

## DIPLOMARBEIT intelligent Learning Tool (in:LeTo)

Ausgeführt im Schuljahr 2010/11 von:

Tobias Buchberger 5AHITS - 1

David Gruber 5AHITS - 6

Nico Henglmüller 5AHITS - 7

Patrick Höfer 5AHITS - 8

Florian Unger 5AHITS - 20

Betreuer/Betreuerin:

VL Michael Borko, Bakk. techn.

Mag. Dr. Walter Rafeiner-Magor

Wien, am 17.05.2011

---

## Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 17.05.2011

Ort, Datum

---

Tobias Buchberger

---

David Gruber

---

Nico Henglmüller  
(Projektleiter)

---

Patrick Höfer

---

Florian Unger

---

## Abstract

Our diploma project in:LeTo, which is the short form of **intelligent Learning Tool**, helps the user to learn complex issues easily. This is based on the display and administration of the learning content. A standardized interface is used to display miscellaneous contents. This interface enables the user to import and learn new information. As an example of such content we created a package about the programming language Java.

The user's learning progress is continuously displayed as graphical statistics. So every user has access to view his progress anytime and thus can improve permanently. Furthermore, only tasks, which match the actual progress, are given to the user.

To synchronize the collected information between students and teachers, we use a distributed environment without any central server. This was implemented with the Java **Remote Method Invocation** technology.

Our diploma thesis is a part of the diploma project, which has been planned and developed in the fifth grade in cooperation with the project management education.

---

## Kurzfassung

Das Diplomprojekt in:LeTo, kurz für **intelligent Learning Tool**, hilft dem Benutzer komplexe Inhalte leicht zu erlernen. Die Basis dafür ist die Darstellung und Verwaltung des Lerninhalts. Um verschiedene Informationen anzuzeigen, wird eine standardisierte Inhaltsschnittstelle verwendet. Dies ermöglicht es Benutzern, verschiedene Materien zu importieren und zu erlernen. Als Beispiel für einen solchen Inhalt wurde im Zuge des Jahresprojekts eine Behandlung der Thematik über die objektorientierte Programmiersprache Java erstellt.

Der Lernfortschritt des Benutzers wird fortlaufend grafisch dargestellt. So hat der Benutzer jederzeit die Möglichkeit seinen Fortschritt einzusehen und sich stetig zu verbessern. Außerdem werden dem Benutzer nur Beispiele zur Verfügung gestellt, welche seinem Lernfortschritt entsprechen.

Um diese gesammelten Informationen zwischen Schüler und Lehrer zu synchronisieren, verwenden wir eine verteilte Umgebung ohne zentralen Server. Diese Technik wird mit Hilfe von Java **Remote Method Invocation** gewährleistet.

Unsere Diplomarbeit ist Teil unseres Projekts, welches im Zuge des Projektmanagement Unterrichts im fünften Jahrgang geplant und implementiert wurde.

---

## Inhaltsverzeichnis

1	Vorwort .....	9
2	Einleitung .....	10
2.1	Projektteam.....	10
2.1.1	Tobias Buchberger .....	10
2.1.2	David Gruber .....	10
2.1.3	Nico Henglmüller .....	10
2.1.4	Patrick Höfer.....	10
2.1.5	Florian Unger.....	10
3	Aufgabenstellung .....	11
3.1	Kontext der Arbeit.....	11
4	Zielsetzung.....	12
4.1	Meta-Methodik.....	12
4.1.1	Scrum .....	12
4.1.2	Bugtracking & Testing .....	14
4.2	Entwurfsmethode.....	17
4.3	Ziele.....	17
5	Technologien .....	19
5.1	Lernpaketformate.....	19
5.1.1	SCORM – Sharable Content Object Reference Model .....	20
5.1.2	Fazit.....	28
5.2	Java Browser .....	29
5.2.1	Lobo Browser .....	29
5.2.2	The Flying Sourcer Projekt.....	30
5.2.3	JEditorPane .....	30
5.2.4	Jdic .....	31
5.2.5	SWT Browser .....	32
5.2.6	Fazit.....	32
5.3	Künstliche Intelligenz.....	33
5.3.1	KI Einleitung .....	33
5.3.2	Neuronale Netze.....	33

---

5.3.3	Expertensystem.....	34
5.3.4	Fazit.....	36
5.4	Datenbanklösungen.....	37
5.4.1	Berkley DB .....	37
5.4.2	SQLite .....	38
5.4.3	JavaDB.....	39
5.4.4	Gegenüberstellung .....	40
5.4.5	Fazit.....	40
5.5	Oberflächen Frameworks .....	41
5.5.1	AWT .....	41
5.5.2	Swing.....	42
5.5.3	SWT .....	43
5.5.4	Fazit.....	44
5.6	Verteilte Versionierungssysteme .....	45
5.6.1	Versionsverwaltung Allgemein .....	45
5.6.2	Arten von Versionierungssystemen.....	45
5.6.3	Einsatz des Versionsverwaltungssystems bei in:LeTo.....	46
5.6.4	JGit.....	46
5.6.5	Fazit.....	48
5.7	Java Compiler.....	49
5.7.1	JDK Überprüfung.....	49
5.7.2	tools.jar einbinden .....	49
5.7.3	Fazit.....	49
6	Design und Implementierung.....	50
6.1	Logger .....	50
6.2	Grafische Oberfläche.....	52
6.2.1	Login.....	52
6.2.2	Haupt GUI .....	53
6.2.3	Events .....	54
6.2.4	ExpandBar.....	55
6.2.5	Tabview .....	57

---

6.2.6	ContentTree.....	58
6.2.7	Table.....	60
6.3	Standardisierte Inhaltsschnittstelle .....	63
6.3.1	SCORM Package .....	63
6.3.2	Implementierung in Java .....	70
6.4	Datenbank .....	71
6.4.1	ER-Diagramm.....	71
6.4.2	Data Access Object Pattern .....	74
6.4.3	Zugriff auf die Datenbank in in:LeTo .....	76
6.5	Versionierungssystem .....	77
6.5.1	Versionierungssystem des Schülers .....	77
6.5.2	Nutzen für den Lehrer.....	77
6.5.3	Synchronisation der Übungen .....	77
6.5.4	Indizierung von Versionen .....	77
6.5.5	Schreiben und Lesen von Dateien .....	78
6.5.6	Dateien schreiben.....	78
6.5.7	Dateien auslesen.....	79
6.5.8	Die Indexdatei.....	81
6.5.9	Die Exercisedatei.....	82
6.5.10	Daten Verschlüsselung.....	82
6.5.11	Dateistruktur .....	82
6.6	Netzwerk Kommunikation .....	84
6.6.1	Synchronisation von Daten.....	84
6.6.2	Erst-Anmeldung bei einer Klasse .....	84
6.6.3	Synchronisation zwischen Schüler und Lehrer.....	85
6.6.4	Synchronisation zwischen Haupt- und Nebenlehrern.....	86
6.6.5	Klasse löschen .....	90
6.6.6	Netzwerkauslastung .....	90
6.7	Expertensystem .....	91
6.7.1	KnowledgeBase.....	92
6.7.2	Rules .....	93

---

---

6.7.3	RuleBase .....	96
6.8	Duplikatskontrolle .....	98
6.8.1	Verfahren zur Duplikatsfindung .....	98
6.8.2	Implementierung.....	99
6.9	Statistik .....	100
6.9.1	Implementierung.....	100
6.9.2	Statistik als Feedback .....	101
6.10	Benutzerverwaltung .....	104
6.10.1	CSV – Import.....	104
6.10.2	LDAP-Import.....	105
6.10.3	Erzeugung von Activationkeys .....	107
6.11	Kompilieren und Ausführen .....	110
6.11.1	Kompilieren .....	110
6.11.2	Ausführen .....	110
6.11.3	System.exit.....	111
6.12	Funktionalitätsüberprüfung .....	112
6.12.1	Testfälle.....	113
7	Zusammenfassung .....	115
8	Produkthandbuch.....	116
9	Benutzerhandbuch.....	117
10	Anhang A: Zeitaufzeichnungen .....	118
11	Literaturverzeichnis .....	119
12	Abbildungsverzeichnis .....	121
13	Tabellenverzeichnis .....	122