



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fakultät für Elektrotechnik, Informatik und Mathematik
Institut für Informatik
Fachgebiet Didaktik der Informatik

Re-Engineering eines Werkzeugs zur Videoannotation und -bearbeitung

Schriftliche Hausarbeit

vorgelegt im Rahmen der Ersten Staatsprüfung für das Lehramt für
Gymnasien und Gesamtschulen
in Informatik

vorgelegt von:
Christiane Beller

Dringenberg, 9. Oktober 2008

Beilage: CD

Gutachter

Prof. Dr. Johann S. Magenheimer
Universität Paderborn
Fakultät EIM, Didaktik der Informatik

Zusammenfassung

Im Alltag beobachtet man Situationen nur aus einer Sicht: seiner eigenen. Für eine bessere Einschätzung und Bewertung ist es allerdings oft hilfreich, die gleiche Situation aus einer anderen Perspektive zu betrachten.

Zur Evaluation von Unterricht ist ein Wechsel der Perspektive hilfreich, um mögliche Schwächen des Unterrichts festzustellen. Für die Unterrichtsevaluation wurde bereits ein Markierungs- und Schnittwerkzeug entwickelt, das im Rahmen der Lehramtsausbildung im Bereich Informatik zum Einsatz kommt. Dieses ist auf die notwendigen zwei Sichten beschränkt: Lehrer- und Schülersicht.

Um das Einsatzgebiet dieses Programms zu erweitern, findet mit dieser Staatsexamensarbeit ein Re-Engineering des Werkzeugs statt. Zu Beginn werden das bestehende Programm sowie die veränderten Anforderungen vorgestellt. Anschließend wird die überarbeitete Version präsentiert. Zudem werden unterschiedliche Szenarien aufgezeigt, in denen das Programm sinnvoll eingesetzt werden kann.

Inhaltsverzeichnis

1	Einleitung	1
2	Das Tool „ViLM“ – bisheriger Stand	3
2.1	Allgemeine Beschreibung	3
2.2	„ViLM – Synchronisation“	4
2.3	„ViLM – Annotate“	5
2.4	Neue Anforderungen	10
3	„ViLM“ – Anpassung und Überarbeitung	13
3.1	Design-Entscheidung	13
3.2	„ViLM – VideoVorbereitung“	14
3.3	„ViLM – Annotate“	16
3.4	Architektur	18
4	Mögliche Einsatzszenarien	19
4.1	Der Vortrag – Ein-Kamera-Perspektive	19
4.2	Die Filmanalyse – Ein-Kamera-Perspektive	20
4.3	Das Bewerbungsgespräch – Zwei-Kamera-Perspektive	22
4.4	Die Unterrichtsanalyse – Zwei-Kamera-Perspektive	23
4.5	Die Gruppenarbeit – Vier-Kamera-Perspektive	24
4.6	Die Pyramidendiskussion – <i>n</i> -Kamera-Perspektive	25
5	Anwendungsfälle	27
5.1	Rhetorik-Schulung – Ein-Kamera-Perspektive	27
5.2	Die Unterrichtsanalyse – Zwei-Kamera-Perspektive	28
5.3	Gruppenarbeit zum Thema „Mauerfall“ – Vier-Kamera-Perspektive	29
5.4	Pyramidendiskussion zur Staatsgründung – <i>n</i> -Kamera-Perspektive	30
6	Verwendete Technologien und Konzepte	33
6.1	Objektorientierung	33
6.2	Die Programmiersprache C#	33
6.3	XML - Extensible Markup Language	33
6.4	MDI-Anwendung	36
6.5	ffmpeg	37
7	Resümee	39

7.1	Aufgetretene Schwierigkeiten	39
7.2	Ausblick	40
	Abbildungsverzeichnis	III
	Listings	V
	Literaturverzeichnis	VII
	A Muster einer Zustimmungserklärung	IX

1 Einleitung

Im Lehramtsstudium Informatik an der Universität Paderborn beinhaltet die Veranstaltung „Methoden des Informatikunterrichts in Theorie und Praxis“, kurz MIU, die theoretischen Grundlagen und Planungen von Unterricht, sowie die praktische Umsetzung von selbst geplantem Unterricht. Im theoretischen Teil des Seminars werden Methoden und Arbeitsformen für Unterricht besprochen und Möglichkeiten der Unterrichtsevaluation präsentiert. Ferner ist die Planung von eigenen Unterrichtseinheiten bzw. -stunden vorgesehen. Da dies die ersten Gehversuche in der Rolle als Informatiklehrerin / Informatiklehrer¹ sind, ist nicht nur die gute Planung von Unterricht entscheidend, sondern auch eine genaue Analyse und Reflexion des jeweiligen Unterrichts. Die eigene Sicht auf den abgehaltenen Unterricht ist sehr eingeschränkt und viele Reaktionen von Schülern bleiben unbemerkt oder geraten in Vergessenheit. Um eine möglichst genaue Untersuchung und Auswertung des Unterrichts durchführen zu können, wird in dieser Veranstaltung das Programm „ViLM“ eingesetzt. Der Name leitet sich von dem Projekt „Visualization of Learning and Teaching Strategies with Multimedia in Teacher Education“ ab, welches sich mit dem Einsatz von Multimedia und Hypermedia zur Unterstützung von Unterricht befasst (siehe hierzu [Mag99]). Es wurde von Oliver Buschjost in der aktuellen Version erstellt².

Mit der bisherigen Version von „ViLM“ lassen sich Situationen aus zwei Perspektiven darstellen. Der Einsatz während eines Bewerbungsgesprächs, einer Schulung oder Ähnlichem ist denkbar, da vom Programm keine Unterrichtsstruktur vorgegeben wird. Einen Bezug zu Unterricht wird derzeit in der vilml.xml-Datei hergestellt. Hier werden die Videos mit „teacherView“ und „pupilView“ benannt. Es gibt aber viele Situationen, in denen eine oder auch mehr als zwei Sichten wünschenswert wären: Für die Analyse von Videos benötigt man nur eine Sicht, für die genaue Analyse eines Bewerbungsgesprächs sind zwei Sichten sinnvoll, für Gruppenarbeiten ist die Anzahl von der Gruppenstärke abhängig. Bedingt durch das verfolgte Ziel kann also eine Bearbeitung von verschiedenen Anzahlen von Videos notwendig sein. Um die gewünschten Einsatzgebiete nachvollziehbar zu dokumentieren, werden in Kapitel 4 Szenarien vorgestellt, welche

¹Im nachfolgenden Teil dieser Arbeit werde ich nur noch die männliche Form verwenden, um den Text übersichtlicher zu gestalten. Natürlich kann die jeweilige Person weiblich als auch männlich sein.

²siehe hierzu auch in der Studienarbeit: Markierungs- und Schnittwerkzeug zur videounterstützten Analyse von Kommunikationsszenarien - Machbarkeitsstudie, Konzeption und prototypische Umsetzung, von Oliver Buschjost, <http://ddi.uni-paderborn.de/studienarbeiten/oliver-buschjost.html> (Stand: September 2008)

die Verwendung von unterschiedlichen Kameraanzahlen beispielhaft aufzeigen. Ferner ist der Wunsch nach einem plattformunabhängigen Player aufgekommen, der lokal installiert werden kann. Dieser soll die Betrachtung der Videos samt Annotationen ermöglichen. Der Player verwendet eine bestehende Annotationsdatei, welche entsprechende Informationen über die Videos, gegebene Kommentare und Ähnliches enthält. Die Videos werden von einem Server als Stream wiedergegeben und müssen somit nicht lokal vorliegen. Aufgrund der für den Player verwendeten Technologie ist eine Umwandlung der Videos in FLV-Dateien erforderlich. Dies bedeutet auch, dass die bisherige, aufwändige Umwandlung der Videos in MP4 entfällt.

Durch die Ausweitung des Anwendungsbereichs auf ein bis beliebig viele Videos und eine plattformunabhängigen Player ist das Re-Engineering des Werkzeugs zur Videoannotation und -bearbeitung unerlässlich. In dieser Arbeit wird die Änderung des Synchronisations- und Annotationswerkzeugs beschrieben. Die Realisierung des Players wird in der Examensarbeit von André Hoffmann bearbeitet und genauer erläutert.

2 Das Tool „ViLM“ – bisheriger Stand

In diesem Abschnitt werden die bisherigen Programme von „ViLM“, den Synchronisations- und den Annotationsteil, genauer erläutert. Diese beiden Bereiche sind die wesentlichen Inhalte dieser Arbeit. Mit dieser Darstellung soll verdeutlicht werden, inwieweit der Einsatz des Programms beschränkt ist. Ferner sollen bisherige Schwachstellen aufgezeigt werden. Da die Verwendung bisher hauptsächlich im Umfeld der Unterrichtsanalyse und -reflexion stattfindet, wird die Beschreibung häufig hierauf Bezug nehmen.

Für die Benutzung der „ViLM“-Software ist ein Windows-System mit installiertem .NET-Framework erforderlich. In manchen Fällen kann die Installation eines entsprechenden Videocodecs erforderlich sein, da sonst die Videos nicht angezeigt werden können. Die „ViLM“-Software muss nicht installiert werden, so dass direkt mit der Bearbeitung begonnen werden kann.

2.1 Allgemeine Beschreibung

Um „ViLM“ verwenden zu können, werden zwei Videos der gleichen Situation, jedoch aus unterschiedlichen Perspektiven, benötigt. Um die verschiedenen Perspektiven widerzuspiegeln, werden während der Unterrichtsdurchführung zwei Kameras eingesetzt. Die eine Kamera zeigt dabei die Sicht des Lehrers, die andere repräsentiert die Schülersicht. Beide Bildbereiche müssen sich überschneiden, da die Videos später synchron abgespielt werden sollen.

Nach dem Unterricht werden die beiden Videos dann mittels des Synchronisationsprogramms von „ViLM“, das lokal auf einem Windows-System gespeichert sein muss, auf einen Zeitpunkt synchronisiert. Die Videos werden hierzu im angegebenen Projektordner im Unterverzeichnis „Video“ abgelegt. Anschließend wird das Annotationsprogramm geöffnet. Ein neu angelegtes Projekt muss in einem Verzeichnis erzeugt werden, in dem sich die zuvor erzeugte Synchronisationsdatei befindet. Aus dieser Datei werden notwendige Informationen über die Videos ausgelesen. In „ViLM – Annotate“ kann in einem dafür vorgesehenen Bereich eine Einteilung der Videos in Phasen, die sich z.B. an den geplanten Unterricht anlehnen, vorgenommen werden. Nach der Einteilung kann jeweils eine beliebige Beschreibung der Phase vorgenommen, ein entsprechender Kommentar abgegeben und verwendetes oder entstandenes Material hinzugefügt werden (genauer: siehe Kapitel 2.3).

2.2 „ViLM – Synchronisation“

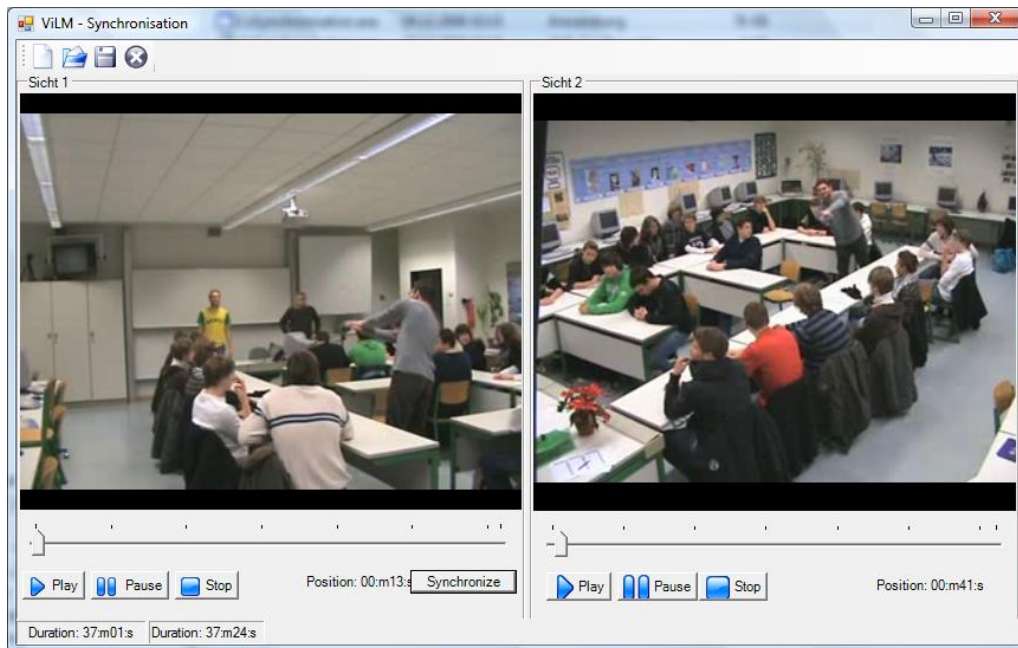


Bild 2.1: Aktuelle „ViLM – Synchronisation“ Oberfläche

Zur Synchronisation muss „CsSynchronisation.exe“ ausgeführt werden. Die bisherige Version von „ViLM – Synchronisation“ bietet eine Oberfläche, die in zwei wesentliche Teile gegliedert ist: Zwei nebeneinander angeordnete Videos mit entsprechender Navigation (Bild 2.1). In der Mitte zwischen den Bereichen besteht die Möglichkeit, eines der Videos vergrößert und das andere entsprechend verkleinert darzustellen. Allerdings sind die Bereiche zur Steuerung der Videos fest vorgegeben, so dass bei der Größenveränderung die Steuerung eines der beiden Videos fast vollständig verschwindet.

Nachdem ein neues Projekt in einem ausgewählten Verzeichnis erstellt wurde, bieten beide Bereiche jeweils zwei Buttons an, um Videos zuzuweisen. Zum einen können Dateien aus dem Projektverzeichnis geöffnet, zum anderen Dateien importiert werden. Durch das Importieren werden die ausgewählten Videos in dem Projektverzeichnis im Ordner „Video“ abgelegt. Falls dieser Ordner nicht existiert, wird er automatisch erzeugt. Existiert das ausgewählte Video in dem Videoverzeichnis bereits, wird zwar ein Fenster mit Kopiervorgang angezeigt, das Video wird aber nicht überschrieben. Dies kann viel Zeit in Anspruch nehmen, falls große Videodateien verwendet werden, ohne dass die eigentlichen Projektdateien beeinflusst werden. Durch das Öffnen eines bestehenden Projektes werden die beiden Videos direkt in die entsprechenden Bereiche geladen.

Soll das Video aus dem Projektverzeichnis geöffnet werden, aber der Benutzer wählt ein anderes Verzeichnis aus, treten später bei der Annotation der Videos Probleme auf. „ViLM – Annotate“ versucht das angegebene Video aus

dem Video-Ordner des Projektverzeichnisses zu öffnen und nicht von der zuvor ausgewählten Quelle.

Das Importieren eines Videos in Sicht 2 (rechts), das sich allerdings bereits im Projektverzeichnis befindet, verursacht die Meldung, dass das „Hinzufügen fehlgeschlagen“ sei, ohne dem Benutzer eine genaue Fehlerbeschreibung zu geben. Dasselbe Vorgehen im Bereich von Sicht 1 (links) läuft ohne Fehler ab. Werden die Videos importiert, so werden die Buttons zum Öffnen aus dem Projektverzeichnis weiterhin angezeigt, auch wenn die Videos gestartet werden. Erst nachdem man diesen Button gedrückt und ein beliebiges, evtl. anderes Video ausgewählt hat, verschwindet dieser aus der Ansicht. Angezeigt wird das Video, das zu Beginn ausgewählt wurde.

Im unteren Fensterbereich, der sogenannten Statusbar, wird in zwei Feldern „Duration“ mit einer Zeitangabe angezeigt. Diese zeigen die Gesamtlaufzeit der jeweiligen Videos. Ferner befindet sich in jedem der Bereiche eine Positionsanzeige („Position“ gefolgt von einer Zeitangabe), welche die aktuelle Laufzeit anzeigt.

Durch den SYNCHRONIZE-Button in der linken Ansicht wird die Synchronisation der beiden Videos durchgeführt, auch wenn dieser optisch einem der beiden zugewiesen ist. Leider kann man auch ohne die Synchronisation durchgeführt zu haben, den PROJEKT SPEICHERN-Button benutzen. Dies verursacht einen für den Benutzer nicht klar einzuordnenden Fehler, welcher ebenfalls auftritt, wenn der Benutzer den Button zum Speichern zwei Mal hintereinander betätigt. Da es keine Rückmeldung der erfolgreichen Speicherung gibt, kann dieser Fall häufiger eintreten, so dass eine plausible Fehlermeldung oder eine Rückmeldung, dass das Speichern erfolgreich durchgeführt wurde, sinnvoll ist.

Für die Synchronisation gibt es einen Button namens PROJEKT SCHLIESSEN. Allerdings zeigt das Programm keine Reaktion bei der Benutzung des Buttons, egal in welcher Situation man ihn betätigt. Falls der Benutzer ein neues oder bestehendes Projekt öffnet, ohne das aktuelle zu speichern, gehen die aktuellen Informationen ohne Warnung verloren. Geschlossen werden kann das Programm nur über den von Windows-Anwendungen standardmäßig benutzten SCHLIESSEN-Button („X“) oben rechts am Fensterrand. Aber auch beim Schließen des Fensters wird nicht auf die Speicherung des Projekts verwiesen.

2.3 „ViLM – Annotate“

Das Programm zur Annotation von Projekten wird über „Annotate.exe“ gestartet. Die Einteilung der Oberfläche von „ViLM – Annotate“ ist bereits nach dem Starten des Programms sichtbar. Sie ist in zwei Hauptbereiche aufgeteilt, die horizontal angeordnet sind. Im oberen Bereich befinden sich die Navigationsleiste, sowie die nebeneinander angeordneten Videos, ähnlich wie bei „ViLM –

Synchronisation“. Im unteren Fensterabschnitt befinden sich die Videosteuerung sowie der Bereich für die Annotationen.

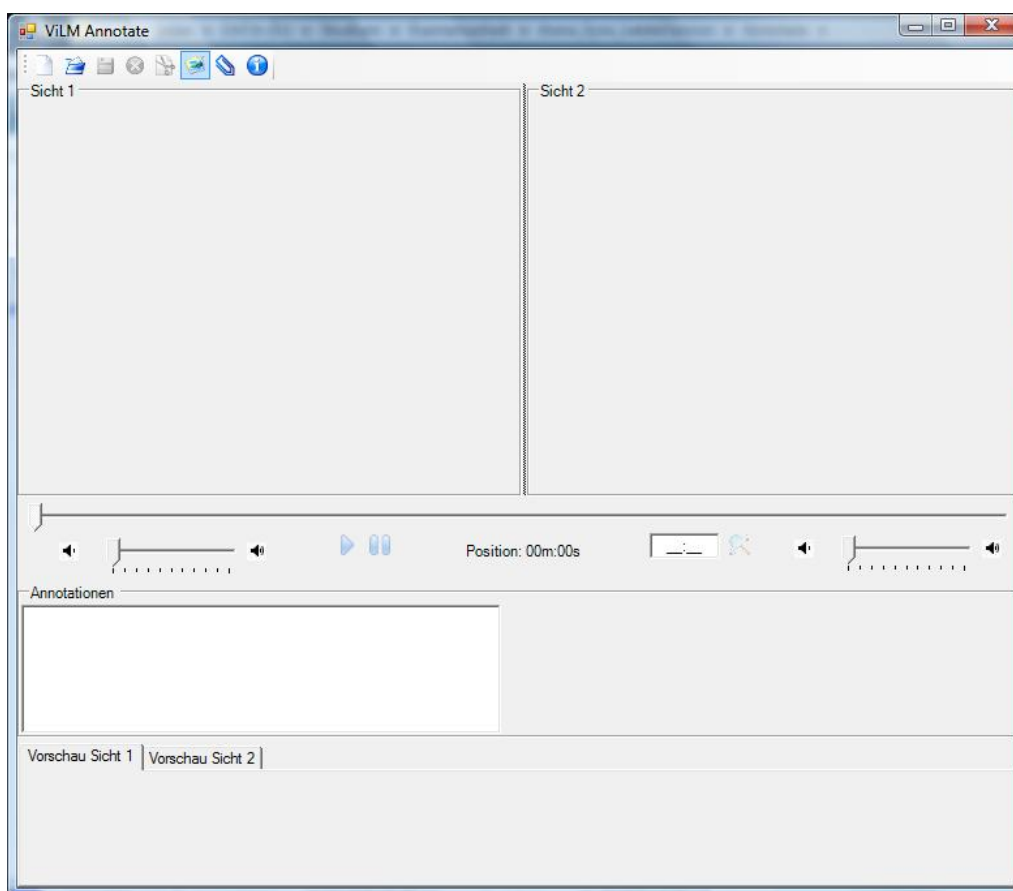


Bild 2.1: Aktuelle „ViLM – Annotate“ Oberfläche nach dem Programmstart

Auch bei „ViLM – Annotate“ lassen sich die Größen der Bereiche verändern. Die Änderung der Videogröße ist entsprechend der bei „ViLM – Synchronisation“. Wird der waagerechte Balken nach unten verschoben, verschwindet erst der Bereich für die Steuerung der Videos und schließlich der Bereich für die Annotationen.

In der Navigationsleiste stehen acht Funktionen zur Auswahl. NEUES PROJEKT ERSTELLEN und PROJEKT ÖFFNEN sowie PROJEKT SPEICHERN und PROJEKT SCHLIESSEN sollten ohne weitere Erläuterungen jedem Benutzer verständlich sein. Die übrigen Buttons werden im nachfolgenden Text erläutert.

Für die Verwendung des Programms ist aber wichtig, dass sich, unabhängig davon, ob man ein neues Projekt anlegen oder ein bestehendes öffnen möchte, in dem ausgewählten Projektordner die von „ViLM – Synchronisation“ erzeugte „vilm.xml“ befindet. Sie enthält die für den Start der Videos notwendigen Informationen. Falls diese Datei nicht existiert, können zwar Notizen gemacht werden, allerdings gibt es hierzu keine entsprechenden Videos. Wird ein solches Projekt beendet, wird eine Fehlermeldung („Unbehandelte Ausnahme in der An-

wendung...“) ausgegeben. Diese beschreibt leider nicht die eigentliche Ursache des Fehlers.

An manchen Stellen zeigt das Programm ungewolltes oder nicht intuitiv verständliches Verhalten, das nun näher beschrieben wird. Falls bereits ein Projekt geöffnet ist und der Benutzer ein anderes Projekt öffnen möchte, wird eine Abfrage zum Speichern des geöffneten Projekts gestartet. Wird statt ein anderes Projekt zu Öffnen aber ein neues Projekt erstellt, erfolgt keine Nachfrage zum Speichern. Dieses inkonsistente Verhalten des Programms ist für den Benutzer ungünstig, falls er bereits Text geschrieben hat und nur durch einen (evtl. auch ungewollten) Klick des Buttons NEUES PROJEKT ERSTELLEN seine gesamte Arbeit löscht.

Falls der Benutzer sein Projekt gespeichert hat und das Projekt schließen möchte, wird er in jedem Fall gefragt, ob er sein Projekt speichern möchte. Diese Reaktion tritt unabhängig davon auf, ob PROJEKT SCHLIESSEN oder die Windows-Standard-Möglichkeit über den SCHLIESSEN-Button („X“) oben rechts gewählt wurde. Dies kann zur Verunsicherung des Benutzers führen, dahingehend dass Änderungen vorgenommen werden können, von denen er nichts bemerkt.

Nach dem Programmstart ist ein Bereich auffällig, der sich unterhalb der Annotationen befindet. Dies ist ein Bereich mit zwei Registerkarten, die mit „Vorschau Sicht 1“ und „Vorschau Sicht 2“ benannt sind. Dieser Bereich ist nur nach dem Programmstart sichtbar und kann, falls kein Projekt geöffnet wurde, mittels THUMBNAIL ANZEIGE ein- bzw. ausgeblendet werden. Dieser Button ist auch während eines geladenen Projekts sichtbar, allerdings kann er nicht verwendet werden.

Des Weiteren gibt es einen Button ANNOTATIONEN SCHNEIDEN. Die Wirkung dieses Buttons ist nicht klar. Scheinbar stellt dieser eine Verbindung zu einem vorgegebenen Server her, um die Umcodierung der Videos in MP4 vorzunehmen. Leider erhält man bei Ausführung dieser Aktion eine Fehlermeldung, die keinen direkten Rückschluss auf die Fehlerquelle zulässt.

Ein mit einer Büroklammer gekennzeichneteter Button namens TOOLSTRIPBUTTON4 öffnet ein Fenster mit der Überschrift „Material-Verwaltung“ (siehe Bild 2.2). Welche Funktion dieses Fenster mit seinen vier Button übernehmen soll, ist nicht beschrieben und kann auch aus dem Kontext nicht erschlossen werden. Der INFORMATIONEN-Button öffnet ein kleines Fenster, in dem der Programmierer bzw. Bearbeiter der Software angezeigt wird.

Der Bereich direkt unterhalb der Videos dient zu ihrer Steuerung. Zum großen Teil ist die Funktionalität intuitiv verständlich. Die Buttons zum Abspielen und Pausieren haben das gleiche Layout, welches von vielen elektronischen Geräten wie DVD- oder MP3-Playern und diversen Video- und Musikprogrammen genutzt wird. In dem Bereich hinter *Position* wird die aktuelle Spieldauer der Videos angezeigt, analog zur Anzeige in „ViLM – Synchronisation“. Der *Slider* („Schie-

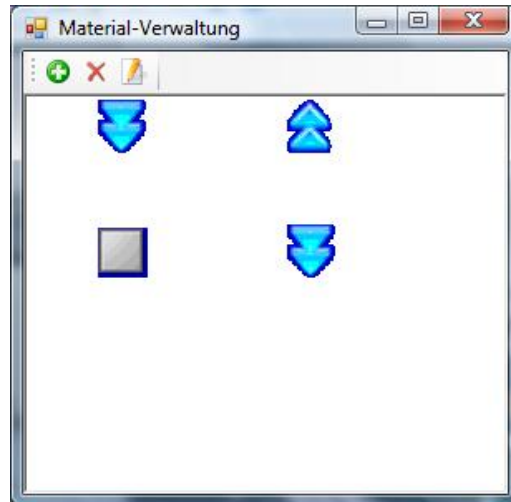


Bild 2.2: Fenster zur Material-Verwaltung

ber“), der sich über die gesamte Fensterbreite erstreckt, dient zum Einstellen der (ungefähren) Position innerhalb der Videos. Jeweils unter den Videos befinden sich zwei kleine *Slider*, über die die Regelung der Lautstärke vorgenommen werden kann. Klickt man auf die Symbole direkt vor bzw. hinter diesen *Slidern*, wird der Ton aus- bzw. auf volle Lautstärke eingestellt. Falls der Benutzer die Lautstärke von beiden Videos auf einen hörbaren Bereich einstellt, die Videos aber nicht exakt synchron ablaufen, wird der Ton von beiden Videos vermischt.

Nun bleibt in diesem Bereich noch ein Eingabefeld und ein Button mit der Beschreibung EINEN FRAME WEITER. Die Eingabe bleibt auf insgesamt vier Ziffern beschränkt, jeweils zwei Ziffern, die durch einen Doppelpunkt getrennt sind. Die Größe der angegebenen Zahlen ist dann aber beliebig. Beim Klicken auf EINEN FRAME WEITER wird nur ein Fenster angezeigt, das „Work in progress: “ gefolgt von der angegebenen Ziffernfolge (ohne Doppelpunkt) ausgibt. So wie es scheint, wurde bisher die vorgesehene Funktionalität nur im Ansatz umgesetzt und bietet dem Benutzer keine Funktion an.

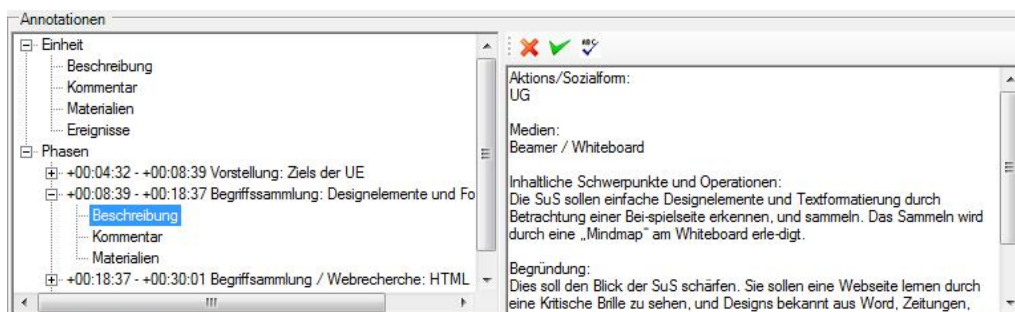


Bild 2.3: Bereich für Annotationen

Der Bereich für die Annotationen (Bild 2.3) ist in zwei Teile aufgeteilt: in dem linken Teil ist eine Baumstruktur zu finden. Im rechten Teil sind Texteingaben möglich. Zu Beginn der Annotation besteht der Baum aus einem Knoten *Einheit*

und einem Knoten *Phasen*. Dem Einheiten-Knoten sind folgende Knoten untergeordnet: *Beschreibung*, *Kommentar*, *Materialien* und *Ereignisse*. Unter *Phasen* ist vorerst kein Eintrag zu finden. Bei Klick auf *Beschreibung* oder *Kommentar* wird rechts ein Menü mit ÄNDERUNGEN VERWERFEN, ÄNDERUNGEN ÜBERNEHMEN sowie SPELL (zur Überprüfung der Rechtschreibung) und darunter ein Feld zur Texteingabe angezeigt. Bei der Auswahl von *Materialien* ist der Aufbau ähnlich zu dem vorher Beschriebenen allerdings mit folgender Funktionalität: MATERIAL EINFÜGEN, MATERIAL LÖSCHEN und MATERIAL BEARBEITEN. Statt des Textfeldes ist ein Bereich zur Anzeige der hinzugefügten Materialien zu sehen. An dieser Stelle können beliebige Dateien angehängt werden, die während der Unterrichtseinheit eingesetzt wurden oder entstanden sind. Wird der Button zur Bearbeitung oder Löschung benutzt, ohne zuvor ein Material ausgewählt zu haben, erscheint eine Fehlermeldung, welche für den Benutzer nicht aussagekräftig ist („Unbehandelte Ausnahme in der Anwendung...“). An dieser Stelle sollte der Benutzer explizit darauf hingewiesen werden, was er zu tun hat, falls er die Funktion benutzen möchte. Bei der Auswahl einer aufgeführten Datei unter *Materialien* kann mittels MATERIAL BEARBEITEN die Datei mit dem Standardprogramm, welches mit Dateien dieses Formats verknüpft ist, geöffnet werden. Durch einen Rechtsklick auf den Knoten *Materialien* wird ein Kontextmenü mit zwei Punkten aufgerufen. Dort sind MATERIAL HINZUFÜGEN, sowie SEQUENZ HINZUFÜGEN zu finden. Material kann also auch über diesen Weg hinzugefügt werden. Die Funktionalität um Sequenzen hinzuzufügen ist allerdings nicht umgesetzt oder beschrieben worden.

Führt man auf *Ereignisse* einen Rechtsklick aus, kann man ein Ereignis mit der aktuellen Zeit innerhalb des Videos einfügen. Dieses wird als Kindknoten zu *Ereignisse* erzeugt. Bei der Auswahl des Ereignisses ist das angezeigte Aussehen im rechten Bereich fast identisch mit dem zu *Beschreibung* oder *Kommentar* mit dem Unterschied, dass hier keine Überprüfung der Rechtschreibung vorgenommen werden kann. Das eingefügte Ereignis wird standardmäßig mit „Stunden:Minuten:Sekunden: Event“ eingefügt. Klickt man nun mit der rechten Maustaste auf diese Benennung, kann man Folgendes auswählen: EREIGNIS UMBENENNEN, EREIGNIS LÖSCHEN oder EREIGNIS ANSPRINGEN. Was sich hinter diesen Funktionen verbirgt, ist selbsterklärend. Leider erfolgt derzeit keine Speicherung der Events.

Unter *Phasen* ist vorerst kein Eintrag zu finden. Ein Rechtsklick auf *Phasen* bietet (nur) die Möglichkeit, eine NEUE PHASE zu erzeugen. Die Benennung des Knotens ist standardmäßig „Startzeit - Endzeit Phase“ gefolgt von einer Zahl. Die Zeiten sind so formatiert, wie bei den Events bereits angegeben. Als Startzeit wird der Zeitpunkt gesetzt, der im Positionsfeld angezeigt wird. Die Endzeit ist immer die Endzeit der Videos. Über das Kontextmenü können diese Zeiten angepasst oder auch angesprungen werden. Weitere Funktionen, die angeboten werden sind NEUE PHASE, UMBENENNEN, LÖSCHEN sowie PHASE NACH OBEN

BEWEGEN und PHASE NACH UNTEN BEWEGEN. Wird an dieser Stelle eine neue Phase erzeugt, wird sie als Unterphase zur aktuellen Phase angelegt. Das Aussehen der Unterphasen ist analog zu den beschriebenen Phasen. Allerdings werden diese Phasen derzeit nicht gespeichert.

Wählt man eine spezielle Phase aus, werden die Unterknoten *Beschreibung*, *Kommentar* und *Materialien* angezeigt. Ihr Aufbau ist analog zu den entsprechenden Knoten unter *Einheit*.

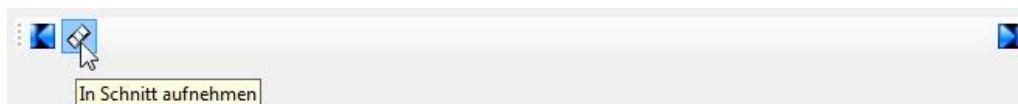


Bild 2.4: Anzeige bei Auswahl einer Phase

Wird nicht *Beschreibung*, *Kommentar* oder *Materialien* zu einer Phase explizit ausgewählt, sondern die Bezeichnung der Phase (mit Uhrzeit) angeklickt, werden im rechten Bereich drei Buttons angezeigt, deren Funktionalität aber nicht direkt klar ist (siehe Bild 2.4). Wenn man den Mauszeiger darüber platziert, erscheint ein Tooltipp, der eine kurze Benennung der Funktionalität angibt: START MARKIEREN, IN SCHNITT AUFNEHMEN und ENDE MARKIEREN. Ohne weitere Erklärungen bleibt aber der Sinn des Buttons IN SCHNITT AUFNEHMEN unklar. Die beiden anderen Buttons sind auch ohne Erläuterungen verständlich.

Nach Beenden der Annotation ist die Annotationsdatei im zu Beginn gewählten Projektordner abzuspeichern. Dies ist erforderlich, damit bei erneutem Öffnen auch die Videos angezeigt werden können. Falls Materialien angehängt wurden, wird ein Ordner namens „Materialien“ erzeugt, indem die eingefügten Dateien enthalten sind.

2.4 Neue Anforderungen

Wie durch die Beschreibung deutlich geworden ist, gibt es derzeit einige Schwierigkeiten bei dem Umgang mit der bestehenden „ViLM“-Software. Dies ist aber nur ein kleiner Aspekt der notwendigen Anpassung des Programms.

Den Wunsch, ein oder auch mehr als zwei Videos zu bearbeiten, kann das jetzige Programm nicht erfüllen. Daher soll eine flexible Version der Software erstellt werden, mit der beliebig viele Videos synchronisiert und anschließend mit Notizen und Materialien versehen werden können. Da „ViLM“ hauptsächlich im Umfeld von Unterrichtsanalyse und -reflexion eingesetzt wird, ist die Struktur für mögliche Beschreibungen vor allem hierauf zugeschnitten und findet sich in der *vilm.xml*-Datei wieder. