiles (see Section 3.1 for a representation of the system architecture).

In Section 2 we describe the structure of the learning material as well as the system architecture consisting of a server and client site, which are both based on software that is either open-source or available under an educational license. Section 3 introduces the prototype for a user interface by describing the possible interaction and configuration for a potential learner as well as further navigation and communication principles that can be integrated into a virtual learning environment. Section 4 finishes the paper with a short conclusion and outlook.

2. Structure of the Underlying Database

The content of the VLE has to be encoded using a certain standard. In this paper, we describe an XML-based approach using several standards and a hierarchical structure. The hierarchy mainly consists of learning objects, learning units constructed from learning objects, and structured learning documents built from learning units. Due to space limitations, it is not possible to describe the used standards and structures with the required details but instead we have to refer to other sources.

2.1. Learning Objects

Learning objects (LO) are small objects describing the content of the VLE. For a consistent appearance of the VLE, the author should follow given guidelines for the writing style, the nomenclature, the design, and the structure of a specific VLE. The size and type of the content of the LO might be fixed by the type of the LO but the author can customize this under the objective that the granulation is to be set according to a planned reusability of the LO. Especially large LO are hardly reusable in other contexts whereas small LO (e.g., one sentence or paragraph) increase the complexity to build and handle the LO. Furthermore, the design of LO should consider that they can only be reused if certain directives are followed for all LO in the VLE. They should, e.g., be written without referencing the context, which means that phrases like "as seen in module X" or "we will describe this in module Y" should be substituted by simple references to other LO. The visualization of references to, e.g., figures within the learning content can be done using standard phrases like "(see Figure X)". Each learning object – as well as the hierarchically constructed learning units and structured learning documents – is wrapped by a learning object metadata wrapper (LOM wrapper), a certified standard to specify the features and content of LO. LOM is a standard by the IEEE (IEEE Learning Technology Standards Committee (2001)) but there are also other standards like ARIADNE (ARIADNE Foundation (2001)), which is based on LOM, SCORM (Cover (2001)), and IMS (IMS Global Learning Consortium (2001a)). For each LO different categories are used to keep information about the content itself, the lifecycle (i.e., the history of changes or its ancestors in case of inheritance from other LO), technical and educational specifications as well as descriptions of rights, relations, classifications, and annotations. The categories contain several elements, e.g., to classify the LO in respect to the learning level (beginner, intermediate, expert) or type of audience (undergraduate learner, graduate learner, practitioner) by using an appropriate and defined vocabulary and level of detail.

Technically, learning objects can contain components like text, tables, figures, glossary entries, media objects like applets or movies, exam questions, or mathematical formulas. LO have to be encoded in a structured and standardized format so that they are independent from the visualization. Standard operations like search can be used, and the LO can be integrated in different VLE based on a similar standard. If the LO are coded non-sensitive to their context, they can be used at different places, e.g., a picture of a car can be used to describe the production process but also in a simulation of an intersection within a traffic system. LO are implemented in

XML using several established standards to encode different types of content. MathML (W3 Consortium (2001)) is a standard for the representation of equations, QTI (IMS Global Learning Consortium (2001b)) for the representation of questionnaires and multiple choice tests, and DocBook (Walsh and Muellner (1999)) for bibliographic information and publication structures. These standards might be adapted to a certain VLE (project specific standards) due to the requirement of topic-specific fields; other features are part of XML or have to be defined for the individual project in advance during the development of the database.

2.2. Learning Units and Documents

Learning objects represent the smallest units within the database and are mainly used to be combined to larger documents in different contexts. As shown in Figure 1 the LO are used to construct larger learning units (LU), which are comparable to pages of a browser or sections/paragraphs within a book. In a final step, these LU are used to build structured learning documents (SLD) resembling courses, books, or guided tours.

Usually learning objects are not directly applicable due to the textual consistency with other LO. Therefore, a new LO should be derived from the best fitting LO and adopted by performing the textual changes. Within the database the history of derived LO is recorded together with the original author and the changes. References to other LO are possible, e.g., to cite bibliographic items, which are also presented as LO. The LU are containers for several LO that can be combined to SLD.

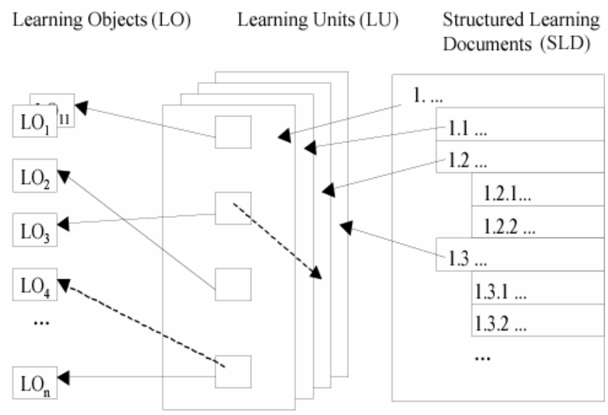


Figure 1: Structure and connection of learning objects, learning units, and structured learning.

Within an authoring tool, which has access to the LOM-database and therewith to the LO themselves, the author of any type of learning material, e.g., LO or SLD, is assisted to combine several LO to a LU or to inherit a new LO from an existing one without having to know the internal representation. Furthermore, the authoring tool would provide a convenient interface to search- or browse-mechanisms through which the author can get a survey of relevant LO she eventually wishes to combine or inherit from.

2.3. System Architecture

The system architecture as shown in Figure 2 uses a client/server-model with databases (e.g., MySQL (MySql.org (2001)) for the storage of LOM-Wrappers including links to the contents (LO, LU, SLD) being stored in documents, navigation information, news and discussion, and the learner profiles, which are kept primarily on the server. Using a web-server (e.g., Apache (Apache Software Foundation (2001b)) and software to convert the XML-based LU (e.g., Cocoon (Apache Software Foundation (2001a) in connection with the Tomcat-Servlet-Engine (Apache Software Foundation (2001c))) into the required format, the learning material is visualized based on the settings of the learner – by using an adaptation mechanism for the style files. Besides working online, a special offline-mode is supported by running a local Java-based engine to perform the access and transformation tasks on a local excerpt of the content database, which is partly downloaded based on the selected LU or SLD. Furthermore, the navigation database is transferred to the client and a local data storage for the learner profiles is used and later replicated to the server as soon as the learner is online again. While being offline advantages of an Internet (or online) VLE like the communication with other learners and the continuous update with news and mail are not available.

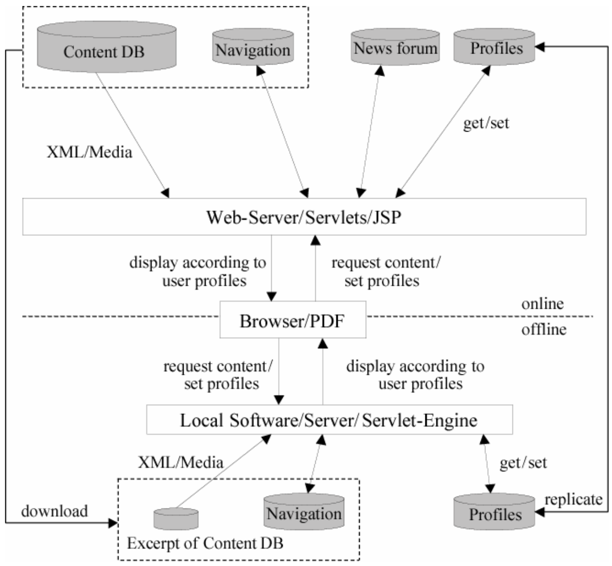


Figure 2: System architecture for the VLE.

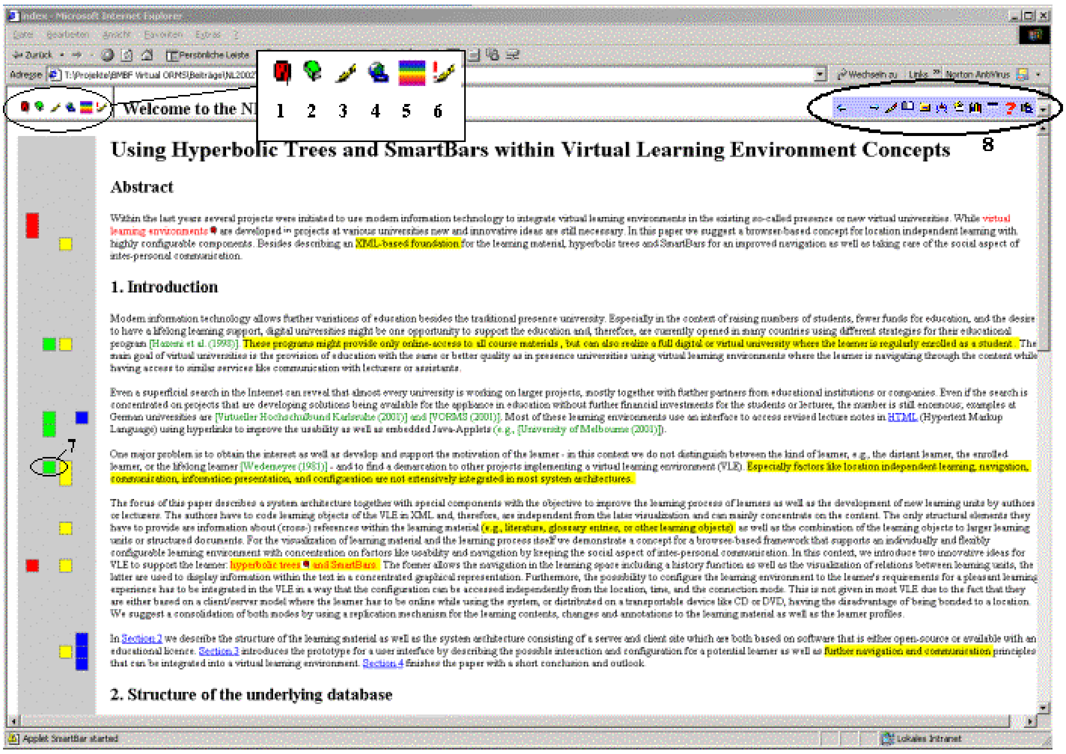


Figure 3: Prototype of the browser-based Virtual Learning Environment.

Another focus for the online-mode has to be the access time – long loading times would cause a decrease of the motivation level of the learner – and on the size of transferred learning materials. Especially for LO containing software like applets demonstrating algorithms, the developer has to consider the download time, the resources for execution on the client (e.g., CPU or memory) as well as compatibility with other software.

3. Browser-based VLE

3.1. Visualization

The VLE is displayed within standard browsers (e.g., Internet Explorer or Opera), which are available for several platforms (see Figure 3 for a screenshot of the VLE within the Internet Explorer). The page is divided into several frames, with a large main-frame for the content and further smaller sub-frames for toolbars, information presentation, and navigation. The top frame is used to display information like news, results, or statistics, depending on the current mode. The toolbar (8) on the right-hand side is used to select integrated functionality like navigation within the VLE or opening an extra window with a presentation of the hyperbolic tree as described in Section 3.3. A further sub-frame in the left top corner – not shown in the screenshot of the prototype – could be used for a logo or computer assistant – presenting helpful comments, memory hooks, or motivation phrases – as well as a colorized marker to show the current topic of the learning material in the main-frame.

One of the innovative aspects of this concept for the VLE is the integration of so-called SmartBars on the left-hand side of the main-frame to aggregate information about the content. In most VLE information about, e.g., links, glossary, or bibliography is given in colorized or underlined text as well as included symbols, which interfere with a fluent reading process. Using SmartBars, the special information within the content of the main-frame is aggregated for the whole line in a corresponding part of a SmartBar – also called SmartMark – which is drawn on the left side. Depending on the kind of information, the learner can access the information by having several options. Whenever the mouse is moved over the SmartMark, the information is displayed in the learning material according to their type, e.g., literature is shown by inserting the author names and the year of the publication and links to the

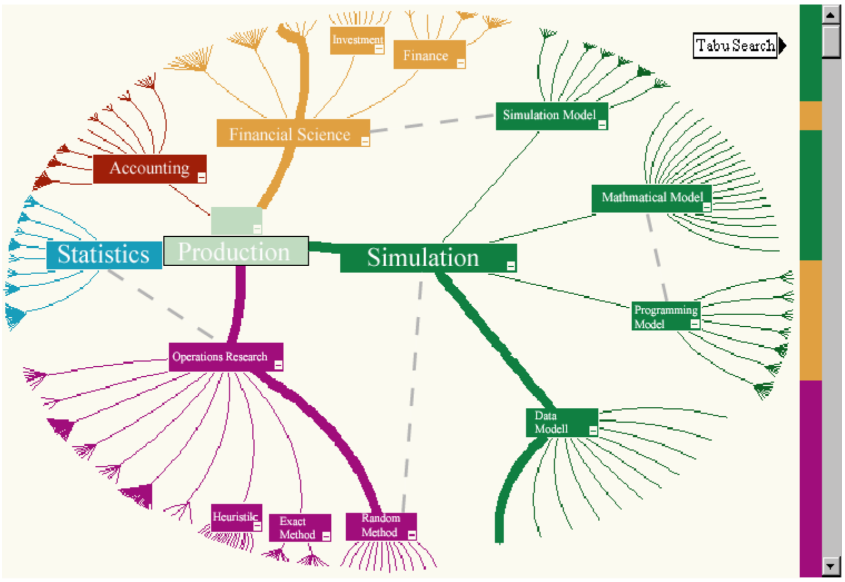


Figure 4: Hyperbolic tree and history of last pages

glossary by drawing a small symbol behind the corresponding word, which is also changing its color.

While the previous action shows the information only temporarily, performing a mouse click on a SmartMark toggles the visibility and interactivity in the content until a further action for this particular information is executed. The activated parts in the content lead to further information by opening either a context sensitive menu or another window with the content of the referred subject. For example, the SmartMark (7) in Figure 4 displays the literature in the content that can be used to open a new small window with the complete bibliographic reference. The prototype includes four SmartBars, which aggregate information or references to index or glossary entries (red or (1)), literature (green or (2)), highlighted text by the learner (yellow or (3)) as well as links within the VLE or to external sources (blue or (4)). Further SmartBars could be used for indication of annotations, news, or discussion boards as well as footnotes. Besides interacting with a single SmartMark, the learner can use the icons above the SmartBars to turn on or off a complete type of information ((1)-(4)), toggle the complete information (5), or highlight certain text in the content and add a SmartMark (6), similar to the use of magic marker within printed documents. Within the VLE, the learner will be able to highlight in different colors, start discussion on a certain subject, or add notes.

3.2. User profiles

Whenever information about a learner is stored on a server, the learner has to be convinced that it is in his own interest and that the data is kept confidential and not used for an evaluation of the learner without his knowledge. If the learner works offline the collected data can only be accessed by the learner himself. This is not the case if the online-mode is chosen. On the other hand, the offline-mode has disadvantages like not having access to the current version, news, and discussion. Stored data for the learner can be the following: information about the learner, configuration of the VLE (e.g., colors, display of information within the text, learning level), current position within a SLD, bookmarks, notes to a special LU (notes can be written for every part of the content), highlighted text (the learner can select a text that will be highlighted), results and statistics of exams, exercises, homework as well as anonymous statistics to evaluate the overall result and to gain a feedback about the quality. The database is stored on the server such that a learner can use another computer having the same configuration as before. Even changing the mode from online to offline or vice versa is possible due to replicating parts of the database from the server to the client.

3.3. Navigation

The VLE has to support common navigation, e.g., back and forward with different step sizes, back to the different contents (main contents, list of figures, tables, or pictures), or links. For a learner further navigation is necessary to find the required information without an extensive search process and to gain an overview of the interdependencies between the learning units. As shown in Section 3.1, SmartBars can be used to integrate further information like glossary or bibliography within the LU without affecting clarity of the textual presentation. The standard to visualize information like content or indices is a hierarchically ordered tree as it is commonly used within file browsers. We want to introduce the use of hyperbolic trees (see also Inxight (2001) for a hyperbolic tree representation similar to Figure 4, Touchgraph (2001) and Wissen.de (2001) for graph based maps) and colors to support the navigation, the representation of an overview as well as the history of the learning process.

Figure 4 shows an example of the interactive way to display linked information. Each node represents a LU in the VLE, the color assigns the node to a specific topic, the solid arcs between the nodes are direct references, whereas the indirect references are shown as dotted and lighter arcs. The kind of shown references as well as the depth – displaying a certain number of nodes or nodes within a specific subject – can be varied to receive the required information without causing an information proliferation. The learner can drag nodes to change the focus of the shown area and, therefore, can traverse the LU by their connection. Furthermore, the learner can analyze the path chosen during the learning session by following the thick arcs. On the right-hand side the history is given as a colorized first-in-first-out-list with a new colorized element for each LU the learner visits. Using the slider, the learner can go through the history and select the requested LU. The colors illustrate the topic of the LU and allow a faster navigation within the history. The information about the connections is automatically gathered from the content of the database and stored in a separate one.

