

COMPUTER ASSISTED LEARNING

CAL + CAT

COMPUTER ASSISTED TEACHING

ARBEITSBERICHTE

Nr. 30

**Heinz Lothar Grob
Carsten Fiedler**

Selector

Mai 2008

**PROF. DR. HEINZ LOTHAR GROB
INSTITUT FÜR WIRTSCHAFTSINFORMATIK
WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER**

LEONARDO-CAMPUS 3, 48149 MÜNSTER, TEL. (0251) 83-38000, FAX (0251) 83-38009
E-MAIL: Grob@wi.uni-muenster.de, WWW: <http://www-wi.uni-muenster.de/aw>

Inhalt

1 Die Idee	1
2 Darstellung von Beispielen	2
3 Der Spielablauf	2
3.1 Selector-Spielkonfiguration	3
3.2 Selector-Durchlauf	4
4 Technische Realisierung in Freestyle Learning	7
4.1 Funktionale Anforderungen	7
4.2 Implementierung	8

Nicht nur im Studium, sondern generell kommt es darauf an, die „passenden“ Begriffe zu verwenden. Ein leicht übertriebenes Beispiel soll dies verdeutlichen: Wer beim Thema internes Rechnungswesen von Erträgen und Aufwendungen spricht, obwohl Leistungen und Kosten gemeint sind, erzeugt bei einem (wissenden) Gegenüber Misstrauen über seinen Expertenstatus – nicht nur in der BWL, sondern generell. An der Wahl der richtigen Buzzwords meint man, Experten erkennen zu können. Diejenigen, die Experten sein (werden) wollen, tun gut daran, sich die passenden Begriffe einzuprägen, um sie sinnvoll in die Kommunikation einzubringen. Dies sind natürlich die ersten Schritte zum Experten. :-)

1 Die Idee

Dem *Selector* liegt die Idee zugrunde, Begriffe ihrer zugehörigen Domäne spielerisch zuzuordnen. Das Spiel sieht vor, in gegebener Zeit und unter parametrisch einstellbarem Zeitdruck die Anzahl der erzielbaren Punkte zu maximieren. Durch diese Wettbewerbssituation – auch wenn man dabei (wie z. B. beim Golfen) gegen sich selbst spielt – wird der Wunsch geweckt, immer besser zu werden, also seine bislang erreichte Bestnote zu erhöhen. Damit der Spieler aus seinen Fehlern lernen kann, werden die falschen und richtigen, aber auch die unbearbeiteten Zuordnungen als Feedback-Informationen textlich aufgelistet.

Die Überlegung, die für die Zuordnung von Begriffen gilt, ist auch für *Grafiken* in Form von Bildern oder Symbolen relevant. Durch die Möglichkeit, digitale Fotos selbst zu erzeugen oder aber durch Google-Bilder zu beschaffen, ist das Gestalten von Selector-Dateien einfach. Die Software sieht vor, sowohl Text- als auch Grafikobjekte zu nutzen. Durch individuelle Gestaltung der Hintergrundfarbe für nicht den Rahmen füllende Fotos oder für Texte lassen sich auch eigenständige Felder erzeugen, die den ästhetischen Effekt der Präsentation erhöhen. Selbstverständlich lassen sich auch Selector-Spiele erzeugen, in denen es um die Wahl der richtigen Farben geht.¹

Die Idee zum spielerischen Selektieren von Begriffen und Grafiken wurde bereits 1998 in Form der singulären Anwendung „Selector“ umgesetzt.² Mittlerweile ist die Software als eigenständige Perspektive (View) Bestandteil des Freestyle Learning (FSL)-Produkts.³

Durch die Integration des Selectors in die FSL-Plattform bietet sich die Nutzung weiterer FSL-Views an. So kann durch das Intro unter Verwendung eines Videos eine Einführung des Autors in die Spielregeln und in den Themenbereich des Selectors gegeben werden. Für eine

¹ ... um beispielsweise aus den Farben blau, weiß, gelb und schwarz mit der Hintergrundmusik „Blau und Weiß, wie lieb ich dich“ die richtige Auswahl zu treffen. :-)

² Vgl. Grob, H. L., Heudecker, T., Selector, Arbeitsbericht Nr. 17.

³ Vgl. Grob, H. L., Brocke, J. vom, Lahme, N., Freestyle Learning – Das mediendidaktische Konzept, Arbeitsbericht Nr. 20.

weitergehende mediale Einführung in das Wissensgebiet, das im Selector bearbeitet wird, bietet sich als View die Text Study an. Die im Selector zu präsentierenden Elementen können beispielsweise im Glossar systematisch aufgelistet und inhaltlich erläutert werden. Durch das Setzen von Links – auch in andere Views – können Zusammenhänge deutlich gemacht werden. Die im Glossar enthaltenen Informationen lassen sich durch den Memorizer spielerisch erlernen – dies ist nicht zuletzt ein gezieltes Training auf das Selector-Spiel. In der Check-up-Perspektive können beispielsweise Multiple Choice-Fragen beantwortet und Lückentexte ausgefüllt werden, um den Wissensstoff, der dem Selector zugrunde liegt, zu kontrollieren. Schließlich bietet sich an, in der Slideshow sämtliche im Selector enthaltenen Elemente sukzessiv zu präsentieren und dabei zu kommentieren und bei Bedarf mit einer dezenten Hintergrundmusik zu versehen. Während für den Selector und die Slideshow die Grafiken automatisch an die standardisierte Größe des Ausgabefeldes angepasst werden, ist die Modifikation der Bildgröße bei den anderen Views individuell vorzunehmen, da dort keine festen Rahmen verwendet werden.

Der Vorschlag, um den Selector herum auch weitere Views einzurichten, stellt einen idealen Anwendungsfall dar, um die Grundidee von Freestyle Learning deutlich zu machen, die darin besteht, eine ausgewählte Wissensdomäne aus unterschiedlichen Perspektiven zu betrachten. Die Menge der Wissens Elemente (z. B. der Grafiken) kann dabei gleich sein – lediglich die Perspektiven wechseln in Abhängigkeit von der Interaktion des Nutzers.

2 Darstellung von Beispielen

Zur Einrichtung einer Selector-Datei sind im ersten Schritt zwei Wissensdomänen festzulegen, die miteinander konkurrieren. Dann ist diejenige Domäne auszuwählen, auf die zu klicken ist. Am einfachsten ist es, dies im Dateinamen zum Ausdruck zu bringen. Denkbar ist aber auch, die Kommentarfunktion für die Erläuterung der Spielregel zu nutzen. Auch eine Darstellung im Rahmen der FSL-View „Intro“ oder „Text Study“ ist vorstellbar.

Zur Illustration soll hier ein Beispiel zu den Wissensdomänen „Internes Rechnungswesen“ und „Externes Rechnungswesen“ erörtert werden. Nach der Sammlung der für die konkurrierenden Domänen relevanten Elemente sind diese begrifflich zu definieren bzw. als Grafik zu integrieren. Empfehlenswert ist, mit einer relativ kleinen Anzahl von Elementen zu beginnen und den Umfang später sukzessiv auszuweiten. Für die Domäne „Internes Rechnungswesen“ können beispielsweise die Begriffe Herstellkosten, Akzeptanzprinzip und Deckungsbeitrag gewählt werden. Der konkurrierenden Domäne „Externes Rechnungswesen“ könnten die Begriffe Herstellungskosten, Rückstellungen und Rechnungsabgrenzung zugeordnet werden.

Allgemein gesagt ist der Selector der Kategorie Edutainment-Software zuzuordnen. Durch den spielerischen Charakter und die zeitliche Dynamik soll durch den Selector die Motivation des Lernenden gefördert werden, Bilder, Begriffe und Symbole richtig zuzuordnen zu können. In diesem Kontext stellt der Selector ein Softwareprodukt zum Bereich E-Learning dar. Aber auch außerhalb des Lernbereichs kann die Benutzung des Selectors attraktiv sein. Dies soll

nun an einem einfachen Beispiel demonstriert werden. Bei der Anfertigung eines Geburtstagsgeschenks in Form einer Selector-Datei sollte sich der Autor als Erstes vom „Geburtstagskind“ etliche digitale Fotos (von Babyfotos bis hin zu aktuellen Portraits) beschaffen. Diese Grafiken können mit Fotos „anderer Persönlichkeiten“ (z. B. von Schau- und Fußballspielern) konkurrieren, die nach Möglichkeit ebenfalls nicht nur aktuelle Portraits, sondern auch Jugend- und Baby-Fotos enthalten sollten. Das Spiel, das unter den Klängen des „Happy Birthday“-Sounds ablaufen könnte, heißt: „Klick auf das Geburtstagskind“ und wird sicherlich viel Freude bereiten!

3 Der Spielablauf

3.1 Selector-Spielkonfiguration

Die Selector-Spielkonfiguration in der Freestyle Learning-Plattform erfolgt durch den Autor. Dieser wählt im Rahmen der FSL-Plattform die Selector-View aus und aktiviert den Editiermodus aus dem Hauptmenü. Die einzelnen Spielthemen sind im Strukturbaum anzulegen. Zur Erzielung einer besseren Übersicht können Ordner für Oberthemen eingerichtet werden. Die erstellte Struktur kann jederzeit neu angeordnet werden.

Sobald ein neues Spielthema angelegt worden ist, wird bei dessen Auswahl das Konfigurationsmenü im Anzeigebereich dargestellt. Im Konfigurationsmenü sind die für ein Spielthema gesammelten Elemente in Form von Text- und Grafikelementen entweder einzeln anzulegen oder es ist eine Vielzahl von Elementen zu importieren. Bereits angelegte Text- und Grafikelemente können natürlich auch gelöscht werden.

Die dem Thema zugehörigen Elemente sind als „zulässig“ zu markieren. Auf diese Elemente ist im Selector-Durchlauf zu klicken. Zur Unterstützung der Spielkontrolle sollten die Text- und Grafikelemente editiert werden. Bei den Textelementen ist eine Kurzfassung des Textes zu empfehlen, während bei den Grafikelementen Kurzbezeichnungen eingegeben werden sollten. Ferner besteht im Konfigurationsmenü die Möglichkeit, für den Spielvorgang eine Hintergrundmusik zu importieren.

In Abb. 1 wird ein Beispiel zum Thema „Klick auf Leuchttürme und -feuer von Sylt“ gezeigt.

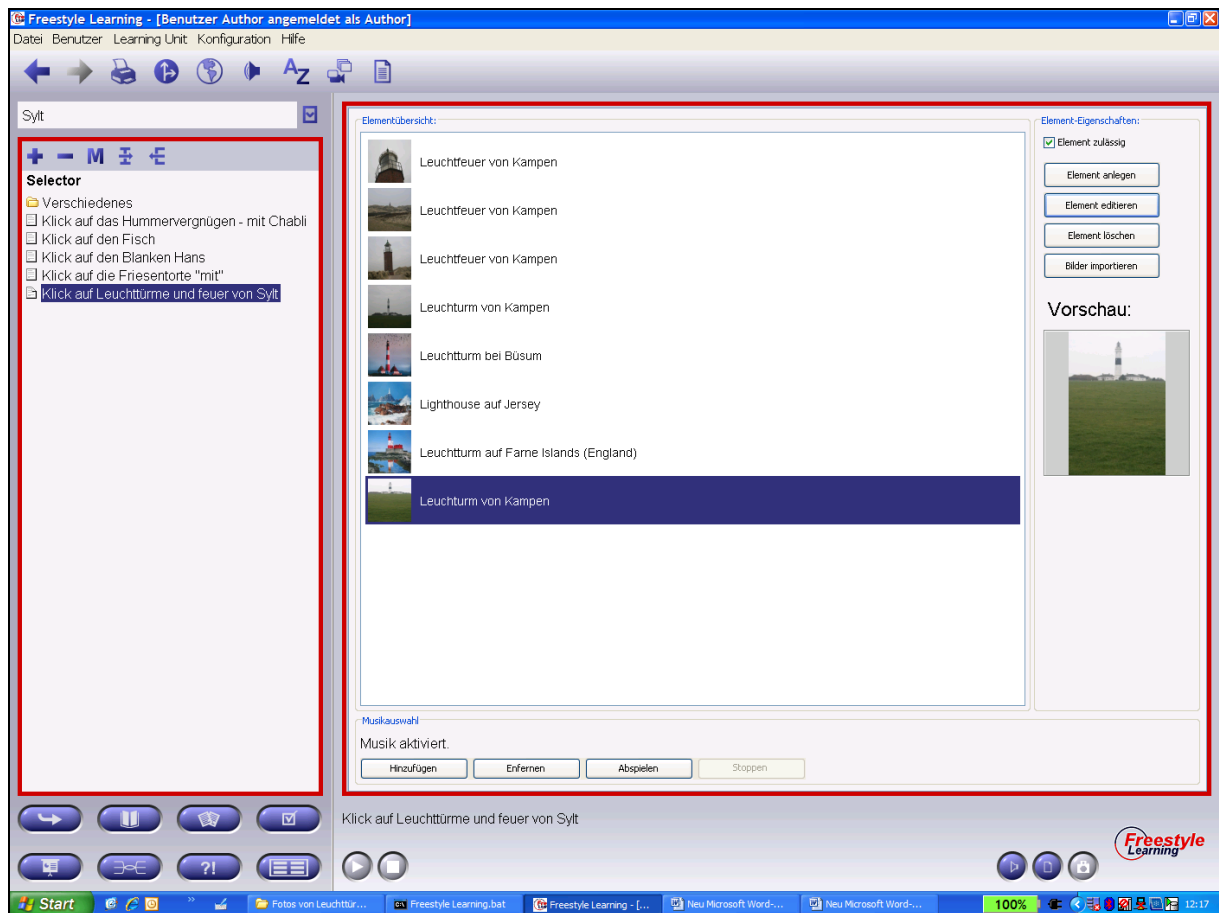


Abb. 1: Konfigurationssicht zum Anlegen von Spieldurchläufen

3.2 Selector-Durchlauf

Nachdem der Lerner die Selector-View aktiviert hat, kann ein Spiel – ein Selector-Durchlauf – gestartet werden, indem ein Themenelement aus dem Strukturbaum ausgewählt und innerhalb der Interaktionsleiste die Starttaste angeklickt wird. In dem nach dem Klick erscheinenden Auswahldialogfenster kann der Spieler die Spielfeldgröße, die Spielgeschwindigkeit sowie die Gesamtspielzeit festlegen. Danach wird das Spiel gestartet.



Abb. 2: Dialogfenster zum Starten eines Selector-Durchlaufs

Das Selector-Spiel findet im Anzeigebereich von Freestyle Learning mit einem dynamischen Wechsel der Bilder statt. Bei einem Klick auf die Taste zur Vergrößerung der Bildschirmausgabe wird der volle Bildschirm genutzt.



Abb. 3: Beispielhafter Selector-View-Spielablauf

Beim Spiel hat der Spieler mit der Maus die relevanten Elemente anzuklicken. Ein Spiel endet entweder nach Ablauf der anfangs angegebenen Zeitspanne oder durch einen Klick auf die „Stopp“-Taste innerhalb des Interaktionsfeldes. Bei Präsentationen im Vollbildmodus muss zur Bedienung der Stopp-Taste in den normalen FSL-Modus verzweigt werden.

Nach dem Beenden des Durchlaufs wird dem Benutzer eine Evaluation über seine erfolgreichen Selektionen präsentiert und die erreichte Gesamtpunktzahl angezeigt. Eine Ausgabe im oberen Teil der Box informiert über richtige und falsche Zuordnungen und übersehene Elemente. Die dort aufgeführten Begriffe sind beim Editiervorgang erzeugt worden.

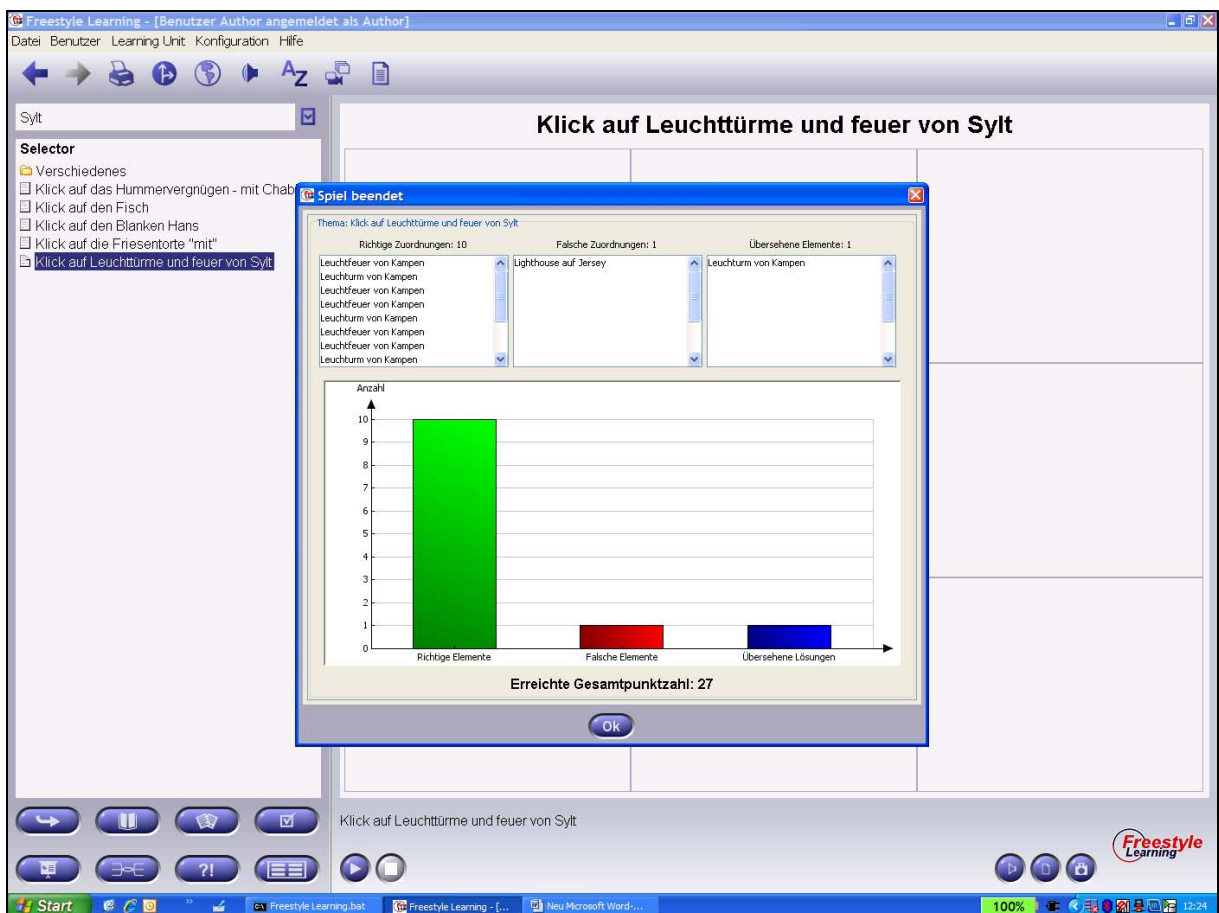


Abb. 4: Evaluationssicht

4 Technische Realisierung in Freestyle Learning

4.1 Funktionale Anforderungen

Die aus dem didaktischen Konzept entstehenden funktionalen Anforderungen der Selector-View werden durch die in Abb. 5 aufgeführten Uses-Cases repräsentiert.

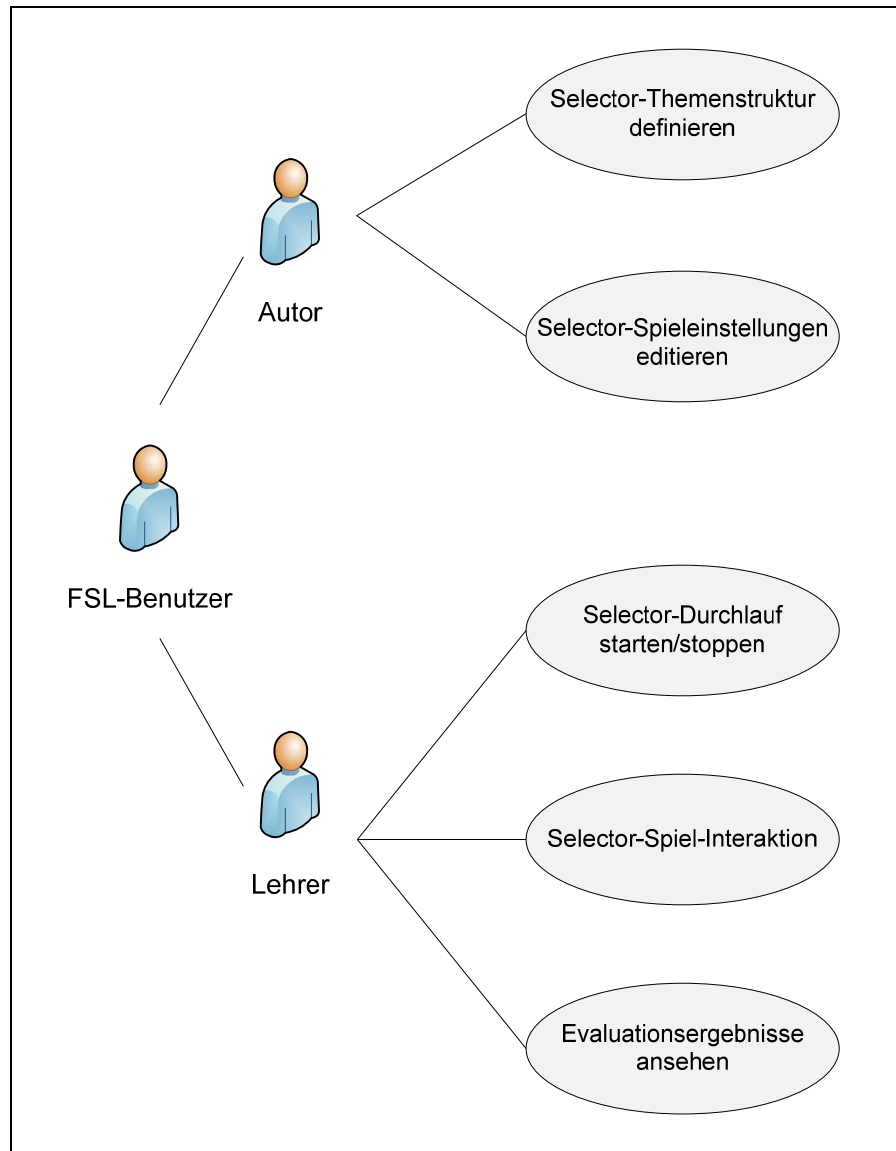


Abb. 5: Selector-View-Use-Case-Diagramm

Das Use-Case-Diagramm zeigt den Benutzer, der die Rolle des Autors oder des Lernalers einnehmen kann, wobei der Autor im Strukturbaum die Spielthemen definiert und im Anzeigebereich die dazugehörigen Spielelemente editiert. Der Lerner hat die Möglichkeit, einen Selector-Durchlauf zu starten, zu stoppen, seine Selektionen im Spielablauf durchzuführen und seine Ergebnisse in Form einer Evaluation anzusehen.

4.2 Implementierung

Die Freestyle Learning-Plattform kennzeichnet sich durch Plattformunabhängigkeit und ist in der Programmiersprache Java geschrieben. Dadurch wurde auch die Selector-View in Java implementiert. Um den didaktischen und technischen Prinzipien zu entsprechen, wurde die neue technische Struktur der Selector-View an die bestehende Learning Unit-View-Schnittstelle (Learning Unit View API) angepasst. Der technische Aufbau einer Learning Unit-View besteht aus Basisklassen, die durch die Learning Unit-View API vordefiniert sind und deren Eigenschaften an die neuen Selector-Klassen weitergegeben werden.

Die Freestyle Learning Unit API stellt sechs abstrakte Java-Klassen zur Verfügung, in denen die essenziellen Eigenschaften (Methoden und Attribute) für die Einbindung einer neuen Learning Unit-View und deren Bestandteile in Form des Strukturbaums, Anzeigebereichs, der Interaktionsleiste und des Statusfeldes vordefiniert werden.¹ Diese Klasseneigenschaften werden durch den Mechanismus der Vererbung an die neuen Learning Unit-View-Klassen weitergegeben. Ferner wird durch den Vererbungsmechanismus eindeutig vorgeschrieben, welche Methoden und Attribute bei der Entwicklung neuer Perspektiven unbedingt implementiert werden müssen, indem die rumpflosen, abstrakten Methoden der Oberklasse innerhalb der davon erbenenden Klasse überschrieben werden und dadurch geerbte Instanzvariablen initialisiert werden.

FSLAbstractLearningUnitViewManager

Die abstrakte Klasse FSLAbstractLearningUnitViewManager bildet die Basisklasse einer Learning Unit-View. In ihr werden alle weiteren essenziellen Klassen der View initialisiert. Ferner stellt sie die Schnittstelle zwischen den View-Klassen und dem Home-Package der Freestyle Learning-Plattform dar. Im Home-Package befinden sich die entsprechenden Manager-Klassen zum Verwalten der einzelnen Learning Units und deren Learning Unit-View-Elemente. Innerhalb des neu erstellten Selector-View-Package erbt die Klasse FLGSelectorManager von der Klasse FSLAbstractLearningUnitViewManager. In ihr werden die entsprechenden Instanzvariablen vom Typ FLGSelectorElements-StructurePanel, FLGSelectorElementInteractionPanel und FLGSelector-ElementsContentsPanel initialisiert.

FSLAbstractLearningUnitViewElementsStructurePanel

Die Klasse FSLAbstractLearningUnitViewElementsStructurePanel stellt die Basisfunktionalität zum Einbinden von Learning Unit-View-Elementen in den Strukturbaum einer Learning Unit-View-Oberfläche zur Verfügung. Innerhalb der Selector-View erbt davon die Klasse FLGSelectorElementsStructurePanel: Diese stellt die Basisfunktionalität zum Verwalten der durch den Nutzer angelegten Spielthemen zur Verfügung.

¹ Vgl. <http://openuss.svn.sourceforge.net/viewvc/openuss/trunk/fsl/src/viewManagerAPI/>.

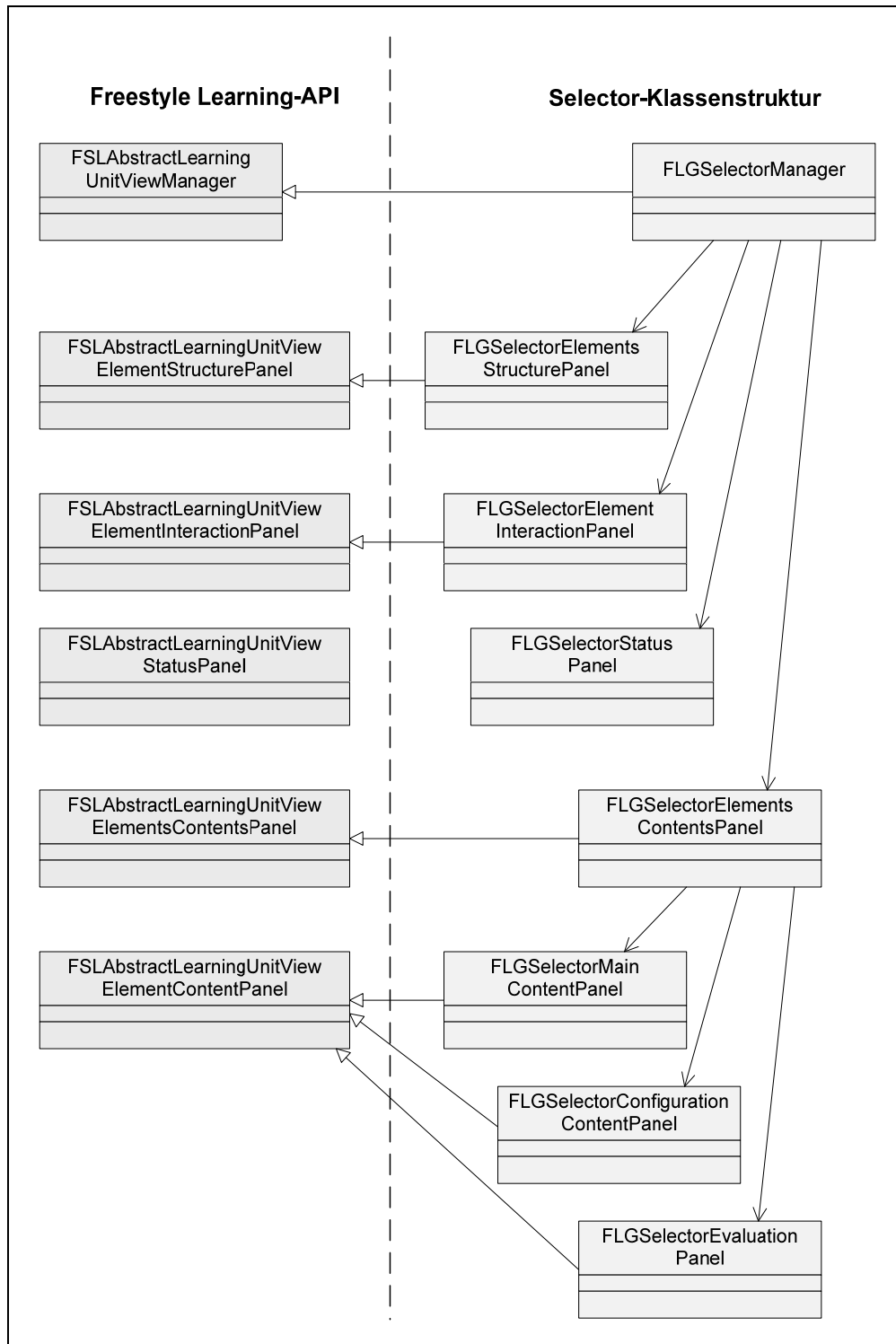


Abb. 6: Klassendiagramm der Selector-View-Integration

FSLAbstractLearningUnitViewElementInteractionPanel

Die Klasse `FSLAbstractLearningUnitViewElementInteractionPanel` bildet die Basisklasse für das Verwalten der Interaktionsleiste, z. B. das Hinzufügen, Aktivieren und Deaktivieren von Schaltflächen. Die Selector-View-Klasse `FLGSelectorElementInteractionPanel` erweitert die-

se Klasse, indem sie ihre spezifischen „Start“- und „Stop“-Schaltflächen integriert und die damit verbundenen Methodenaufrufe verwaltet.

FSLAbstractLearningUnitViewStatusPanel

Die Struktur des Statusfeldes der Freestyle Learning-Oberfläche wird durch die Klasse `FSLAbstractLearningUnitViewStatusPanel` vordefiniert. Innerhalb des `Selector-View-Packages` erbt davon die Klasse `FLGSelectorStatusPanel`. Diese Klasse dient zur Anzeige der Namen der aus dem Structure Panel gewählten Elemente sowie zur Fortschrittsanzeige innerhalb des Selector-Durchlaufs.

FSLAbstractLearningUnitViewElementsContentsPanel

Innerhalb der Klasse `FSLAbstractLearningUnitViewElementsContentsPanel` werden die verschiedenen Sichten des Anzeigebereichs verwaltet. In der `Selector-View-Implementierung` erbt davon die Klasse `FLGSelectorElementsContentsPanel`. Diese initialisiert das Content Panel für die Konfiguration, das Content Panel für den eigentlichen Spielablauf sowie das Content Panel für die Evaluation der Ergebnisse.

FSLAbstractLearningUnitViewContentPanel

Die Klasse `FSLAbstractLearningUnitViewContentPanel` bildet den konkreten Aufbau einer Sicht im Anzeigebereich ab. In der `Selector-View` werden drei verschiedene Content Panel-Typen benötigt. Diese werden durch die von der Klasse `FSLAbstractLearningUnitViewContentPanel` ererbenden Klassen `FLGSelectorElementMainConfigurationPanel` für die Selector-Konfiguration, `FLGSelectorElementMainContentPanel` für die Selector-Spielfläche und die Klasse `FLGSelectorEvaluationContentPanel` initialisiert und verwaltet.

Über die obigen Basiselemente hinaus wurden Klassen für Dialogabfragen und Hilfsklassen für die Verwaltung der Learning Unit-View-Datenhaltung implementiert.

Die persistente Datenhaltung der Freestyle Learning-Plattform erfolgt über XML-Deskriptoren. Um einmal erstellte Selector-Spielkonfigurationen persistent zu halten, werden diese ebenfalls in XML-Dateien gespeichert. Dazu wurden die bestehenden XML-Deskriptoren um den Datentypen `GridObject` erweitert. Ein `GridObject` beschreibt ein Element des Selector-Spielfeldes, das entweder aus einem Texteintrag oder einem Verweis auf eine Bilddatei besteht. Zusätzlich wird bei einem `GridObject` vermerkt, ob dieses für die Auswahl zulässig ist. Für jedes Thema werden die `GridObjects` in den entsprechenden Themen-Deskriptor eingefügt. Alle Themenelemente (im FSL-Termini auch View-Elemente genannt) bilden die vollständige Konfiguration einer Learning Unit-View.

```
<!ELEMENT GridObjekt EMPTY>
<!ATTLIST GridObjekt id NMTOKEN #REQUIRED>
<!ATTLIST GridObjekt type CDATA #REQUIRED>
<!ATTLIST GridObjekt text CDATA #IMPLIED>
<!ATTLIST GridObjekt imageFileName CDATA #IMPLIED>
<!ATTLIST GridObjekt clickAllowed (true|false)REQUIRED>
```

Abb. 7: XML-Deskriptor-Struktur eines GridObject

Der in Abb. 7 dargestellte XML-Deskriptor zeigt den grundsätzlichen Aufbau eines **GridObject**. Ein **GridObject** besitzt einen Identifier (*id*) zur eindeutigen Adressierung des Elementes und das Attribut „*type*“ zur Unterscheidung zwischen Text- und Grafikelementen. Beide Attribute müssen zur eindeutigen Zuordnung gesetzt werden. Dies wird durch das Schlüsselwort „**REQUIRED**“ angegeben. Bei einem Textelement wird der entsprechende Textausdruck als Attribut und bei einem Grafikelement der Verweis auf die Grafikdatei gespeichert. Darüber hinaus muss vermerkt werden, ob ein **GridObject** zulässig oder unzulässig ist. Dies wird durch den Datentyp **boolean** in Form des Attributes „*clickAllowed*“ angegeben. Der GridObject-Deskriptor selbst wird wiederum in ein View-Element-XML-Tag integriert. Ein View-Element-XML-Tag bildet die XML-Struktur für ein Themenelement ab. Die Gesamtheit aller View-Element-Tags bildet den vollständigen Selector-XML-Deskriptor einer Selector Learning Unit-View.

```
<selectorDescriptor>
<viewElement id="e1"
  parentId="none"
  title="Investitionsrechnung"
  folder="true"/>
<viewElement id="2"
  parentId="e1"
  title="VOFI-Spiel"
  folder="false"
  type="text"
  musicFileName="song1.wav">
  <gridObject id="Kalulationszinsfuss"
    type="text"
    text="Kalulationszinsfuss"
    isClickAllowed="true">
  <gridObject id="vofi"
    type="image"
    imageFileName="vofi.gif";
    isClickAllowed="true"/>
  <gridObject id="HGB"
    type="text"
    text="HGB"
    isClickAllowed="false"/>
</viewElement>
</selectorDescriptor>
```

Abb. 8: Exemplarischer Selector-Descriptor

Der XML-Descriptor aus Abb. 8 beschreibt den Aufbau einer installierten Selector-Learning Unit-View. Diese besteht aus einem Ordner mit dem Titel „Investitionsrechnung“ und einem Unterelement in Form eines Spielthemas mit dem Titel „VOFI-Spiel“ mit der eingestellten Musik aus der Datei „song1.wav“. Das „VOFI-Spiel“ besteht in diesem Fall aus zwei gültigen Elementen, einem Textelement „Kalkulationszinsfuß“ und einem Bildelement, das in der Datei „vofi.gif“ abgespeichert ist. Ferner ist ein nicht zulässiges Textelement angelegt worden, das den Begriff „HGB“ beinhaltet.

Arbeitsberichte der Reihe „CAL+CAT“

- 1 Computergestützte Lehre an der Universität
- 2 CAL + CAT
- 3 Multimedia
- 4 Hypertext
- 5 Entwicklung von CAL-Software
- 6 Erwartungen der Praxis an CAL
- 7 Evaluation der multimedialen Lehre in der Leistungs- und Kostenrechnung
- 8 Internet-Unterstützung der Lehre
- 9 Entwicklung und Nutzung multimedialer Multiple-Choice-Software
- 10 cCAL – Cooperative Computer Assisted Learning
- 11 Das Internet im Mittelpunkt einer computergestützten Hochschullehre (cHL)
- 12 Dynamische Web-Präsenzen
- 13 CAL-Software für die Volkswirtschaftslehre
- 14 Das cHL-Redaktionssystem InterUSS
- 15 Kinetische Grafiken
- 16 Weblog-Mining bei cHL-Anwendungen
- 17 Selector (der Inhalt des Arbeitsberichtes ist aufgrund technischer Neuerungen veraltet und daher aus dem Programm genommen worden)
- 18 Das Granulatkonzept – Entwicklung eines Lernsystems für die computergestützte Hochschullehre (wurde in den 20 integriert und daher aus dem Programm genommen)
- 19 Internationale Kooperation in der universitären Ausbildung
- 20 Freestyle Learning – Das mediendidaktische Konzept
- 21 Freestyle Learning – Plattform
- 22 Das Konzept der computergestützten Hochschullehre
- 23 Das Konzept der computergestützten Hochschullehre Wirkungen, Chancen und Risiken – Ergebnisse einer Expertenbefragung (Teil I)
- 24 Strategische Potenziale von Open Source – Software für die computergestützte Hochschullehre (cHL)
- 25 Informationsverarbeitung in der Hochschullehre
- 26 Die Zukunft der computergestützten Hochschullehre Prognosen und Szenarien – Ergebnisse einer Expertenbefragung (Teil II)
- 27 Wissensnetzwerk Controlling – Evaluationsergebnis der didaktisch-pädagogischen Begleitgruppe
- 28 Ein Open Source-basiertes Learning-Managementsystem für Volksbanken
- 29 Erweiterung von Freestyle Learning zur Administration von Online-Content