3.4. Communication

VLE are used for distant learning and, therefore, the integration of communication channels is an important aspect. Besides referring to the authors of text passages by announcing the (E-Mail-) address and Internet-presentation, news and discussion boards can be used to discuss the content. The learner can begin a new board or annotation for every part of the content or on a certain subject. Other users can, if it is not declared as private or for a certain group of learners, access, read, and comment these. Therefore, the learners are supported to start working groups and discussion about the subjects in the VLE. The communication is supported by certain Internet technologies such as whiteboards, online video, Internet telephony, and document transfer. The advantage of a broad offer of communication possibilities does not only help the learners but also the authors due to feedback, error reports, and supplementations. A news channel can be used to announce lectures, events, or exams.

4. Conclusion

In this paper we have shown a brief overview of a concept for a flexible VLE that supports the authors preparing the learning documents by using an XML-foundation for the content as well as the learners by giving them the freedom to choose their personal configuration, communication and navigation possibilities, i.e., SmartBars and hyperbolic trees, independence of the location as well as the choice of the connection mode. The VLE is not yet realized and later evaluations of learner's feedback have to show if the learner requires the flexibility and tools for navigation and communication. However, preliminary interrogation about the prototype of the SmartBars revealed very promising responses. Furthermore, it will be an important aspect to force the development of authoring tools for the content within this project and to establish a refereeing process for the content of the learning material in order to keep a certain standard as it is common for contributions in scientific journals and edited books. The concept is intended to be realized within the VORMS project (see VORMS (2001)) that started in 2001 at six universities in Germany.

Acknowledgements

This work was supported by grants from the BMBF (Bundesministerium für Bildung und Forschung (2001)) in the program "Neue Medien in der Bildung" (08NM094D).

Literatur

Apache Software Foundation (2001a, September). *Cocoon Software*. http://xml.apache.org/cocoon. Date of last check.

Apache Software Foundation (2001b, September). *The Apache Software Foundation*. http://www.apache.org. Date of last check.

Apache Software Foundation (2001c, September). *The Jarkate Site – Jakarta Tomcat*. http://jakarta.apache .org/tomcat/index.html. Date of last check.

ARIADNE Foundation (2001, September). *ARIADNE Project Home Page*. http://ariadne.unil.ch. Date of last check.

Bundesministerium für Bildung und Forschung (2001, September). *Bundesministerium für Bildung und Forschung bmbf*. http://www.bmbf.de. Date of last check.

Björck, U. (2001, September). *List of Virtual Universities.* http://www.ped.gu.se/ulric/vus.html. Date of last check.

Cover, R. (2001, September). *Shareable Courseware Object Reference Model Initiative (SCORM).* http://xml.coverpage.org/scorm.html. Date of last check.

Hazemi, R., S. Hailes, and S. Wilbur (Eds.) (1998). *The Digital University: Reinventing the Academy*. Springer, London.

IEEE Learning Technology Standards Committee (2001, September). *Standard for Information Technology -- Education and Training Systems -- Learning Objects and Metadata.* http://ltsc.ieee.org/wg12. Date of last check.

IMS Global Learning Consortium (2001a, September). *IMS Implementation Handbooks*. http://www.imsproject.org/implementationhandbook/index.html. Date of last check.

IMS Global Learning Consortium (2001b, September). *IMS Question & Test Interoperability Specification*. http://www.imsproject.org/question. Date of last check.

Inxight (2001, September) *Inxight Software*. http://www.inxight.com. Date of last check.

Learning Lab Lower Saxony (2001, September). *Learning Lab Lower Saxony*. http://www.learninglab.de. Date of last check.

Massachusetts Institute of Technology (2001, September). *Educational Uses of the WWW at MIT*. http://web.mit.edu/is/courseweb/courses.html. Date of last check.

Multibook (2001, September). *Multibook.* http://www.multibook.de. Date of last check.

MySQL.org (2001, September). *The Site for the MySQL Community*. http://www.mysql.org. Date of last check.

ORWorld (2001, September). *http://www.or-world.com*. Date of last check.

Touchgraph (2001, September). *Touchgraph Link Browser* http://www.touchgraph.com/browser/LinkBrowser.html. Date of last check.

University of Melbourne (2001, September). *tutOR*. http://www.tutor.ms.unimelb.edu.au. Date of last check.

University of Phoenix (2001, September). *University of Phoenix Online*. http://online.phoenix.edu. Date of last check.

Virtueller Hochschulverbund Karlsruhe (2001, September). *Vikar: Virtueller Hochschulverbund Karlsruhe*. http://www.vikar.de. Date of last check.

VORMS (2001, September). *Virtuelles Studienfach Operations Research/Management Science*. http://www.vorms.org. Date of last check.

W3 Consortium (2001, September). *Mathematical Markup Language (MathML) 2.0 Recommendation*. http://www.w3.org/Math. Date of last check.

Walsh, N. and L. Muellner (1999). *DocBook: The Definitive Guide.* O'Reilly, Beijing.

Wedemeyer, C.A. (1981). *Learning at the Back Door: Reflections on Non-Traditional Learning in the Lifespan*. University of Wisconsin Press, Madison.

Wissen.de (2001, September). *Wissen.de Matrix* http://www20.wissen.de/xt/default.do?MENUNAME=VisualIndex&MENUID=40,156,538,547&TOPICID=1017970&RESOURCEID=1017970. Date of last check.

Mining the Data From Experiments on Algorithms using Maximum Likelihood Clustering

David L. Woodruff[1], Ramanpreet Singh[1] and Torsten Reiners[1,2]

[1] University of California at Davis
Davis CA 95616 USA
[2]Technische Universität Braunschweig
Braunschweig, Germany

Contact:
Prof. David L. Woodruff
GSM
U.C. Davis
Davis CA 95616 USA
dlwoodruff@ucdavis.edu

Mining the Data From Experiments on Algorithms using Maximum Likelihood Clustering

Abstract

Data mining represents an exciting information technology frontier. This paper connects statistics, computer science, and operations research using information technology as the catalyst. We examine the use of heuristic search as a basis for data mining algorithms and we demonstrate that data mining can be used to improve heuristic search algorithm performance. In summation, we describe an important problem, provide a neighborhood structure for the problem, and demonstrate its value for heuristic algorithm development.

1 Introduction

Data mining represents an exciting information technology frontier with numerous opportunities to exploit data in new ways (see e.g., [3, 15]). One of our goals is to examine the use of heuristic search as a basis for data mining algorithms and we demonstrate that data mining can be used to improve heuristic search algorithm performance.

The data mining community often refers to the activities of interest to us as *data segmentation.* The statistics community refers to them as cluster finding (with no *a priori* metric). The connections between data mining and statistics are fairly clear. For example, Glymour et al. [11] refer to data mining as being "on the interface of computer science and statistics." One of our goals is to highlight potential contributions of and to operations research. Statistical methods of finding clusters in data are described and applied to data generated by the parameters and performance of algorithms applied to a hard problem: the problem of finding maximum likelihood clusters in data.

The analysis of algorithms applied to hard problems gives rise to a new set of hard problems. How can the results of experiments on algorithms be exploited? In this paper we describe some statistical methods of exploratory algorithm performance analysis.

Often, one has a hypothesis concerning algorithm performance and experiments are conducted to test it. Certainly, this is an important activity. However, in this paper we are interested in using algorithm performance data to see if new relationships can be discovered rather than in testing hypothesized relationships.

The self-referential nature of our work requires some modifications to standard notation. A simple example is that the optimization literature typically uses x as a variable while the statistics literature typically employs x to represent data; since we span the fields, we eschew the use of x altogether to avoid confusion.

We define the generic hard problem to which algorithms are applied as

$$\begin{array}{rl} \min_\tau & f(\tau) \quad \text{(P)} \\ \text{Subject to:} & \tau \in \Xi \end{array}$$

where the set Ξ is intended to summarize the constraints placed on the decision vector τ. We refer to all data for the problem – the data that specifies the *objective function* $f(\cdot)$ and Ξ – as (P). In some cases it may be convenient to use a vector P that implies, rather than specifies, the problem instance. For the problem instances of interest to us here, there are no known algorithms that can find provably optimal solutions in a reasonable amount of time, so heuristic algorithms are employed that find reasonably good solutions fairly quickly.

We will refer to the heuristic algorithm parameters as ρ. We denote by $h(\rho; P)$ the vector giving the results of applying a heuristic algorithm parameterized by ρ to the problem instance specified by (P). The results are a vector because at the very least they will include the best value of $f(\cdot)$ found and some measure of the time or effort required to find it and in many situations we will be interested in additional statistics.

During experimentation, we execute the algorithm and the result is a tuple

$$(P, \rho, h(\rho; P)).$$

It will be convenient to refer to one such 'record' as a vector z and the (arbitrarily ordered) collection of records from multiple runs as Z. We might also refer to a set of records, Z, as a dataset (or a *sample* or a *population* depending on the context). To be consistent with statistics literature on clustering, we will refer to the dimension of each vector as p and the number of vectors as n.

In this paper, we examine ways of looking for naturally occuring clusters in a dataset, Z, of points in $\Re^p$. We describe a version of this type of estimator that explicitly allows for outliers. This methodology is applied to vectors generated by experiments on algorithms. In the next section we develop the methodology while simultaneously providing connections with other research literature. In §3, we report on application to data generated by a Simulated Annealing (SA) algorithm as parameterized by Johnson et al. applied to the problem of finding clusters. Since the performance of this algorithm with respect to its parameters has been studied extensively it provides a useful demonstration of the methods. This self-referential data mining is continued when we do the same thing using Reactive Tabu Search (RTS). The final section offers conclusions and directions for further research.

2 Clustering with no *a priori* Metric

2.1 Estimators

We are presented with a collection of points in $\Re^p$ (i.e., vectors) and we ask if there are naturally occuring clusters. We begin by assuming that we are looking for some number of clusters g, which can be varied by the analyst (or 'data miner'). But we do not assume that we know *a priori* an appropriate distance metric. This is in sharp contrast to the large literature concerning methods that assume that a distance metric is known or that the Euclidean metric is valid (a small sample includes [22, 16, 8, 5]).

Since the assumption of a Euclidean metric is so popular and gives rise to powerful algorithms, we should pause to reflect on why it is generally not valid for our application. The reason is that many of our measurement scales are completely arbitrary, so we want a clustering method that is invariant to affine transformations of the data. This cannot be achieved by simply standardizing the measurement scales because the Euclidean metric ignores the covariance structures during the estimation process.

There is a smaller, but growing literature devoted to clustering when no metric is assumed, but instead a statistical model is assumed. In this model the data are generated in two stages: 1) a cluster distribution is selected from among g possibilities and 2) an observation is drawn from the selected cluster. The notation π_i is used to denote the probability of selecting cluster i. Cluster i is assumed to be modeled by a multivariate normal distribution with density f and mean and covariance $\theta_i = (\mu_i, \Sigma_i)$. This results in the mixture likelihood

$$L(Z;\theta) = \prod_{j=1}^{n} \left[\sum_{i=1}^{g} \pi_i f(\mathbf{z}_j, \theta_i) \right] \tag{1}$$

where $\theta = (\theta_1, ..., \theta_g, \pi_1, ..., \pi_g)$.

A thorough examination of criteria based on the likelihood is given by Banfield and Raftery [1]. Their paper proposes a number of criteria that maximize the likelihood conditional on a clustering, under a number of assumptions about the relative sizes and shapes of the clusters. A popular method is to solve problem (MINW) [7], which finds the clustering that minimizes the determinant of the pooled covariance $|W|$ where

$$W = \sum_{i=1}^{g} W_i,$$

$$W_i = \sum_{j=1}^{n_i} (z_{ij} - \bar{z}_i)(z_{ij} - \bar{z}_i)^T,$$

z_{ij} is the j^{th} vector assigned to the i^{th} cluster, $\bar{z}_i$ is the mean of the vectors assigned to the i^{th} cluster and n_i is the number of points assigned to cluster i.

This corresponds to maximum likelihood under the assumption that the vectors z are multivariate normal with homogeneous but otherwise unrestricted cluster covariances. Algorithms proposed for this minimization include hierarchical agglomeration [22, 17] and local search [21]. We caste this minimization into the framework given in the introduction by giving the vector τ two indexes. We let τ_{ij} be one if the j^{th} vector in the population or sample has been assigned to cluster i and zero otherwise. The objective function is then $|W|$.

We have no particular reason to expect homogenous cluster covariance structures, so we make use of an objective that is similar from a computational standpoint, which is

$$\sum_{i=1}^{g} n_i \log \left| \frac{W_i}{n_i} \right| .$$

The minimum corresponds to a maximum likelihood under the assumption of heterogenous covariance matrices. It was first given by Scott and Symons [20] and adjusted by Banfield and Raftery [1]. Call the problem with this objective function (MIND).

For either objective function, we need to constrain the τ_{ij} to be either zero or one. To ensure that every point is in some cluster we also require that $\sum_{i=1}^{g} \tau_{ij} = 1$ for all $j = 1, \ldots, n$. Refer to these as the *placement* constraints. In addition, we need to constrain each cluster to contain some minimum number of points, H. Clearly, $H > p$ or else the determinant will be undefined. This can be restated as $\sum_{j=1}^{n} \tau_{ij} \geq H$ for all $i = 1, \ldots, g$. We refer to these as *size* constraints.

When used in a data mining application such as the one that we have proposed, it is extremely unrealistic to assume that all data comes from a population that has contributed enough points to form a cluster that can be identified. Banfield and Raftery [1] have proposed maximum likelihood clustering with a cluster numbered $g+1$ that is assumed to come from a multi-variate Poisson. However, we have no reason to expect a Poisson process to contribute, so we need a more general model. Hence we make use of a pure *outlier* model based on [25].

We notice that MINW is a multiple cluster generalization of the minimum covariance determinant (MCD) estimator of Rousseeuw [18, 12, 19, 23] for location (mean) and shape (covariance) of one cluster in the presence of outliers. This causes us to add a cluster numbered $g+1$ to protect against up to some number of outliers T. Starting with problem (MIND) we change only the placement constraint so that the summation is to $g+1$ rather than g and add a constraint $\sum_{j=1}^{n} \tau_{ij} \leq T$ for $i = g+1$. We refer to this as the *outlier* constraint. In particular, we retain the objective function that sums over the first g groups. The effect is that the points labeled as *outliers* (i.e., assigned to cluster $g+1$) "don't count." It is easy to prove that the $g+1$ group will always have T members for optimal solutions so the outlier constraint can be rewritten to be an equality constraint. This has some computational advantages.

Refer to the (MIND) problem modified to allow for T outliers as problem

(MINO). So to summarize: the data for problem (MINO) consists of the tuple (p, n, Z, g, H, T) and a solution to an instance of the problem is implied by giving a vector τ that satisfies the constraints and minimizes the objective function. The apparent redundancy between p, n and Z is useful for visual inspection. A full mathematical programming formulation of (MINO) is given in the Appendix.

2.2 Neighborhood Structures

There have been no algorithms reported in the literature for (MINW) or (MIND) that can find a provably optimal solution without enumerating every possible solution. These problems are of the sort that must be addressed using heuristics that find good solutions in a reasonable amount of time. For computational reasons, neighborhood based search algorithms are attractive for the problems in the (MINW) family.

Neighborhoods are based on *moves* from one solution to another. All of the solutions that can be reached from a given solution in one move are said to be in the neighborhood of the solution. We use the notation $\mathcal{N}(\tau)$ to indicate the set of solutions that are neighbors of a solution τ.

For (MINW) and (MIND) an obvious neighborhood is one where a point is moved from one group to another. For solutions where the size constraints are not binding, the neighborhood has $(g-1)n$ solutions. There are fewer neighbors of solutions for which one or more of the size constraints are binding. This neighborhood is used by Späth [21] as well as Coleman and Woodruff [4] in first-improving local search algorithms.

The neighborhood structures are more complicated for (MINO) due to the presence of the outlier group. Moves involving the outlier group always involve two points: one coming into the outlier group and one leaving. So when none of the size constraints are binding, the neighborhood has $(n-T)(g-1)+(n-T)T$ solutions. For a solution that causes s size constraints to bind, the neighborhood size is $(n-T-sH)+(n-T)T$. We cannot use first improving neighborhoods that have been developed for the older (MINW) and (MIND) because after a move involving the outlier group, continuation through the neighborhood is undefined. We now describe in detail the computations needed to support local search for the estimator defined by (MINO).

Once the objective function has been computed for a solution, one can anticipate the effect on the objective function of all moves with far less computational effort than computing the objective function given an entirely new solution vector. Update formulas for the covariance determinant are used as described in Hawkins [13]. Also, for steepest descent algorithms, the best swaps involving outliers can be precomputed.

The feasible moves in a neighborhood can be classified into two types.

1. Move not involving outlier group: This move investigates the possibility

that the j^{th} data vector z_j has been misclassified to cluster i instead of i', $i' \neq i$ and $i' = 1, 2, ...g$. That is move z_j from cluster i to a different cluster i', provided such a move satisfies the size constraints. This involves setting $\tau_{ij} = 0$ and $\tau_{i'j} = 1$ and decrement n_i by 1 and increment $n_{i'}$ by 1.

2. Move involving outlier group: This move investigates the possibility that the j^{th} data vector z_j, currently assigned to cluster i, should be an outlier and a data vector z_k, currently assigned to the outlier group ($\tau_{ik} = 0, \ \forall i$), should be a member of the cluster i'. Here there are two possibilities:
   (a) $i' = i$. That is move z_j from cluster i to the outlier group and move a member of the outlier group z_k back to cluster i. This involves setting $\tau_{ij} = 0$ and $\tau_{ik} = 1$.
   (b) $i' \neq i$. That is move z_j from cluster i to the outlier group and move a member of the outlier group z_k to a different cluster i'. This involves setting $\tau_{ij} = 0$ and $\tau_{i'k} = 1$.

Implementation of local search heuristics can be broken broadly into two parts, first computing the objective function for the current solution and second evaluating all feasible moves in the neighborhood of the current solution and making the best move if there exist improving moves.

2.2.1 Evaluation of the Objective Function

We introduced

$$\sum_{i=1}^{g} n_i \log \det(\frac{W_i}{n_i})$$

as the objective function to be evaluated, but it turns out to be useful to compute this by augmenting the data vectors z_k with a 1 (to adjust the matrix W_i for the mean) defining $y_k = (1; {z_k}^T)^T, k = j_1, \ldots, j_{n_i}$ and writing the partitioned matrices

$$Y_i = (y_{j_1}, y_{j_2}, \ldots, y_{j_{n_i}})$$

and

$$Z_i = (z_{j_1}, z_{j_2}, \ldots, z_{j_{n_i}})$$

where we let $J_i = (j_1, j_2, \ldots, j_{n_i})$ be the set of indices of the n_i elements assigned to cluster i in the current solution (see, e.g., [13]). Then,

$$Y_i Y_i^T = \begin{pmatrix} n_i & n_i \overline{z_i}^T \\ n_i \overline{z_i} & Z_i Z_i^T \end{pmatrix}$$

Now, defining $C_i = Y_i Y_i^T$,

$$\begin{aligned} \det(C_i) &= n_i \det(Z_i Z_i^T - n_i \overline{z}_i \overline{z}_i^T) \\ &= n_i \det(n_i \textstyle\sum_{k=j_1}^{j_{n_i}} (z_k - \overline{z_i})(z_k - \overline{z_i})^T) \\ &= n_i \det(W_i) \\ &= n_i^{p+1} \det(\frac{W_i}{n_i}) \end{aligned}$$

Hence, the objective function is equal to

$$\sum_{i=1}^{g} n_i \log(\frac{\det(C_i)}{n_i^{p+1}}) \tag{2}$$

Now, since

$$C_i = \sum_{k=j_1}^{j_{n_i}} y_k {y_k}^T \tag{3}$$

this gives an opportunity for efficiently computing the covariance matrix C_i and hence the objective function. This is done by taking each data vector z_j (of dimension p), creating a $p+1$ dimensional vector $y_j = (1 : z_j^T)^T$, and calculating a $(p+1) \times (p+1)$ dimensional matrix $y_j y_j^T$. The covariance matrix C_i is then constructed by incrementally adding $y_j y_j^T$ for all data points of the i^{th} cluster. Once the matrices C_i are constructed, their determinants are calculated. Now knowing the membership n_i of each cluster and the dimension p of the data, calculation of the objective function using equation (2) is straightforward.

2.2.2 Evaluating Objective Function for a Move and Making the Best Move

Every move involves adding and/or removing a data point from a cluster. Evaluating the changed objective function for such a move involves updating/downdating the determinants of the covariance matrices of the affected clusters. Equation (3) can be used directly for this purpose, however this requires the determinants of the matrices to be recomputed for each move being considered. Since calculation of the determinants is an expensive operation the following update formulas (see, e.g., [21]) are used:

$$det(C_i + \beta y_k {y_k}^T) = det(C_i)(1 + \beta {y_k}^T {C_i}^{-1} y_k)$$

where $\beta = +1$ when y_k is being added, and $\beta = -1$ when y_k is being removed from the cluster i.

For the case of a swap (case 2a above) where an element y_j is removed from cluster i and a different element y_k is added to the same cluster, the update formula is:

$$det(C_i + y_k {y_k}^T - y_j {y_j}^T) = det(C_i)[(1 - {y_j}^T {C_i}^{-1} y_j)(1 + {y_k}^T {C_i}^{-1} y_k) + ({y_j}^T {C_i}^{-1} y_k)^2]$$

These formulas provide the updated determinants for the affected clusters that are then used to calculate the objective function for the move. The objective function is evaluated for each feasible move in the neighborhood, and the best move determined. Making the selected move involves setting the τ_{ij} appropriately. In addition, the inverses of the covariance matrices need to be maintained and updated for the evaluation of the next move. The following standard formulas are used to update or downdate these inverses.

$$(C_i + \beta y_k {y_k}^T)^{-1} = {C_i}^{-1} - \frac{\beta {C_i}^{-1} y_k {y_k}^T {C_i}^{-1}}{1 + \beta {y_k}^T {C_i}^{-1} y_k}$$

This inverse of $(C_i + \beta y_k {y_k}^T)$ exists whenever $1 + \beta {y_k}^T {C_i}^{-1} y_k \neq 0$.

It is well known that computational errors accumulate when using these formulas repeatedly, particularly the downdating formula. Hence the algorithms recompute the covariance structures after every 10 moves.

2.2.3 Precomputation of a Change in the Objective Function for Moves Involving Outliers

In moves that involve outliers, we need to determine the best outlier to move back into one of the clusters and the cluster to which to move it. For this we need to evaluate the change in objective function for each of the T outliers moving to each of the g clusters, and then determine the outlier and cluster corresponding to the minimum change. This involves making Tg computations of determinants. Since for a given solution this computation remains the same for all moves involving outliers, this computation is done only once. An exception to this is when the outlier is to move to the group from which a data point is moving to the outlier group. In this case there is some small inaccuracy which could result in a sub-optimal choice of outlier to move, but for the sake of efficiency this is acceptable (with data of any significant size the chance of this would be negligible.) When a move involving outliers is made, recomputation of the determinants is done only for the affected clusters and the new outlier. When a move that does not involve outliers is made, recomputation is done only for the two affected clusters.

3 Application of and to Johnson et al. SA

Armed with a neighborhood structure we can employ local search heuristics. To illustrate the use of the clustering methods, we first employ and study a version of simulated annealing, which is a general purpose heuristic defined relative to a neighborhood structure and a *cooling schedule.* At each iteration k, a solution, τ' is selected at random from the neighborhood of the incumbent solution, $\tau^{(k)}$. Let Δ be $f(\tau') - f(\tau^{(k)})$. The move is said to be *accepted* and $\tau^{(k+1)} \leftarrow \tau'$ if $\Delta < 0$ or if

$$e^{-\Delta/c} > R$$

where c is the "temperature" that is controlled by the cooling schedule and R is a pseudo random deviate drawn from a distribution that is uniform on (0,1). Otherwise, the move is said to be *rejected* and $\tau^{(k+1)} \leftarrow \tau^{(k)}$. The solution vector

τ that satisfies the constraints and for which $f(\tau)$ is the lowest seen to that point in the search is retained as the "current champion." For our applications, the initial solution $\tau^{(0)}$ is selected at random.

The cooling schedule developed and parameterized by Johnson et al. [14] is popular and has a number of properties that make it worthy of study using clustering algorithms. The basic idea of the cooling schedule is a geometric progression of temperatures, with multiple random trials at each temperature. After some number, L, of moves have been generated and tested the temperature is reduced.

The Johnson et al. algorithm has four parameters: ρ =(TF, IP, MP, SF) with the following meaning as outlined in their paper:

TF TEMPFACT: When the temperature is to be reduced, the current temperature is multiplied by TF to give the new temperature.

IP INITPROB: Based on an abbreviated trial annealing run, a temperature is found at which the fraction of accepted moves is approximately IP, and this is used as the starting temperature.

MP MINPERCENT: This is used to test whether the annealing process is *frozen* and should be terminated. A counter is incremented by 1 each time a temperature is completed for which the percentage of accepted moves is MP or lower. The counter is reset every time a new champion is found. If the counter ever reaches 5, the algorithm is terminated. The value of 5 is actually a fifth parameter in disguise with a fixed value. Due to strong interactions with MP, leaving this parameter fixed at 5 seems reasonable.

SF SIZEFACTOR: We set $L = \lambda$*SF, where λ is approximately the average neighborhood size (e.g., $\lambda = (n - T)(g - 1) + (n - T)T$)

The parameter values recommended by Johnson et al. are (0.95, 0.4, 2, 16). Extensive experiments across a broad range of problems indicates that the CPU time and best objective function value seen vary in fairly consistent ways as a function of parameters. This consistent performance makes us suspect that the algorithm represents a good opportunity to illustrate our data mining methods.

3.1 Experiments

Our experiments are done with a fixed data set and fixed clustering parameters, so that only the four SA algorithm parameters vary. The data set that we make use of is the famous Fischer Iris Data (FID) due to Fischer [6] which has 150 points with four values each. We know that it has three clusters, but for these simulations we pretend that we don't know what they are. The problem instance is characterized by the vector ($p = 4$, $n = 150$, FID, $g = 3$, $H = 35$, $T = 2$)

We execute the Johnson et al. SA algorithm with randomly selected parameter vectors and observe the CPU time in seconds, CPU, and the best objective function value seen for MINO, f^*. I.e., $h(\cdot) = (\text{CPU}, f^*)$. The pseudo random number stream is also varied for each run. The SA parameter values are chosen as follows: IP uniform on [0.1, 0.7], TF uniform over a log scale on [0.9, 0.99], SF uniform over a log scale on [8, 128], and MP uniform on [3, 8]. The experiment consisted of 200 runs. Once the runs of SA on the Fischer Iris Data are complete there is a new data set that consists of the four SA parameters and the results given as CPU and f^*. Refer to this data set as SAI, which is characterized by $p = 6, n = 200$; each record in the data set gives an instance of (TF, IP, MP, SF, CPU, f^*) .

The data set has a mean of (0.948, 0.389, 5.475, 40.945, 325.315, -1647.390) with standard deviations (0.026, 0.169, 1.665, 31.390, 306.452, 12.761). The correlation matrix (lower half without the diagonal) is

	TF	IP	MP	SF	CPU
IP	0.00				
MP	-0.12	-0.06			
SF	0.05	-0.01	0.10		
CPU	0.63	0.49	-0.12	0.01	
f^*	-0.16	0.10	0.37	0.11	-0.13

where the *correlation* between i and j is defined as

$$\frac{W_{ij}}{\sqrt{W_{ii}W_{jj}}}$$

Given the large range over which the parameters are varied, the correlations are not very surprising. CPU time has a significant positive correlation with the values of IP and TF (the correlations are 0.63 and 0.49). It is a little bit surprising that time is essentially uncorrelated with quality (-0.13) but this is no doubt due to such a broad set of parameter values. Clustering is in order.

3.2 Data Mining Methods

Data mining is a somewhat tedious and not a very mathematically satisfying activity. The goal of data segmentation is to discover interesting clusters. Such a qualitative objective cannot give rise to a fully quantitative and straightforward process. Rather than proceeding in Gauss-like fashion from assumptions to results, one tries a lot of parameters for the clustering problem and examines the results to see if they might be interesting. As noted in Cabena et al. [3]:

> The analysis of results is inseparable from the data mining step in that the two are typically linked in an interactive process. The exploratory nature of data mining ensures that this is always the case.

During the process of mining the results of SA on the iris data, we vary the minimum cluster size, H, the number of clusters, G, and the number of outliers, T. We use SA to find good clusters in the SA performance data but resist the temptation to recurse again and cluster the data mining runs themselves as part of our study of SA. Dozens of parameter combinations are tried and each time a clustering of the SA results is found by SA, we examine it to see if it is "interesting."

For the purpose of demonstrating the methods we can cheat a little when using SA applied to the iris data. We know what a good clustering of the iris data is because we know in advance which types of flowers generated which data (except, of course, for the chance that some of the flowers were originally misclassified). There seem to be a handful of local minima for MINO applied to the iris data with our neighborhood structures and a wide range of parameters. We can characterize those local minima with less than eight classification errors as "good" results and the rest of the local minima as "bad." It turns out that good classifications have objective function values at and below about -1650. Hence, clusters of SA performance data can be characterized as "good" or "bad" if they happen to have most or all objective function values below or above this value. As a result of intuitively appealing parameterization of SA developed by Johnson et al. and the extensive results reported for this algorithm we also know what sort of correlations should be seen when the algorithm is working well.

3.3 Clusters Found

Even with the ability to "cheat" we must sift through far too many clusterings to report. Perhaps the most interesting clustering was found with the parameters $H = 30$, $g = 2$, and $T = 10$. Since $p = 6$, there are enough points designated as outliers to treat them as a small cluster for the purpose of reporting results. These parameters result in a clustering with 87 points in the first cluster and 103 in the second.

Bear in mind that each data record being clustered in SAI describes a run of SA to cluster the iris data using (TF, IP, MP, SF, CPU, f^*). The clusters of interest are

Cluster 1: Mean: (0.9707, 0.359, 5.2, 41.8, 406, -1649.8)
Standard Deviation: (0.01, 0.16, 1.70, 27.62, 305.22, 12.83)

Correlation Matrix:

	TF	IP	MP	SF	CPU
IP	0.24				
MP	0.02	-0.20			
SF	0.02	-0.04	0.21		
CPU	0.60	0.75	-0.21	-0.04	
f^*	-0.08	0.00	0.48	0.06	0.11

Cluster 2: Mean: (0.9264, 0.408, 5.7, 36.7, 199, -1644.8)
Standard Deviation: (0.01, 0.17, 1.63, 29.33, 105.70, 12.58)

Correlation Matrix:

	TF	IP	MP	SF	CPU
IP	0.10				
MP	0.02	0.03			
SF	0.01	0.02	0.00		
CPU	0.42	0.83	0.07	0.08	
f^*	0.09	0.15	0.29	0.26	0.07

Outliers: Mean: (0.9692, 0.446, 5.9, 76.5, 924, -1652.9)
Standard Deviation: (0.03, 0.16, 1.45, 55.99, 644.75, 7.78)

Correlation Matrix:

	TF	IP	MP	SF	CPU
IP	0.41				
MP	-0.49	-0.47			
SF	-0.71	-0.21	0.25		
CPU	0.95	0.52	-0.31	-0.76	
f^*	0.38	0.28	-0.17	0.22	0.25

The points designated as "outliers" are the most interesting. This designation is a little bit strange. They are outliers in the sense that they are not in either of the groups in the clustering with the best objective function value. Note that the mean values of the four SA parameters are near the recommended values for Johnson et al. and that the correlations with time and objective function value are mostly pro-intuitive and almost all significant to at least 0.9 (if the correlation is $r = 0$ and the number of points in the cluster is N, the variate $r\sqrt{(N-2)/(1-r^2)}$ has a Student-Fisher t distribution with $N - 2$ degrees of freedom; this can be used to test the hypothesis that $r = 0$). Based on the work of Johnson et al. and other experience with SA, these correlations represent exactly the sort of behavior we would expect to see when the algorithm is working properly. In fact, all of the points have low objective function values.

The first cluster also contains some points with low objective function values, but only a few. Many of the correlations are what we would expect from good performance, but many others are insignificant. For the second cluster, most of the correlations that we would expect to find are insignificant (e.g., the correlation between SF and CPU is nearly zero). It is no surprise, then, that this cluster is composed almost entirely of points that correspond to bad clusterings of the iris data.

For the outlier group shown above, the correlations are quite similar to those predicted by experience with successful annealing and furthermore, the mean parameter values for this small "cluster" happen to be very near the values recommended by Johnson et al. Regardless of the values of g and H that were used, we were never able to find large clusters with good objective function values and correlations that conform with the expectations of successful simulated annealing. This application of data mining causes us to believe that SA may not be the best algorithm for this problem so we now explore the use of RTS.

4 Application of and to Reactive Tabu Search

Reactive Tabu Search (Battiti and Tecchiolli [2]) is a variant of Tabu Search [9, 10]. The particular implementation of RTS that we describe here is the one in the Woodruff [24] class library and differs slightly from the version described by Battiti and Tecchiolli.

A tabu search proceeds from trial solution τ^k to a trial solution

$$\tau^{k+1} \in \mathcal{A} \subset \mathcal{N}(\tau^k)$$

where $\mathcal{A}$ is the set of admissible moves (for simplicity, we show it without parameter subscripts). The set $\mathcal{A}$ is formed by removing from $\mathcal{N}(\tau^k)$ any solutions that would reverse attributes of the most recent κ moves and that do not satisfy any aspiration threshold. The moves that are in $\mathcal{N}(\tau)$ but not in $\mathcal{A}$ are said to be tabu. The value of κ is changed dynamically during reactive tabu search (RTS). It is increased if a repeated solution is detected and decreased slowly over time and/or if all moves are tabu. The initial value of κ for our implementation is n/IK, where IK is a parameter. For our experiments, the factors controlling the amount of increase (TI) and decrease (TD) were not interesting (for example, a 10% change works well enough).

The distinguishing feature of RTS is its reliance on long term memory that is concerned primarily with repeated solutions. If too many solutions (given by the parameter RP) are repeated too often (given by the parameter CH) then the search is assumed to be trapped by a particular cup. The metaphor is of a chaotic attractor.

In addition to recording repeated solutions, the number of moves between repeated solutions is also stored (recorded in the variable Running_Average). This value is assumed to be proportional to the size of the domain of attraction of the chaotic attractor. A parameter that sets the maximum cycle length that will influence Running_Average as well as cause changes in the tabu list length is called CM. When RP solutions have been repeated CH times, the search executes a large number of random moves (with the number proportional to Running_Average) in order to try to escape the domain of attraction of the chaotic attractor. In our implementation, we simply create a new, random solution to escape rather than making random moves. In the implementation we employ, the tabu list length is reset to n/IK after an escape.

As a basis of comparison, note that other forms of tabu search likewise make use of frequency-based memory as a foundation for diversification strategies (to drive the search into new regions), but the frequencies are generally recorded for attributes of solutions, such as values of variables and functions, rather than for complete solutions. In addition, the trigger for diversification in other forms of TS typically consists of either an iteration count or a limit on the number of iterations without finding a new best solution.

4.1 Simulated Data

Unless it is restricted to use so little time that it does not escape once or twice, RTS always finds the best known solution for the problems of clustering the iris data. Such short runs require orders of magnitudes less cpu time than SA runs and RTS without escapes is hardly RTS so we conclude that the iris data will not be good test-bed. We turn to simulated data.

Based on arguments given in Coleman and Woodruff [4] suggesting that clusters arranged on a line are the hardest for (MIND) algorithms to find, we create clusters placed along a main coordinate diagonal. Since our methods are affine equivariant, the clusters are generated from a standard multivariate normal with loss of generality. We measure the distance between clusters in terms of the unit of measurement $Q_p = \sqrt{\chi^2_{p;0.001}}$, which is more or less the radius of the sphere around the mean that contains almost all the good points. If the outliers are centered at a distance of $2Q_p$, then these spheres should not overlap. We implement outliers at a distance of dQ_p by adding dQ^*_p to each component, where $Q^*_p = \sqrt{\chi^2_{p;0.001}/p}$. This places the outliers at the correct distance out a diagonal. For the experiments reported here, the first cluster is centered at the origin, the second at a distance of $1.5Q_p$ from the first and the third at a distance of $2.5Q_p$ from the second ($4Q_p$ from the first). The data set has 400 points in dimension 10 with the number of points in each cluster roughly equal. We refer to this simulated data set at SLD.

4.2 Experiments and Clusters Found

The problem instance is characterized by the vector ($p = 10$, $n = 400$, SLD, $g = 3$, $H = 30$, $T = 0$). The algorithm RTS is executed with a different randomly selected starting point for each execution and with parameters selected randomly from the uniform distributions: CH and RP on [1,6], IK on [3,40], TD on [.60, .95], TI on [1.05, 1.35], and CM on [200,600]. For some of the runs CH was correlated with RP and/or TD was negatively correlated with TI. Each execution is terminated after 1200 iterations. The run results have been gathered into a data set for which each record contains (CH, RP, IK, TD, TI, CM, CPU, f^*). Of the 400 runs, 132 resulted in the zero classification errors and the lowest f^* seen.

The data set immediately bifurcates into two large clusters: the 132 with the best f^* value and the other with all other objective function values. We can refer to these as the "good" cluster and the "bad." In classic hierarchical fashion, we can then attempt to find interesting clusters from within these two clusters. Although hundreds of clusterings were examined for the bad cluster, no interesting clusters were found so we turn our attention to the good cluster.

The good cluster is coplanar when the f^* value is included so we remove for further processing. When clustering the resulting data, the interesting clusterings observed were well represented by the result when we looked for two clusters and allowed ten outliers. The interesting interactions were for CH, RP, IK; all other correlations were statistically insignificant. The portions of the correlation matrices of interest were:

1:

	CH	RP
RP	1.00	
IK	0.36	0.36

2:

	CH	RP
RP	-0.18	
IK	0.07	-0.02

This leads to an interesting family of hypothesis that we would not have considered without the benefit of data mining: the three parameters CH, RP, and IK could perhaps be combined into one parameter. This is suggested (but by no means proven) by the fact that for good runs when CH and RP were perfectly correlated, the correlation with IK is also positive. We use the word *family* because the data mining results suggest an infinite variety of exact, well formed statistical hypothesis. This is the goal of data mining: to suggest hypotheses. So in that sense, we suggest that we have demonstrated that data mining can be used in exploratory algorithm analysis.

Although this paper is about data mining rather than hypothesis testing, we did conduct experiments to explore the hypothesis. We found no statistically significant difference between the algorithm with four parameters and the algorithm with six. In fact, the algorithm with four had slightly better average performance. It is beyond the scope of this paper to describe and defend new tabu search algorithms, but we have demonstrated that data mining can be applied to heuristic algorithm performance data and important hypothesis can be discovered.

5 Conclusions and Directions for Further Research

In summation, we have described an important problem, demonstrated its value for heuristic algorithm development and provided a neighborhood structure for the problem. The self-referential nature of our work highlights the directions for further research. Research opportunities exist both in the use of data mining for algorithm development and algorithm development to support data mining.

As we have shown here, data mining tools can be useful during the exploratory phase of algorithm development. Sophisticated algorithms applied to complex problems are hard to analyze. Although some hypothesis are straightforward and suggest themselves to anyone working on the problem, others are surely waiting to be discovered by data mining tools.

In the current excitement over data mining, the fruit is still on low branches. Practitioners are able to extract useful hypothesis from data using straightforward algorithms. Here again, there are surely interesting facts that can be culled from data using more sophisticated algorithms in support of more sophisticated methods such as maximum likelihood clustering. The advantage of model based statistical methods is that probabilistic statements can be made about the results. The perceived drawback is that they often result in difficult optimization problems. This presents an important research opportunity at the interface between Statistics, Computer Science and Operations Research.

Acknowledgements

This work was supported by a grants from the National Science Foundation, the Du Pont Corporation and NIH grant P42 ES04699.

Appendix - Mathematical Programming Formulation of (MINO)

The description of the estimator and the corresponding minimization problem as given in §2.1 makes use of notation common in the statistics literature. The notation also happens to be useful for describing the neighborhood structures. In the interest of rigor and to provide connections with other literature, a mathematical programming version is given here.

We have as data the number of groups, g, a minimum cluster size, H, number of outliers to guard against T, and n points in R^p with each point given as Z_j for $j = 1, \ldots, n$. Our objective is to select $\tau_{ij}, i = 1, \ldots, g+1, j = 1, \ldots, n$ so as to

$$\text{minimize} \sum_{i=1}^{g} n_i \log \left| \frac{W_i}{n_i} \right| \qquad \text{(MINO)}$$

subject to

$$
\begin{array}{lcll}
W_i & = & \sum_{j=1}^{n} (\tau_{ij} Z_j - \bar{z}_i)(\tau_{ij} Z_j - \bar{z}_i)^T & i = 1, \ldots, g \\
\bar{z}_i & = & \left(\sum_{j=1}^{n} \tau_{ij} Z_j \right) / n_i & i = 1, \ldots, g \\
n_i & = & \sum_{j=1}^{n} \tau_{ij} & i = 1, \ldots, g \\
\sum_{i=1}^{g+1} \tau_{ij} & = & 1 & j = 1, \ldots, n \qquad \text{(placement)} \\
\sum_{j=1}^{n} \tau_{ij} & \geq & H & i = 1, \ldots, g \qquad \text{(size)} \\
\sum_{j=1}^{n} \tau_{ij} & \leq & T & i = g+1 \qquad \text{(outlier)} \\
\tau_{ij} & \in & \{0, 1\} & i = 1, \ldots, g+1, \\
 & & & j = 1, \ldots, n.
\end{array}
$$

References

[1] J.D. Banfield and A.E. Raftery. Model-Based Gaussian and Non-Gaussian Clustering, *Biometrics*, 49 (1993) 803–821.

[2] R. Battiti and G. Tecchiolli. The Reactive Tabu Search, *ORSA Journal on Computing*, 6 (1994) 126–140.

[3] P. Cabena, P. Hadjinian, R. Stadler, J. Verhees, and A Zanasi. *Discovering Data Mining.* Prentice Hall, Upper Saddle Rivier, NJ., 1994.

[4] D.A. Coleman and D.L. Woodruff. Cluster Analyse for Large Data Sets: Efficient Algorithms for Maximizing the Mixture Likelihood, 1997, Technical report, GSM, University of California, Davis CA, 95616.

[5] U. Dorndorf and E. Pesch. Fast Clustering Algorithms, *ORSA Journal on Computing*, 6 (1994) 141–153.

[6] R.A. Fischer. The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, 7 (1936) 179–188.

[7] H.P. Friedman and J. Rubin. On some Invariant Criteria for Grouping Data, *Journal of American Statistical Association*, 62 (1967) 1159–1178.

[8] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, 1992.

[9] F. Glover. Tabu Search - Part I, *ORSA Journal on Computing*, 1 (1989) 190–206.

[10] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publisher, Boston, Dordrecht, London, 1997.

[11] C. Glymour, D. Madigan, D. Pregibon, and P. Smyth. Statistical Themes and Lessons for Data Mining, *Data Mining and Knowledge Discovery*, 1 (1997) 11–28.

[12] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistic - The Approach on Influence Functions.* John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1986.

[13] D.M. Hawkins. The Feasible Solution Algorithm for the Minimum Covariance Determinant Estimator in Multivariate Data, *Computational Statistics and Data Analysis*, 17 (1994) 197–210.

[14] D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon. Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning, *Operations Research*, 37 (1989) 865–892.

[15] J. Komorowski and J. Zytkow. *Principles of Data Mining and Knowledge Discovery : First European Symposium, PKDD '97.* Springer, Berlin, 1997.

[16] J.M. Mulvey and H.P. Crowder. Cluster Analysis: An Application of Lagrangian Relaxation, *Management Science*, 25 (1979) 329–340.

[17] F. Murtagh and A.E. Raftery. Fitting Straight Lines to Point Patterns, *Pattern Recognition*, 17 (1984) 479–483.

[18] P.J. Rousseeuw. Multivariate Estimation with High Breakdown Point. In W. Grossmann, G. Pflug, I. Vincze, and W. Werz, editors, *Mathematical Statistics and Applications, Volume B.* Reidel, Dordrecht, 1985.

[19] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, New York, 1987.

[20] A.J. Scott and M.J. Symons. Clustering based on Likelihood Ratio Criteria, *Biometrics*, 27 (1971) 387–397.

[21] H. Späth. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples.* Ellis Horwood Limited, Chichester, 1985.

[22] J.H. Ward. Hierarchial Groupings to Optimize an Objective Function, *Journal of the American Statistical Association*, 58 (1963) 236–244.

[23] D.L. Woodruff. Ghost Image Processing for Minimum Covariance Determinant Estimators, *ORSA Journal on Computing*, 7 (1995) 468–473.

[24] D.L. Woodruff. A Class Library for Heuristic Search, *INFORMS Computer Science Technical Section Newsletter*, pages 1–5.

[25] D.L. Woodruff and D.M. Rocke. Synthesis of Outlier Detection and Multivariate Cluster Finding, 1998, Technical report, GSM, University of California, Davis CA, 95616.

Some Thoughts on how to Educate OR/MS

Torsten Reiners and Stefan Voß

Institute of Information Systems
University of Hamburg
Von Melle-Park 5, D-20146 Hamburg, Germany
e-Mail: reiners@econ.uni-hamburg.de, stefan.voss@uni-hamburg.de

Abstract

E-learning can be seen as the latest technology hype at universities not changing anything or a new methodology to motivate learners. Negative examples of e-learning are HTML-pages of the learning material without any extra value for learners. More advanced approaches support the learner by several extras like interactive exercises but especially in case of quantitative courses, there are further requirements for an adequate presentation. In this paper, we demonstrate the development of SMART-FRAME, the prototype of a virtual learning environment incorporating a hierarchical structured design of the XML-coded learning material including meta-data definitions. Compared to the numerous other virtual learning environments we directly apply the meta-data during the transformation process of the learning material to the desired output format, integrate innovative components for navigation and presentation of information within the learning material. That is, we focus on the learner by using the individual configuration as well as behavior to support and increase the learning experience.

We present two examples how courses in Operations Research/Management Science can be realized. First, a simulation course is sketched including the discrete event-driven simulation tool SIMTOOL for education and the learning method synchronized blended learning. The other course is about meta-heuristics where the course outline includes a presentation of the basics explaining local search and meta-heuristics, the applet Meta-Visu to visualize meta-heuristics step-by-step, and a concept how the theory can be internalized solving real-life problems within an experimental environment.

1 Introduction

Thinking of education most of the lecturers as well as the growing number of students at least in public universities in Europe are confronted with fewer funds for education, crowded classrooms, packed seminars, limited capacity for office hours, and insufficient practical experiences. Several (political) programs try to limit student enrollments even though there are no significant changes so far. Examples for such programs are higher university fees or required grades (numerus clausus) to get into certain programs of study. During the last years, several (research) projects at universities as well as commercial companies were initiated to analyze the potential of using modern information technology to increase the variations of education besides traditional courses. Trying to overcome the known problems and compensate the lack of lifelong learning support, the idea of virtual learning environments (VLEs) as well as virtual universities was born and also realized in numerous approaches using different technologies and strategies (see, e.g., [19]). These programs usually provide only online-access to all course materials (e.g., [26]), but might also be a completely digital or virtual university where the learner is regularly enrolled as a student (e.g., [42]; see also [8] for a list of virtual universities).

The main goal of virtual universities is the provision of education with the same or better quality as in presence universities using mainly VLEs where the learner is navigating—on- or offline—through the electronically reproduced learning material while having access to further mandatory services like communication—with lecturers, teaching assistants, and other students—or participation in (virtual) practical courses and exams.

Even a superficial search in the Internet can reveal that almost every university is participating in projects developing VLEs, mostly together with further partners from educational institutions or companies. Even if the search is concentrated on projects that are developing solutions being available for the appliance in education without further financial investments for the students or lecturer, the number is still enormous; examples at German universities can be found in [25, 43, 44]. Most of these learning environments have an interface to access revised lecture notes in HTML (Hypertext Markup Language) using hyperlinks to improve the usability. Besides simple presentations of course material without any further media pedagogical revision, different virtual collections of learning materials can be found in the Internet using a more sophisticated approach than hyperlinked documents like embedded Java-Applets (e.g., tutOR [41] and the thereon based tutORial [21]). Even though these developments cannot be seen as a VLE due to the lack of essential communication, navigation, and content management systems, they are of great value for learners. The combination of short theoretical introductions to different subjects including an interactive component to visualize the theory by running several examples is well balanced. Therefore, the learner is not demotivated by endless textual explanations but enabled to directly apply the learned material to (small) exercises.

Other systems like OR-World [30] or MultiBook [28] have an advanced foundation due to the usage of XML (eXtensible Markup Language) for the coding of the content as well as certain functionality to be called a VLE. Otherwise, there are still limits in the flexibility for the learners that provide a more or less strictly guided learning experience. Sophisticated—and established on the international market—commercial approaches regarding the support of teaching and managing course material are, e.g., Blackboard [9] and WebCT [50]. In many countries, e.g., the United States of America or Australia, it is common practice to use proprietary software like Microsoft Excel within the classroom instead of web-based learning environments. The lecturer can use the given functionality to easily implement complex visualizations of problem formulations and their solutions including a step-by-step development, whereas the learner already knows the software and, therefore, can focus on the learning material and given exercises; see, e.g., [7, 12]. The disadvantage of the latter approach as well as the commercial products is that they are proprietary software. The designer might be limited by offered functionality and technology and, therefore, might not be able to realize the desired didactical methods to present the material.

In general, the demands on the learning material, its presentation as well as integration within a VLE depends on the subject. In this paper we discuss some of our thoughts and experiences regarding the subject operations research and management science (OR/MS). We have to identify the major didactical methods for OR/MS being required to successfully teach the learning material in an adequate way. For this reason, the subject OR is special with respect to the strict necessity to acquire a multitude of knowledge as well as practical skills in areas like modeling, algorithms, software-engineering, or project management. To allow a self-paced class in OR, we have to ask for certain characteristics as well as cognitive and constructive tools to support the learner within the VLE.

In this paper we focus on simulation as well as meta-heuristics as important OR tools. For these areas we have to define certain (basic) requirements for virtual self-paced courses. In simulation, we can distinguish two major types of simulation tools. First, there are commercial products, e.g., emPlant, which are rather integrating professional components to solve realistic problems than supporting education. Second, research tools, e.g., simjava [24] or JSim [27], integrating new ideas in the field of simulation are not focussing on supporting learners, i.e., explaining the process of creating a model step by step. Furthermore, a simulation tool that extensively explains how to create models mostly lacks certain features. Especially supporting methods that allow to observe the learner and to react accordingly in respect to the learners' input. Therefore, we developed a simulation tool focusing on the didactical components.

Compared to simulation where the learner can find extensive learning material in books and the Internet, meta-heuristics are either not described at all, only given by static hypertext documents, or just a collection of interesting and interactive applets without the textual background explaining the theory. Therefore, the concept for a course has to include a complete overview of, e.g., local search as a basic mechanism of meta-heuristics as well as its application to realistic problems. Besides constructing several interactive components to teach the basics, we work on a concept to integrate a software generator allowing the learner to work with "real" software on "real" problems.

To structure our thoughts the rest of the paper is organized as follows. First, we describe criteria how to start the development of learning material for a virtual course (Section 2.1), the necessity of practical experiences within the course (Section 2.2), the structure of the learning material and presentational design (Section 2.3), innovative components for VLE (Section 2.3.2) as well as the system architecture supporting the preliminary developed concept (Section 2.3.3). Furthermore, we briefly outline our motivation of developing a new VLE instead of reusing existing ones. Section 3 outlines two courses: Simulation and meta-heuristics. We present SimTool, a simulation tool to be used within VLEs and describe how a course unit using SimTool can be realized especially in conjunction with the idea of synchronized blended learning (Section 3.1). Section 3.2 will outline briefly how a course on meta-heuristics can be designed, starting with a basic introduction to local search, followed by applets explaining some meta-heuristics itself, and completed by showing a possible way to support the process of internalizing the learning material. Section 4 concludes the paper and outlines the current status of our research as well as future developments.

2 Technology based on Didactical Thoughts

2.1 Process Model developing Courses in Virtual Learning Environments

The design of virtual courses involves the selection of didactical methods as well as the technology to realize the learning material. Even though the instructional designer should not be restricted in his decision at all, we generally have to distinguish two scenarios:

Technology driven design: Nowadays, it is common that the universities select a (commercial) VLE that has to be used by all faculties on campus. The obvious advantages are the preservation of the corporate identity for (external) courses as well as—and this might be the most important one—the unified form of presentation and access to all courses required to gain a degree in a certain field of study. On the other hand, most VLEs limit the degree of freedom with respect to using didactical methods. Especially due to the restriction that commercial systems usually miss an interface to extend or modify basic components. For example, learning paths are generally implemented as a simple concatenated list of HTML-pages not allowing excursions. Therefore, the usage of advanced concepts like the integration of facultative excursion, adaptation of the learning path according to the configuration or behavior of the learner, or transformation of the learning material according to the technology level of the learner are not possible, e.g., the conversion of SVG-graphics to bitmap graphics in case of missing functionality within the browser.

Didactical driven design: Here, the instructional designer decides on the didactical methods used to present the learning material. Afterwards, the required technologies are selected (or implemented), installed, and used to create a virtual course. Compared to the other approach, the instructional designer itself is not limited to a given set of methods but can combine several technologies to realize a course including the most adequate presentation of the learning material. Therefore, the focus of the course can be set according to the individual learners' needs, e.g., specialized applets or inclusion of interactive examples, instead of forcing the learner to a given learning method. Especially in case of a large variety of subjects, the appropriate learning methods can not be realized using the same technology without reducing the quality. Disadvantages are the required knowledge about learning technologies and their interfaces for an integration process as well as the time consuming development of components, and, for the application within large universities and corporations, the risk of failure in having a corporate identity.

In our research, we apply the second scenario. Based on a concept how the learning material should be presented to the learner, we selected existing (freely available) technologies. Especially the hierarchical structure of the learning material and its dynamical processing based on the configuration of the learner restrict the number of existing VLE such that we start the development of a prototype from scratch allowing us to integrate all desired (innovative) components.

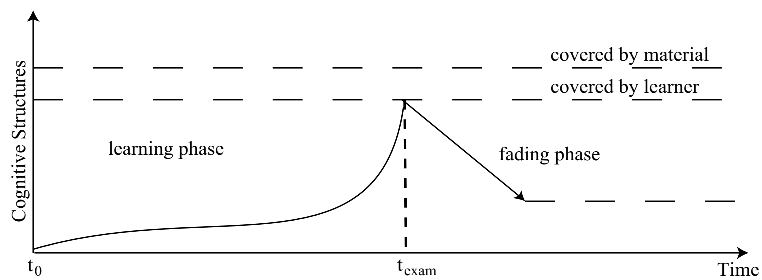


Figure 1: Learning curve as assumed in traditional classrooms

2.2 Relevance of Practical Exercises

One aspect of our research is the evaluation of the learning behavior in the context of the developed learning material as well as increasing the motivation of learners studying OR/MS. Based on personal experiences from either participating and offering lectures on university level within different fields of study over the last years, we declare a hypothesis regarding the learning behavior as well as the characteristic of building cognitive structures; see [1, 11, 40] for a corresponding background.

Hypothesis:

> *The internalization process, i.e., building cognitive structures, will be supported by our concept for a VLE incorporating practical exercises based on real-life scenarios. Furthermore, the fading process subsequent to the exams is reduced.*

Due to the ongoing research within the coming semesters, we have to leave it currently unacknowledged if our concept of improving the learning experience and, therewith, the motivation and building of cognitive structures is sustainable. Furthermore, an evaluation over a period of several semesters have to be applied to prove a general appliance of our learning concept.

In Figure 1 the characteristic of building cognitive structures is shown over the period of a course and the time span subsequent to the course. In the beginning the learning material is unknown to the learner and has to be fully acquired during the course. Generally, the motivation is high during the first sessions but decreases after a certain (short) time period. Therefore, the process of building cognitive structures slows down and learning material from the first sessions already starts to vanish. In addition, the learner does not start to memorize or practically exercise the learning material immediately. Getting closer to the end of the course and, in particular, the exams the learner tries to annex the learning material within a fixed time period—in most countries a term similar to pre-exam-week is well known. This allows to recall most of the learning material during the exam but ignores long period memorization processes. After the exam, and, therewith, the course, the learner is not using the gained knowledge properly in the context of the subject, at least in most cases. Therefore, the phase of forgetting and fading of the newly build cognitive structures immediately starts. Even though the learner will not fall back on the level from before the course the learner might not be able to either apply the learned

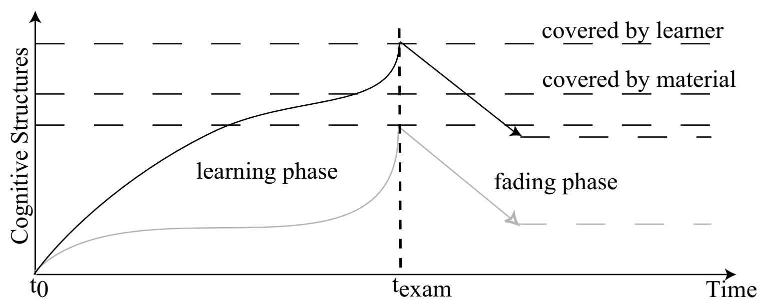


Figure 2: Learning curve as assumed using practical exercises within VLE

material within a new context or recall specific details. For example, the learner might remember that there was a standard deviation on observations but does not know the difference to the variance or even know the equation.

Figure 2 shows the learning curve as we assume it according to our hypothesis using practical exercises during the course. Due to the practical exercise, the motivation of the learner actually working with the learning material is increased resulting in an improved accumulation of cognitive structures. Furthermore, motivated learners work with the learning material, ask questions going beyond the coverage of the course, and, therefore, might learn more than required. Near the exam, the learning curve also raises steeper because the learner has to memorize the learning material that is not part of the practical exercises. After the course, the fading phase occurs as before but here, we assume that the degree is much less than before due to the intensive internalization; see also [40] for a discussion of the added value of practical training compared to theoretical classes.

Our concept incorporates the idea of having a theoretical introduction where the learner can study the basics. During the course the learner is supposed to use interactive components as well as participate in practical exercises like developing simulation models or using meta-heuristics to solve problem instances; see Section 3.

2.3 Components to Improve Virtual Learning Environments

The idea of developing another VLE is questionable due to the numerous projects world-wide. Nevertheless, instead of "recycling" existing environments by customizing their components to our needs—as well as integrating missing parts—we developed a concept for a VLE using freely available software and modern technology.

Based on the experiences from the recent project OR-World [30] as well as analyses of major (and commercial) packages like WebCT [50] we observed several aspects, which do not fulfill our requirements of an adequate support of the learning process. Most products are to some extend a collection of tools that are either developed as stand-alone applications and later integrated into the VLE or implemented to have further features without reaching a level of functionality being even close to alternative choices. This results in non-ergonomic user interfaces and—which is probably the worst part—in unstructured learning material.

Especially WebCT—one of the major software package for virtual education—has to be seen as an ambivalent example due to the design where the courses are handled as a set of HTML-pages managed by the VLE and components like email only support very basic functionality. In addition to interactive presentation of learning material based on HTML, we see the necessity to integrate further didactical components that allow an improvement of the learning experience. In this paper we demonstrate how the pedagogical paradigm synchronized blended learning can be used to teach simulation to learners in a simple way such that the learner's motivation is increased and a high level of internalization is obtained. This might be one of the major problems independent of the kind of learner, e.g., the distant learner, the enrolled learner, or the lifelong learner [51]. Furthermore, our concept incorporates approaches that allow the learner to obtain simple, flexible, and configurable support for an independent—in terms of time and location—learning process without losing the social aspect of interpersonal communication. Especially factors like location independent learning, navigation, communication, information presentation, and configuration are not extensively integrated in most system architectures.

The VLE OVISS (Offenes Virtuelles StudienSystem, see [33] and [32] for the predecessor JEWEL (Java Enabled Worldwide Economic Learning)) is an open-source application in JAVA to interactively deliver the learning material in form of books including training exams by presenting questions in various forms like multiple choice or ordering of components. OVISS is based on XML and supports several international specifications like QTI (Question & Test Interoperability, see [23]) and LMML (Learning Material Markup Language, see [39]). The fact that the integration of further components, the adaptation of the user interface regarding a learner-specific configuration and the handling of learning material—i.e., the inclusion of meta-data of the learning objects—is restricted, influenced our decision to develop a new concept for a VLE. We implemented a basic prototype to demonstrate improvements for the virtual education.

In the sequel we describe the design of the learning material, the system architecture, and components that are currently not commonly used within the educational field but able to improve the quality as well as the learning experience of the learners. The described VLE in its current version has to be seen as a prototype in which the focus of the implementation is rather set on the relevant components to demonstrate our concept than on being a complete environment for education on university level.

2.3.1 Structure of the Learning Material

The learning material is encoded using a hierarchical and modular structure of different learning objects as shown in Figure 3. Depending on the content, size, and relation to other objects, we distinguish four types of learning objects with different levels of granularity and the following meaning. Note, that the learning material is coded in different languages within each learning object.

- **Media element (ME)** is a small unit without further partitioning. Here, it is important that other learning objects are not included but referenced. Examples are components like text fragments (where the size of the fragment should cover a small logical unit), tables, figures, glossary entries, formula systems, or media objects (which can be, e.g., applets, movies, or

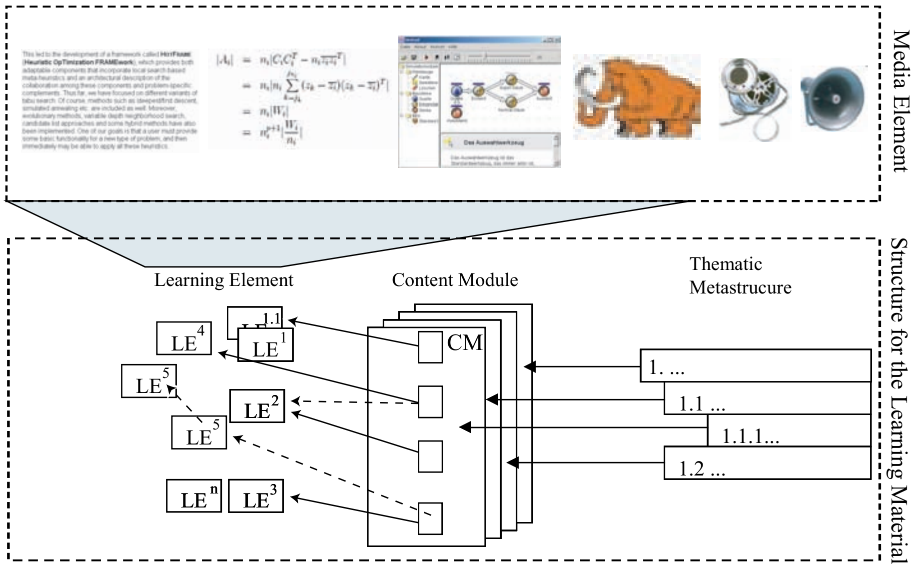


Figure 3: Structure of the hierarchical and modular learning material

pictures).

- **Learning element (LE)** is a composition of objects (media elements as well as learning elements) to produce a semantic unit. One view for "semantic unit" is that the content of the learning element can be displayed on one page where the term page refers to either a web-page including scrolling or a printed section being longer than the actual size of the paper. Note that the content of a learning element does not have to be displayed completely. The author as well as the learner can specify so-called page-breaks to split learning elements into smaller parts for the presentation, which are shown using standard navigation.

- **Content module (CM)** is superior to learning elements grouping these to larger units. The content module should cover a certain content of the lecture or a special field, or include the slides of a course or seminar. The presentation to the learner within a VLE should be in form of a table of contents or hyperbolic tree or graph, respectively.

- **Thematic metastructure (TM)** covers a whole thematic field in the VLE and could either be the work of a department in a specific field or the courses a learner has to take in accordance to get a certain degree.

Authors of learning material may start to construct media elements, which represent the smallest units of "information". Note that especially for the coding of non-sensitive MEs certain rules are specified to increase the possibility for a later reusability of the MEs in different contexts. For example, the same applet can either be used to describe the normal distribution within a statistical course or—changing the context of the applet—within a simulation course to describe the occurrences of certain events. Therefore, the design of media elements should be as general as possible. Specific context sensitive adaptations, e.g., the caption of a figure, can either be done using mechanisms from the VLE or by deriving new learning objects that can be modified by the author. Learning elements as well as content modules and thematic metastructures are build by combining learning objects of the same or lower aggregation level. For a detailed description on designing the learning material in German see [36].

Each learning object is also characterized by a meta-data wrapper. Within different categories the meta-data wrapper keeps information about the content itself, the lifecycle (i.e., the history of changes or its ancestors in case of inheritance from other ME), technical and educational specifications as well as descriptions of rights, relations, classifications, and annotations. Each category contains several elements, e.g., to classify the learning object in respect to the learning level (beginner, intermediate, expert) or type of audience (undergraduate learner, graduate learner, practitioner) by using an appropriate and defined vocabulary and level of detail. The meta-data is used by the authors to gather information about possible reusability of learning objects, to find learning objects to be used within a specific course, e.g., in respect to the difficulty level, or to select corresponding learning objects based on the individual configurations, e.g, for an automatic adaptation of the learning material.

We use the certified specification LOM V6.4 (Learning Object Metadata) by [20]; see also [36] for a critical description. Other specifications for meta-information are, e.g., ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe; see [5]), which is based on LOM, SCORM (Sharable Courseware Object Reference Model; see [10]), and IMS [22]. The meta-data of the learning material can be stored in either a relational (e.g., MySQL [29]) or XML database (e.g., Xindice [2]) incorporating mechanisms to distribute the learning material itself on a local server or other addresses in the Internet.

Furthermore, we implemented a method that assembles the requested learning object using the meta-data information for every subsidiary learning object. The advantage of this approach is a real comprehension of the meta-data where other projects, e.g., projects using LMML, directly link learning objects of the learning material ignoring the meta-data. LMML uses direct links in form of URLs to other learning objects and, therefore, information of the meta-data is not employed. LMML relativizes this by using own meta-data information within the learning material. This is against our beliefs where the content should be separated from the descriptive meta-data.

The learning material itself is coded using XML. XML seems to be the best choice because it possesses the required implementation features, can be easily transformed to other output formats by using XSL-transformations (eXtensible Stylesheet Language, e.g., to reuse learning materials in software from other vendors), and allows the usage of standard operations like full textual search. There are several established specifications to encode different types of content, the following list shows the used (original or slightly adapted) specifications: MathML is a specification for the representation of equations [47], QTI for the representation of questionnaires and multiple choice tests [23], DocBook for bibliographic information and publication structures [49], and especially LMML for classifying the learning objects in items like "exercise", "objective", "algorithm", or "image"; see Figure 4 for a list of items, [39] for a detailed description as well as [34, 36] for a discussion with respect to combining LMML with LOM.

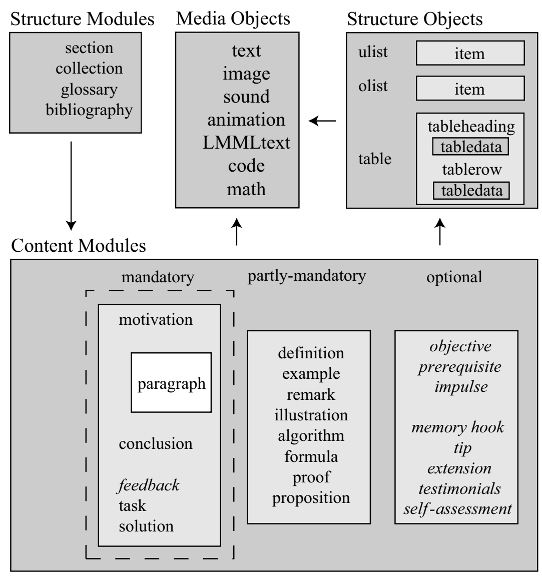


Figure 4: Structure of the Learning Material Markup Language

LMML applies a further semantical classification on the learning material—besides the type of learning object and the meta-data. LMML distinguishes four categories in a hierarchical structure as shown in Figure 4. The following description is based on LMML but includes partly our extensions. Starting with a structure module, the learning material is composed by content modules. Each semantical unit of the learning material—e.g., part of a course—has to motivate the learner (*motivation*), containing the content itself (*paragraph*), summarize the learned material (*conclusion*), and should present a feedback to the learner by, e.g., asking questions or presenting exercises including the solutions (*feedback, task, solution*). Besides these mandatory modules the learning unit can be enriched by incorporating further content modules like semi-mandatory modules—due to the fact that theses modules are part of the LMML-specification and also are the most important kinds of modules to build a learning unit—or optional modules—mainly to increase the learning experience and the motivation of the learner. Another aspect of modularizing the learning material is its presentation. The learner has the capability of defining the look of each module for recognizing the module during the learning process as well as configuring, which types of modules should be included at all. For example, a beginner might need a motivation whereas the expert mostly requires a summary of the learning material in accordance to receive an overview of the important facts of the content.

2.3.2 Innovative Components within Virtual Learning Environments

Developing a VLE implies innovative components due to a demarcation to other products as well as legitimation of using (human) resources for these implementations. Within our VLE several components can be seen as innovative because

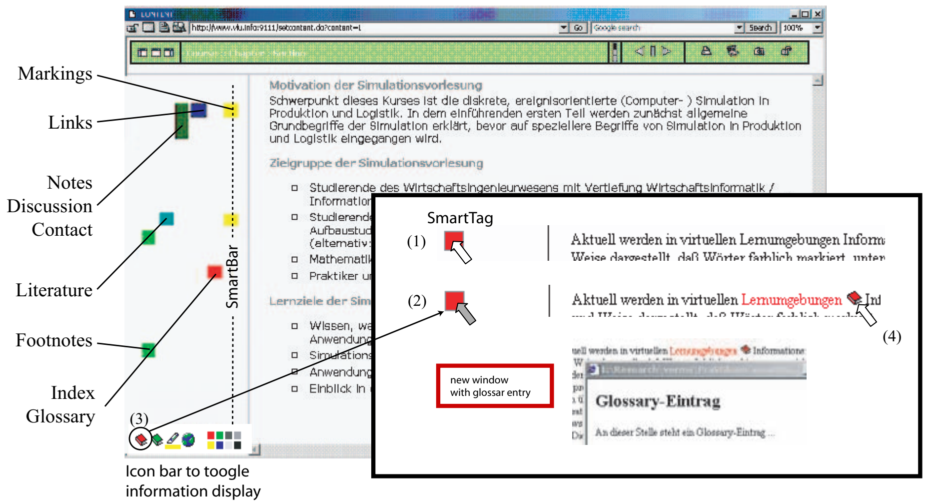


Figure 5: Screenshot and functionality of SmartBars

of either applying existing technologies within VLEs, suggesting new didactical methods or forms of presentation, or designing a modular interface based on modern technologies and software products. In particular, we introduce synchronized blended learning (see Section 3.1.3), SmartBars to prevent information inundation by aggregating and visualizing certain information within the learning material, and the integrated concept of using the meta-data for the presentation of the learning material.

Nowadays, the presentation of learning material is mixed with information to support the learner organizing and memorizing the structure of the content for longer periods. Even so the amount of information should be limited, some learning material includes several symbols, colors, and styles to represent links, references to literature or glossary entries, footnotes, or accentuation of important parts. By overusing the presentation of information, the readability and, therewith, the motivation can extensively be reduced. Therefore, we developed the concept of SmartBars as shown in Figure 5. On the side of the learning material, SmartBars for the different information are displayed. This means that information-specific colored SmartTags (small boxes) are drawn next to the text line with the corresponding information. The learner can now decide how the information should be visualized:

- **no information** means that the SmartBars are turned off and no information is shown in the learning material
- **hidden information** means that the information within the learning material is shown whenever the learner is moving the mouse cursor over a SmartTag
- **shown information** means that the information within the learning material is displayed according to their type by either clicking with the mouse on a SmartTag or the associated icon. For example, a glossary entry is represented by a small book within the learning material after being activated by the learner. The activation process is shown in the framed box

in Figure 5: The learner can move the mouse over the SmartTag (1) for a temporary display, click the SmartTag (2) for a constant display of the symbol for a glossary entry, or turning all symbols on or off by using the icon for glossary entries (3). Clicking on this symbol (3) opens a new window showing the glossary entry.

Further functionality of the SmartBars are the learner-specific insertion of textual markings, discussion forums as well as notes. These inserted items are correlated with the learning material allowing the learner to actually operate with the text similar to printed material where text marker and margin notes are common tools to support the learning process. In case that every annotation by the learner is associated with a time stamp, the learning process gets more comprehensible in a way that the recognition of the learning progress and, indirectly, the internalization is supported.

Two other components distinguish our research from other projects: The online inclusion of the meta-data while generating the output for the learner based on their individual configurations and the design of the system architecture, i.e., the modularly designed VLE; see the following section for a detailed description.

2.3.3 System Architecture—SmartFrame

Based on [15, 16] identifying the requirements for virtual learning, we developed a concept for a user interface to access the learning material based on individual configuration and desires of the learners for an improved learning experiment. Due to the evaluation the design had to incorporate as much flexibility as possible in terms of changing the presentation of the interface, the learning material, and, therewith, the learning experience itself. Furthermore, the interface should be independent from the specific representation of the learning material as described in Section 2.3.1.

SmartFrame (Smart Technology for Research and Modern Education) is developed in respect to the demanded features. Using an Apache-Tomcat Server the user interface is dynamically generated by selecting, combining, and executing Java Server Pages (JSP) with methods to, e.g., access the database or process user input. The coordination of this process is done by special servlets configured by XML-files, which can be user-specifically changed according to the individual preferences; see also the lower right-hand side of Figure 6 for a schematic overview. Furthermore, the browser-based interface is detached from the learning material by exclusively referencing the learning objects by their identifier. This allows an integration as well as blending of XML-based learning material and external learning materials, e.g., HTML-pages, without adapting the logic of SmartFrame. This is especially relevant for the design and administration of courses and learning paths. Regarding the XML-based learning material as described in the previous section, certain information has to be passed from the interface to the transformation process from XML to the output format. This is done using a common session object storing user specific information about the setting.

The concept of SmartFrame comprises (standard) components regarding the navigation, communication, and administration as described in [35]: Navigation includes modern technologies like hypergraphs, detailed visualization of the

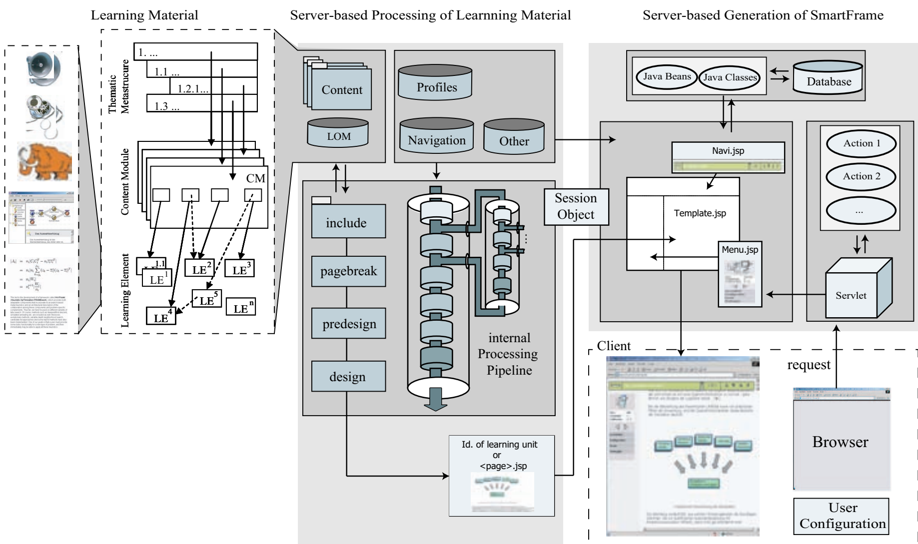


Figure 6: System architecture of SmartFrame

learning process including a time-based history, extensive processing of learning paths like intersecting, combining, and redefining. Communication includes discussion boards, (group-oriented) chat-rooms, email-systems, white boards, and instant messaging; administration includes management of learners and courses, analysis of statistics, and authoring tools for the learning material and learning paths. Special design interest is put into the usability. Therefore, a hierarchical structure accessing the components is used whereas the focus is set on intuitive, ergonomic, and fast accessibility.

2.3.4 System Architecture—Processing the Learning Material

Another key technology of our concept is the usage of XML-based learning material, the specification of the learning objects by meta-data, the inclusion of the meta-data within the transformation process from XML to a user-specified format, and the implementation based on up-to-date web-based open source software packages. While the concept is related to the OR-World project (see [30]) we extensively extend the concept as described in, e.g., [35, 36]. The implementation based on up-to-date technologies that follow international standards and specification increases the world-wide acceptance and (re)usability of existing learning material. Furthermore, the system is not using proprietary components but is compatible with different operating systems as well as free of external copyrights.

Figure 6 shows the system architecture. On the left-hand side the structure of the learning material is shown, which is stored within (distributed) files. Several databases are used to keep information about the meta-data, user profiles, navigational information like learning paths, and further information about several components like communication. The learning material is processed by special software (Cocoon [3] in connection with the Tomcat-Servlet-Engine [4]) and the XSLT-specification (e.g., eXtensible Stylesheet Language for Tranfor-

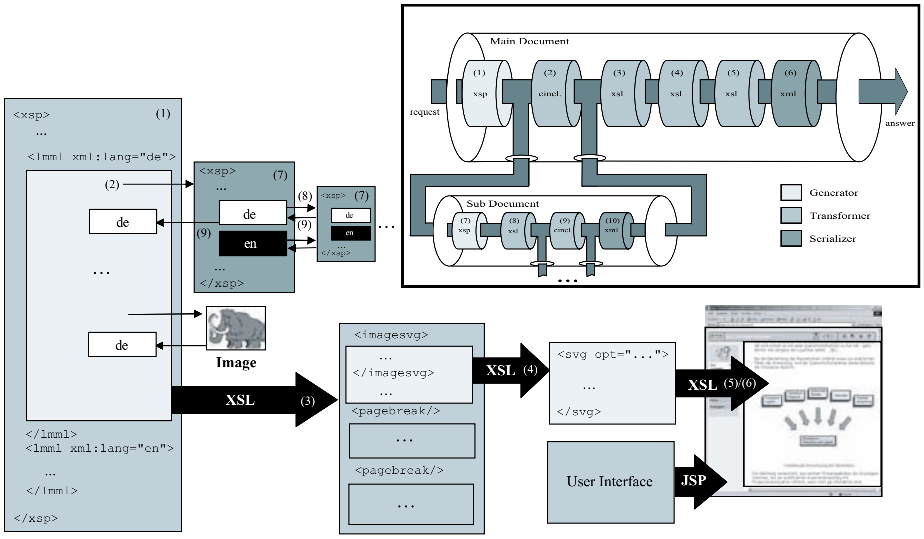


Figure 7: Transformation process of the learning material

mations; [48]) into the requested format, which depends on the kind of usage, the settings of the learner and the medium for the presentation. For example, slides are presented within the browser using a larger font while the same material can also be transformed to a pdf-document with six slides on one page given as a handout to the learners. The result of the transformation is passed to SmartFrame where it is integrated into the VLE; see Figure 7 for a description of the transformation process. Based on a request—e.g. the visualization of a learning object with a given identifier—the corresponding XML-based file for the learning object is taken and processed using a pipeline-based transformation process. The pipeline is shown in the framed box whereas the numbers correspond with the schematic view of the actual processes. The following steps are initiated, numbers in brackets refer to the numbers in the figure. For a detailed description of the transformation process see [36].

- The file of the requested learning object is used as the source of the transformation process (1). The file is generally given as an XSP-file (eXtended Server Pages; XSP-documents can include Java-code and are part of the Cocoon technology). The Java-code is mainly used to access the database for information about the learning object and to directly include other learning objects based on the meta-data. This is also called the generator starting a new pipeline.

- Due to the hierarchical structure each learning object can have references to other objects, which have to be included. This is done by (recursively) integrating the results of other pipelines (2). Based on an XSP-document (7) the sub-document in the requested language is extracted and serialized into XML (8,10), whereas further included references to other learning objects are resolved by opening further pipelines (9).

- After combining the fragments of the (recursively) referenced learning objects regarding the preferred language as well as the requirement of adding elements for the inclusion—i.e., using HTML-tags to include SVG-graphics—, three further transformation processes are initiated. The requested part of the document—pagebreaks (3) used to divide a semantic unit in units to be presented on one page; see also Section 2.3.1—has to be extracted and transformed in two consecutive transformation processes (4,5) according to the user-specific configuration.

- The result of the transformation is serialized, i.e., converted to the requested and from Cocoon supported output format like HTML, XML, or PDF (6).

The main aspects of this system architecture are its direct transformation of the learning material according to the request and configuration of the learner, direct influence of the meta-data of the learning objects, and modularity regarding the whole transformation process. The system architecture shows a client-server-architecture where the learner has to be online to access the learning material. A concept for an offline version incorporates a local installation of the software package with a limited set of the learning material. Within a replication process the required learning materials as well as the personal configuration are synchronized with the server such that the learner can change between on- and offline mode—and its working place—keeping a status quo regarding the VLE.

3 Course Design

In this section we sketch the design of two learning units: Simulation and meta-heuristic. Section 3.1 describes the application SimTool, a simulation tool to be used specially for education within the classroom and the self-paced study. Furthermore, we demonstrate the learning method synchronized blended learning using SimTool as an example. In Section 3.2 a basic course introducing the principles of local search and meta-heuristics is outlined and concluded with an outlook how the learner can be motivated to apply the learned theory on realistic problems.

3.1 Simulation

The subject simulation in conjunction with meta-heuristics is well suited to demonstrate our approach of teaching OR/MS due to the high degree of required practical experiences. As mentioned above, it is not sufficient to offer an extensive theoretical background without doing practical exercises. Only the application of gained knowledge as well as solving real-life problems during the processing of given exercises allows an intensive internalization of the learned material and, therefore, building a sufficient qualification for real-life scenarios.

3.1.1 Simulation Tool SimTool

Teaching simulation means training the recognition of relevant relations within a given scenario, extracting the main components, modeling reality in a "simplified" way as a simulation model, and performing experiments by adjusting the model or incorporating new elements. Therefore, the focus during the development of a simulation tool is mainly set on "practical" education, which

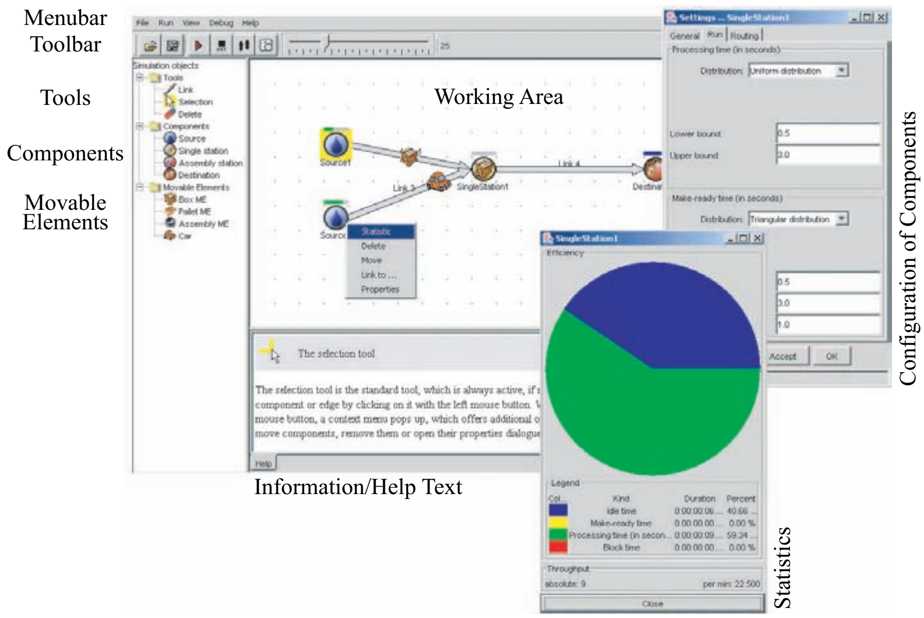


Figure 8: SimTool: Application for interactive education in simulation

means that the presentation of the theoretical learning material has to be specially prepared in terms of interactivity. Here, we concentrate on the two main aspects: Step-by-step introduction into the subject—whereas the main focus is set on the description of components and their relation—as well as guided (and controlled, respectively) interactive solving of exercises and exam questions. On the other side, the simulation tool follows the principles of discrete event-driven simulation and, therefore, can be used in the same context as commercial (and proprietary) software packages like emPlant (whereas the functionality in SimTool is currently not as advanced). Note that the main focus is not set on efficiency, creating large simulation models, or a large amount of functions but rather on demonstrating basic principles.

Within this section, we describe the simulation tool including the design and the implemented prototype. Section 3.1.2 exemplifies the usage by outlining an interactive example where the learner is guided towards the concrete goal of the learning unit: Configuration of an existing gas station model. Finally, Section 3.1.3 describes the integration into the synchronized blended learning concept.

Teaching and learning simulation requires—besides good lectures—practical demonstrations and exercises. In a preliminary study learners were asked to identify the requirements of a discrete event-driven simulation tool that can be used within the classroom, for exercises as well as exams. The list included standards like a simple but powerful user interface, appealing graphics, and visualization of all processes but also further features like lesson specific configuration, feedback to performed actions, guidance by the application while solving a problem, and adaptation of the difficulty level regarding the learner's background knowledge.

One of the main features of SimTool is the possibility of being able to configure basically everything. Next to selecting the language of the interface, Lesson Definition

```
# Available tools
tools = Select

# Menus
menuFile = Save,SaveAs,Exit
menuHelp = About
menuDebug = EH,ObjectList
menuView = Comment,Toolbar,Zel,Grid,GridOptions

# Buttons on the toolbar
buttons = Save,Start,Init,Simspeed

# Localization
country = GB
language = en

# Properties for the source component
Source.popup = Properties,Statistic
Source.props.general = id,createelements,maxelements
Source.props.run = mkreadytimes,createInt

      ⋮
```

Action Definition

```
<es:action name="Finish">
      ⋮
<conditions connector="and">
  <conditions connector="or">
    <condition srcId="Entrance"
               function="getMaxME"
               classType="instance"
               condType="guiStatebutton">
      <returnValue compareOperator="==" type="stdType">
        <value>0</value>
        <type>java.lang.Long</type>
      </returnValue>
    </condition>
    <condition srcId="Entrance"
               function="getMaxME"
               classType="instance"
               condType="guiStatebutton">
      <returnValue compareOperator="&gt;=" type="stdType">
        <value>1000</value>
        <type>java.lang.Long</type>
      </returnValue>
    </condition>
      ⋮
</es:action>
```

Figure 9: Snapshot of the lesson configuration and action script

the degree of functionality can be changed using configuration files, also called lessons. The user interface (menus, toolbar, list of components to build simulation models, popup-menus for the components as well as items within the popup-menus) can be adapted; see also Figure 8 for a screenshot. Besides modifying the interface and degree of interactivity, the simulation model and, therewith, the lesson itself, can be controlled by using a script-oriented event-action mechanism to guide the learner according to the goal of the lesson. Depending on the type of XML-based scripts and events to trigger certain actions the lesson can either be a demonstration of a scenario to introduce the learner to the subject or be similar to a game where the learner has to solve a problem by "playing" with the simulation tool. In the latter case, the event system is used to detect major errors by the learner while allowing as much free exploration as possible. Whenever the learner changes the model by, e.g, adding components or changing attributes, the values are checked on correctness and, if necessary, an appropriate feedback is given. Note, that a feedback can also be positive such that the learner can see a progress in solving the problem and stays motivated to find the solution. See Figure 9 for an example of the lesson configuration (left-hand side) and script language (right-hand side). The lesson only allows the usage of the selection tool, i.e., to change properties of a component, whereas general changes to the simulation model are not allowed. That is, new components can not be added, existing ones not be deleted or moved, and connections between the components are unchangeable. Furthermore, the menus, the buttons in the toolbar, the language, and the available elements of components within the load model are specified. The script language shows an example where the status of the simulation model is checked according to the correct setting. The script is executed whenever called by an event, here in case of the finish button being pressed.

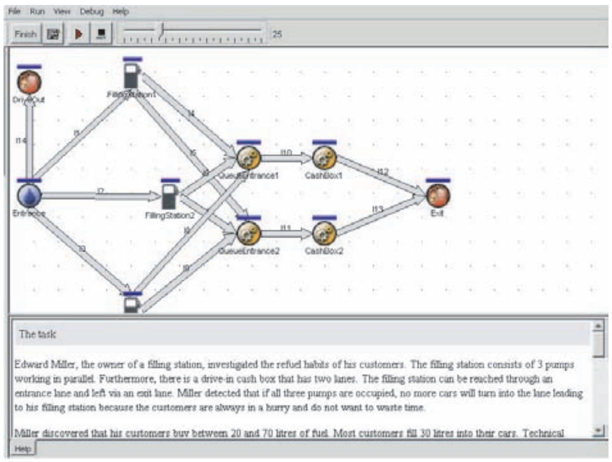


Figure 10: Simulation lesson to learn the configuration and extension of simulation models

The XML-based script language can be used to do everything a learner could do with the user interface. Therefore, a lesson could start with a script building parts of a simulation model asking the learner to finish it by inserting the missing components. The event-action mechanism allows to react on events, e.g., inserting a new component, changing the attributes of a component, or fulfilling a certain outcome of production of goods. Another example could be that the learner is asked to insert the missing components while events could be used to show messages whenever the learner is doing the right actions but also prevent—including the display of a message—existing parts of the model from being changed.

3.1.2 Basic Course Design

For ease of exposition we restrict ourselves to describing the case where the difficulty level is set for beginners. The course is split into two parts: Configuration of a given model and extension of the resulting model according to a given task. First, the problem formulation is given together with the starting simulation model. Here, a gas station owner investigates the customers by writing protocols of their behavior. Based on these observations, the owner constructs the simulation model as shown in Figure 10, which has to be configured by the learner. This has to be done by setting the appropriate values for the existing components. The first part of the lesson allows the learner to understand the problem formulation, to become familiar with the components of the simulation model, and to transfer the knowledge from previous lessons about the components as well as the basics in statistics in order to set the correct configuration values.

Due to the setting of the simulation tool in the first part of the lesson, the learner does not have the possibility to change the model. Besides the popupmenus for changing the configuration of components and showing the statistics, only the play, stop, speed, and finish buttons are available. During the configuration process the learner can freely interact with the model to evaluate the results of the settings. Here, no special events are defined even though the author of the lesson could have used certain events to prevent truly impossible settings like negative values for the appearance of customers. The learner indicates the final solution by pressing the finish button. The system then analyzes the setting and reports whether the solution is correct. A correct solution—according to the problem formulation—will cause the simulation to run showing the results of the settings. Afterwards, this model is used as the starting model for the second part of the lesson. In case that the learner has not found the correct setting, a message box indicates the error together with a possible remark that the feedback level was adjusted. The degree of the help offered by the simulation tool is specified by the current feedback level. In this example, the feedback level is increased each time when three incorrect answers are given allowing the learner to access further information. According to the feedback level, the learner gets 1) information about the type of the components, 2) description about the configuration possibilities, and 3) detailed information how to set the values. Finally, if the learner is still not able to provide the correct solution, the configuration is automatically set with the correct values by the system such that the learner can continue with the lesson. Note that incorrect solutions have to be protocolled to grade the learner.

The second part of the lesson is on a higher level. Here, the learner has to improve the simulation model by changing the structure, e.g., by adding another pump or cash box. The learner has to chose the best alternative for a configuration whereas both the cost and the throughput are used to measure the quality of the configuration. The settings for the distributions are fixed due to the fact that it would be trivial to increase the revenue by allowing more cars to use the gas station having faster pumps. Compared to the previous scenario the learner can freely explore different settings and is not restricted by a limited set of tools and components. However, we use the event action mechanism to warn about changes of the simulation model that are diverging from the problem formulation. In case that the learner deletes all pumps the system shows a warning but does not necessarily prohibit the action. Also other changes like adding too many pumps or cash boxes results in warnings, whereas further size restrictions for components have to be defined.

The main advantage of a course design as described in this section is the increasing difficulty during the lesson, a feedback system allowing the learner to receive appropriate help during the solution process, the exercise how to read and interpret a textually presented problem formulation, the transfer to a working simulation model, and the creative process where the learner has to extend the model according to a given objective without being restricted to a predefined solution. Defining the objective as the maximization of the revenue the learner can, e.g., reduce the cost while serving a similar number of customers or invest money to serve more customers. The simulation tool SimTool—due to the script language and event action mechanism—offers the author of lessons a high degree of freedom to define lessons in which the learners rather learn-by-doing than just memorizing the learning material for the exam.

3.1.3 Synchronized Blended Learning

The learning material has to be developed with respect to fulfilling different requirements of learners. Some learners prefer to learn in a classroom listening

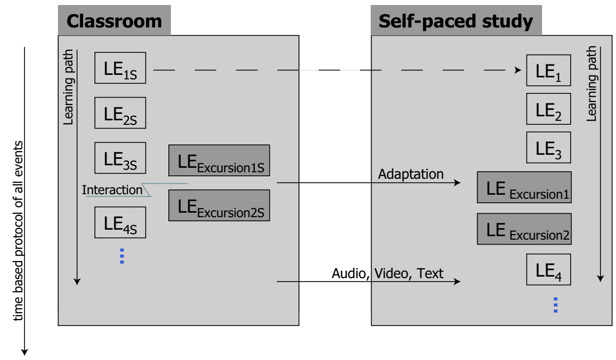


Figure 11: Concept of synchronized blended learning

to the lecturer of the course, while others have time-schedules or other reasons like child care or illness, which do not allow them to attend the lecture regularly. These learners might be restricted to use other learning methods like the self-paced study in VLEs. The concept of SmartFrame integrates several components supporting the learner in respect to allowing a certain degree of freedom selecting the preferred learning method as well as using a unified form of presenting the learning material independent from the didactical form of presentation. Furthermore, the system architecture of SmartFrame allows the integration of synchronized blended learning as introduced by [17]. This concept is based on blended learning—a term mainly used for marketing reasons—being the combination of learning methods (e.g., classroom presentation or virtual seminar) and didactical concepts allowing the learner to select the preferred form of studying the learning material. Compared to the blended learning where the different learning methods are offered in an asynchronous way, synchronized blended learning integrates an adaptive synchronization process to synchronize the learning material almost instantly. Besides having a choice of the learning method, SmartFrame is supposed to be used within different scenarios like classroom, virtual seminar, or self-paced study and, therefore, reduces the number of media changes during the learning process. The following two examples show how synchronized blended learning as well as the simulation tool SimTool can be used within a course in simulation.

The classroom presentation consists of the learning elements LE_{iS} whereas these are shown as slides by the VLE; see also Figure 11. Furthermore, each slide corresponds with a learning element in the self-paced study. After presenting slide LE_{3S}, the lecturer gets interrupted by a question from the audience (interaction). The lecturer can either answer the question directly or present an excursion using slides not being planned to be part of the presentation (i.e., $\mathrm{LE}_{Excursion1S}$ and $\mathrm{LE}_{Excursion2S}$). Afterwards, the presentation is continued with slide LE_{4S} on the originally planned learning path. These interactive changes of the learning material are unknown to learners who are not able to participate in the classroom session. Therefore, the self-paced course—as well as other media like the source of printed material—has to be modified synchronically by adding special learning elements containing the textual counterparts of the slides. These learning elements represent default descriptions of the slides and have to be seen as temporarily integrated elements that have to

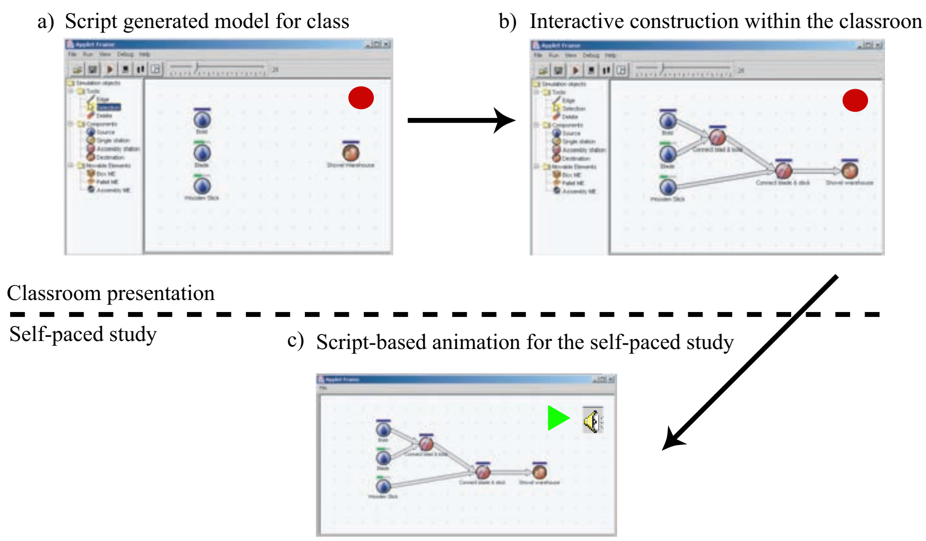


Figure 12: Synchronized blended learning using SimTool

be modified within a later rewriting process adapting the presentation to the existing self-paced study learning material. Furthermore, the learning path is altered such that the extra learning objects are integrated in the existing learning material, being marked as an excursion. Additionally, recordings of audio and video sequences from the classroom—the learner asking the question as well as the lecturer giving the answer—are projected to the VLE allowing the distant learner to get the same information as the attending ones. As mentioned before, the predefined learning elements have to be further integrated by rewriting or adding further learning elements from other sources like learning material databases, whiteboards, or discussion forums.

Another example of synchronized blended learning is shown in Figure 12. The simulation tool is used within the classroom to demonstrate the modeling as well as the analysis of simulation processes. Based on a problem formulation the lecturer is interactively constructing the simulation model whereas the learners should be intensively integrated by asking questions about the modeling. Assuming that a gas station has to be designed based on a given scenario description. Using SimTool allows to use special features to improve the course presentation. For example, the script-language can be used to construct the initial simulation model step-by-step. The lecturer can explain each step as well as reuse the model for subsequent extensions during the course. Furthermore, the quality of the lecture can be improved taking advantage of the event-based system. While it is tiring to see the insertion and configuration of every pump the trigger can be set in a way that it fires as soon as the first pump is fully integrated into the simulation model. Therefore, the lecturer can extensively explain the process in the beginning being able to use "pre-coocked" components—the last two pumps are inserted by scripts—to continue the lesson.

During the lesson all steps have to be protocolled including the ones where "errors" were corrected. Further documentation can be provided by recording the audio or video sequences within the classroom. These recordings—the protocol as well as the other media files—can be transferred to various forms of the learning material as well as leaning methods, e.g., the self-paced study where the material is shown in a demonstration mode. While watching the model being developed in the same way as in the classroom, the learner can listen to the audio or view the video sequences. Compared to traditional virtual courses, the learner can listen to the classroom discussion while seeing the development of the simulation model and, therefore, achieves a better understanding of the lesson. There are similar concepts where interactive learning material is presented as a classroom presentation, but these courses are not synchronized with classroom lessons, especially with respect to modifications. Furthermore, learners from the classroom are able to repeat the lesson without loss of information to internalize the learning material. Another approach to use synchronized blended learning is the training and intensification of learning material, which is synchronized with the classroom, e.g., transfer of the learning path or developed simulation models by the lecturer.

3.2 Meta-Heuristics

In this section we introduce our course unit about meta-heuristics. Meta-heuristics are important modern tools for optimization, but so far, they lack an appropriate integration in lectures as well as practical applications. Most traditional presence courses either do not comprehensively teach meta-heuristics at all or teach meta-heuristics just using textbooks—where meta-heuristics currently are not included or only briefly mentioned—in classes with limited discussion opportunities and also transfer only static learning content into the VLEs. As a result of this presentation the main advantages compared to other optimization methods are neither sufficiently motivated in most cases nor sufficiently illustrated using good examples that are based on real-world scenarios. That is, meta-heuristics still have a shadowy existence with the exception of some famous algorithms like the genetic algorithms.

Our course on meta-heuristics within the VLE has to introduce the learner to the basics assuming no specific prerequisites or previous knowledge about the subject; see Section 3.2.1. Similar to simulation, a purely theoretical course is not preferred regarding the internalization process (see Section 2.2) and, therefore, we demonstrate an approach where the VLE is combined with an external framework to run experiments on realistic problem instances; see Section 3.2.2.

3.2.1 Meta-Heuristics—Basics

Here, we only outline the design of the course, see [34, 37] for a detailed description. The following definition can be used to motivate the learner whereas the learning units themselves have to explain the components as well as their interactions step-by-step. The formal definition of meta-heuristics is based on a variety of definitions from different authors. Basically, a meta-heuristic is a top-level strategy that guides an underlying heuristic solving a given problem. That is, a meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate iteratively a complete (or incomplete) single solution or a collection of solutions. The subordinate heuristics are, e.g., high (or low) level procedures, simple local search, or just a construction method. Meta-heuristics

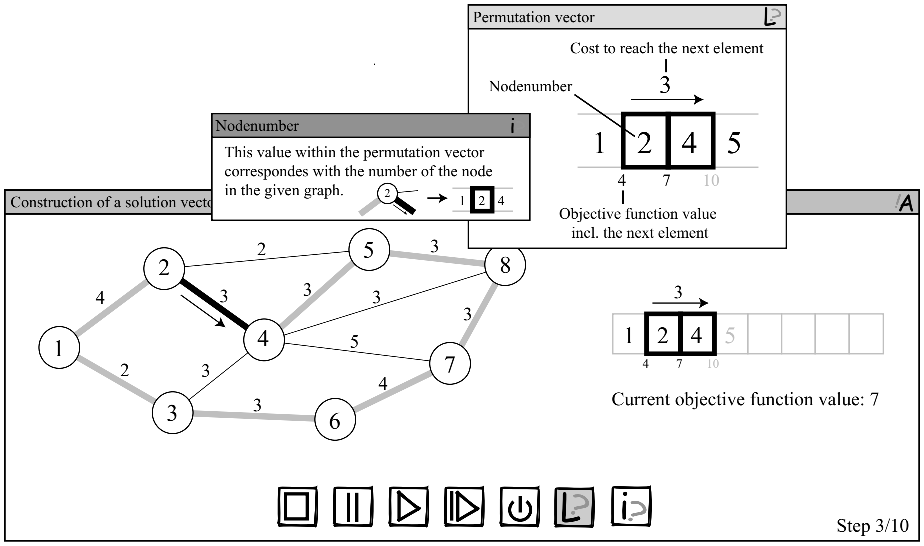


Figure 13: Visualization of the transformation process between the problem and the solution representation

may use learning strategies to structure information in order to find optimal or near-optimal solutions efficiently; see, e.g., [46, p. IX] as well as [18, 31, 45].

Within the course about the basics, the general idea of optimization and heuristics has to be presented. Assuming a given problem terms like problem formulation, solution space, objective function (value), feasibility, feasible and optimal solution, complete enumeration, complexity, and heuristic have to be explained. The learning process is supported by showing specific problem-based examples such that the terms are not only described theoretically using mathematical notation. For example, the idea of searching the solution space can be visualized by showing a ball being placed on a peak—solutions with a bad objective function value. The analogy with the search process is demonstrated by releasing the ball, which then rolls downwards into the nearest valley. If the ball gets no external impulse it will stay in the valley representing at least a local optimum.

Afterwards, the main components of meta-heuristics have to be explained and demonstrated. In addition to the previous list terms like move, neighbor, and neighborhood are of relevance as well as the relation between them. Figure 13 shows, e.g., how the transformation of the problem formulation—here given as a graph for the traveling salesman problem—into a solution (permutation) vector can be done. Within the interactive graphic the learner can construct step-by-step the solution whereas every step is visualized in the graph as well as in the solution. Extended help is provided if the learner uses the button L? for a legend being displayed in new windows. Here, all components and visualized information are explained in details. Further visualizations of other components can be found in [34, 37], especially for the relation between solution and neighbor as well as neighbor and neighborhood. The course to be designed will focus on the meta-heuristics tabu search and simulating annealing including their variations; see also [38, 45].

Figure 14: Applet to visualize the internal processes of meta-heuristics

3.2.2 Meta-heuristics—Applying Theory

After internalizing the basics the learner should apply the knowledge to problem instances. Here, we distinguish two steps: (1) Having small instances to demonstrate the behavior of the meta-heuristics within a controlled environment and (2) providing an environment for experiments where the learner can apply algorithms from an optimization class library.

We developed an applet allowing to visualize meta-heuristics, see Figure 14. The example shows first improving descent on a traveling salesman problem with a solution being coded as a permutation vector. On the left-hand side are the current solution, the so-far best solution, and a history of all previously visited solutions. The neighborhood of the current solution is shown as a list of solution vectors in the middle column. The right-hand side represents the user interface providing access to diagrams about the objective function value development, a textual report, and controls for the optimization process. Further help, instructions, or other information are shown in the frame on the bottom. The next step is indicated by coloring the corresponding solution red—whereas the learner can choose its own solution by double clicking a solution in the neighborhood—and, after selecting the button "next step", also animated by moving the solutions to their new position. Other meta-heuristics are visualized in the same manner. Depending on the type, further information like the current temperature and, therewith, the probability of selecting a move (simulated annealing) or the status and content of the tabu list (tabu search) has to be shown.

The previously described learning unit would be appropriate for beginners not having a detailed background in OR. In case of advanced learners who want

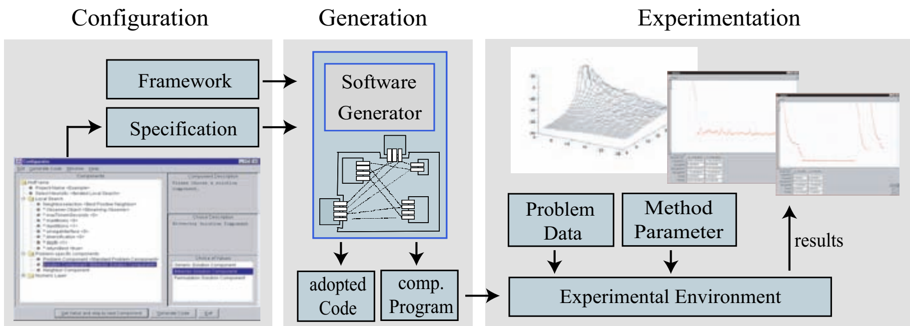


Figure 15: Concept how the leaner can transfer the knowledge to practical problem instances

to learn about new algorithms or need to look up details, the illustrative presentation is both too long as well as not detailed enough. For this group, a learning unit needs a clear focus and a profound explanation using, e.g., mathematical formulas.

Subsequent to "solving" small problem instances the learner has to gain practical experiences by solving non-trivial problems to achieve certain OR-skills. This is done by providing a special interface where the learner selects and configures components from an optimization class library using a specialized applet. We use the framework HotFrame (Heuristic OpTimization FRAMEwork; [13]). The framework supports both adaptable components that incorporate local search based meta-heuristics and an architectural description of the collaboration among these components and problem-specific complements. Methods such as steepest/first descent, simulated annealing, and tabu search are included next to evolutionary methods, variable depth neighborhood search, candidate list approaches, and some hybrid methods.

The configuration of the main components of the meta-heuristics like the solution representation, the move, and the neighborhood according to the problem type is done using a configuration tool in form of an applet. Therefore, we obtain an automatically generated source code (in C++) with todo-parts that have to be replaced with special source code from the learner. The todo-parts can range from a simple method to read the problem formulation up-to a complete new solution representation. Afterwards, the resulting algorithms are applied to (realistic) problem instances, e.g., from ongoing research projects or Internet libraries; see, e.g., [6].

Figure 15 shows the complete sequence that the learner has to perform whereas the learner is supported by step-by-step instructions within the learning environment. Using a software generator for the framework the learner needs to specify the problem as well as algorithmic details by selecting predefined meta-heuristic components to be used in the later program. The generator itself does not output a final program but a customized source code, which has to be completed by the learner. Finally, the VLE provides an environment to execute experiments where the compiled program is executed on several problem instances with a variation of specified parameter settings for the meta-heuristics. The results are collected and can be evaluated by the learner using several statistical methods and visualization forms.

The configuration tool can only grasp certain adaptations of the source code. Therefore, an incremental application process (adoption path; see [14]) has to be enabled such that the learner can successively extent the initial search process. The adoption path includes three to four steps: Coding the computation of the objective function, implementing an adaptive computation of the move evaluation, which replaces the default evaluation by computing objective function values for neighbor solutions from scratch (in this context, one may also implement some move evaluation that differs from the default one, implied change of the objective function value), defining and implementing a new neighborhood structure or an adapted tabu criterion by specific solution information or attribute components, and extending the meta-heuristic itself.

4 Conclusion

The presentation of learning material within VLE requires different concepts than in traditional classroom presentations or textbooks. Especially, it is important to incorporate interactive components to enrich the learning material such that an improved learning experiment is created where the learner is motivated to participate and go even further than necessary to pass the final exams. Having courses about the theory is necessary but real internalization of the learning material can only be achieved if the learner can apply everything in realistic and understandable scenarios.

In this paper we presented our motivation to design a new VLE with new (and innovative) components that should support the learner as well as the instructional designer. Although the VLE SmartFrame is a prototype, we demonstrated important features like SmartBars, the hierarchical structured learning material, and synchronized blended learning. Furthermore, we sketched two virtual courses in the field OR/MS. Both fields are well suited to be realized as an interactive course due to their practical orientation. The course in simulation is enhanced by the simulation tool SimTool, which allows lesson designs based on XML-scripts as well as an event-action mechanism to guide the learner. The course of meta-heuristic is special insofar as there is no real coverage of that subject; neither in textbooks nor in VLE. In addition, we described a concept how the learned basic knowledge can be applied to real-life problems using an experimental environment including the usage of optimization software class libraries.

The ongoing research has to prove if the concept is valuable for the learner as well as e-learning in general. Besides generating complete virtual courses in both fields the subjects have to be integrated into OR/MS by a general OR-course as well as combined in a course about online-optimization. Classic but also advance evaluation methods will be used for extensive validation of our research.

References

[1] M. Alavi, G.M. Marakas, and Y. Yoo. A comparative study of distributed learning environments on learning outcomes. *Information Systems Research*, 13(4):404–415, 2002.

[2] Apache Software Foundation. Apache Xindice. `http://xml.apache.org/xindice/`, January 2003. Date of last check.

[3] Apache Software Foundation. Cocoon software. `http://xml.apache.org/cocoon`, January 2003. Date of last check.

[4] Apache Software Foundation. The Jarkate Site – Jakarta Tomcat. `http://jakarta.apache.org/tomcat/index.html`, January 2003. Date of last check.

[5] ARIADNE Foundation. ARIADNE project home page. `http://ariadne.unil.ch`, January 2003. Date of last check.

[6] Beasley, J.E. OR-library. `http://www.ms.ic.ac.uk/info.html`, January 2003. Date of last check.

[7] P.C. Bell. Teaching business statistics with Microsoft Excel. *INFORMS Transactions on Education*, 1(1):18–26, 2000. `http://ite.informs.org/Vol1No1/bell/bell.html`.

[8] U. Björck. List of virtual universities. `http://www.ped.gu.se/ulric/vus.html`, January 2003. Date of last check.

[9] Blackboard, Inc. Blackboard. `http://www.blackboard.com`, January 2003. Date of last check.

[10] R. Cover. Shareable Courseware Object Reference Model Initiative (SCORM). `http://xml.coverpage.org/scorm.html`, January 2003. Date of last check.

[11] H. Dahmer and J. Dahmer, editors. *Effektives Lernen: Anleitung zum Selbststudium, Gruppenarbeit und Examsvorbereitung*. Schattauer, Stuttgart, 4. edition, 1998.

[12] J.R. Evans. Spreadsheets as a tool for teaching simulation. *INFORMS Transactions on Education*, 1(1):27–37, 2000. `http://http://ite.informs.org/Vol1No1/evans/evans.html`.

[13] A. Fink and S. Voß. Hotframe: A heuristic optimization framework. In S. Voß and D.L. Woodruff, editors, *Optimization Software Class Libraries*, pages 81–154. Kluwer, Boston, 2002.

[14] A. Fink, S. Voß, and D.L. Woodruff. An adoption path for intelligent heuristic search componentware. In E. Rolland and N.S. Umanath, editors, *Proceedings of the 4th INFORMS Conference on Information Systems and Technology, INFORMS, Linthicum*, pages 153–168, 1999.

[15] C. Frank. Studentische Erwartungen an computergestützte Lernumgebungen: Eine empirische Erhebung, 2001. Presented at Symposium Operations Research 01, Duisburg, Germany.

[16] C. Frank, S. Kassanke, and L. Suhl. Meeting students expectations and realizing pedagogical goals within the development of a virtual learning environment. In *Proceedings of Elearn, World Conference on E-learning in Corporate, Government, Healthcare, & Higher Education*, pages 1124–1129, Montreal, Canada, October 2002.

[17] C. Frank, T. Reiners, L. Suhl, and S. Voß. A Conceptual Framework for Synchronized Blended Learning on the Web. Technical report, University of Technology Braunschweig, 2002.

[18] F. Glover and M. Laguna, editors. *Tabu Search*. Kluwer, Boston, 1997.

[19] R. Hazemi, S. Hailes, and S. Wilbur, editors. *The Digital University: Reinventing the Academy*. Springer, London, 1998.

[20] IEEE Learning Technology Standards Committee. Standard for Information Technology: Education and Training Systems – Learning Objects and Metadata. `http://ltsc.ieee.org/wg12`, January 2003. Date of last check.

[21] IFORS. tutORial. `http://www.ifors.ms.unimelb.edu.au/tutorial`, January 2003. Date of last check.

[22] IMS Global Learning Consortium. IMS Implementation Handbooks. `http://www.imsproject.org/implementationhandbook/index.html`, January 2003. Date of last check.

[23] IMS Global Learning Consortium. IMS Question & Test Interoperability Specification. `http://www.imsproject.org/question`, January 2003. Date of last check.

[24] Institute for Computing Systems Architecture, University of Edinburgh. *simjava*, January 2003. Date of last check.

[25] Learning Lab Lower Saxony. About L3S. http://www.learninglab.de/english/news/about_the_L3S.html, January 2003. Date of last check.

[26] Massachusetts Institute of Technology. Educational uses of the WWW at MIT. http://web.mit.edu/is/courseweb/courses.html, January 2003. Date of last check.

[27] J.A. Miller, A.F. Seila, and X. Xiang. The jsim web-based simulation environment. In *Future Generation Computer Systems (FGCS), Special Issue on Web-Based Modeling and Simulation*, volume 17, pages 119–133. Elsevier North-Holland, October 2000. http://chief.cs.uga.edu/ jam/jsim/papers/mailfiles/simjournal.ps.

[28] Multibook. Multibook. http://www.multibook.de, January 2003. Date of last check.

[29] MySQL.org. The Site for the MySQL Community. http://www.mysql.org, January 2003. Date of last check.

[30] OR-World. http://www.or-world.com, January 2003. Date of last check.

[31] I.H. Osman and J.P. Kelly. Meta-heuristics: An overview. In I.H. Osman and J.P. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 1–21. Kluwer, Boston, 1996.

[32] W. Reiß. JEWEL: Java Enabled Worldwide Economic Learning. http://wiwi.uni-paderborn.de/jewel/, January 2003. Date of last check.

[33] W. Reiß. OViSS: Offenes Virtuelles Studien System. http://wiwi.upb.de/~vwl08/OViSS/oviss_current/PBSS2002.html, January 2003. Date of last check.

[34] T. Reiners, D. Reiß, I. Sassen, and S. Voß. Simulation und Meta-Heuristiken als Bestandteil quantitativer BWL-Ausbildung: Von der Technik zur Anwendung. Technical report, University of Technology Braunschweig, 2003.

[35] T. Reiners, D. Reiß, and S. Voß. Using hyperbolic trees and SmartBars within virtual learning environment concepts. In *Proceedings of the World Congress Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education (NL 2002)*, volume #100029-03-TR-026, pages 1–7, Millet Alberta, 2002. ICSC-Naiso Academic Press. [ISBN: 3-906454-31-2].

[36] T. Reiners, D. Reiß, and S. Voß. XML-basierte Kodierung von Lernobjekten. Technical report, University of Technology Braunschweig, 2003.

[37] T. Reiners and S. Voß. Teaching meta-heuristics. Technical report, University of Technology Braunschweig, 2002.

[38] C.C. Ribeiro and P. Hansen, editors. *Essays and Surveys in Metaheuristics*. Kluwer, Boston, 2002.

[39] C. Süß. Learning Material Markup Language. http://www.lmml.de, January 2003. Date of last check.

[40] R. Schulmeister. *Lernplattformen für das virtuelle Lernen. Evaluation und Didaktik*. Oldenbourg, 2003.

[41] M. Sniedovich and A. Byrne. tutOR. http://www.tutor.ms.unimelb.edu.au/frame.html, January 2003. Date of last check.

[42] University of Phoenix. University of Phoenix Online. http://online.phoenix.edu, January 2003. Date of last check.

[43] Verbund Norddeutscher Universitäten. Methodenlehre-Baukasten. http://www.izhd.uni-hamburg.de/mlbk/home.html, January 2003. Date of last check.

[44] Virtueller Hochschulverbund Karlsruhe. Vikar: Virtueller Hochschulverbund Karlsruhe. http://www.vikar.de, January 2003. Date of last check.

[45] S. Voß. Meta-heuristics: The state of the art. In A. Nareyek, editor, *Local Search for Planning and Scheduling, Lecture Notes in Artificial Intelligence*, volume 2148, pages 1–23. Springer, Berlin, 2001.

[46] S. Voß, S. Martello, I.H. Osman, and C. Roucairol, editors. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer, Boston, 1999.

[47] W3 Consortium. Mathematical Markup Language (MathML) 2.0 Recommendation. http://www.w3.org/Math, January 2003. Date of last check.

[48] W3 Consortium. XSL Transformation (XSLT). http://www.w3.org/TR/xslt, January 2003. Date of last check.

[49] N. Walsh and L. Muellner. *DocBook: The Definitive Guide*. O'Reilly, Beijing, 1999.

[50] WebCT. WebCT.com. `http://www.webct.com`, January 2003. Date of last check.

[51] C.A. Wedemeyer. *Learning at the Back Door: Reflections on Non-Traditional Learning in the Lifespan*. University of Wisconsin Press, Madison, 1981.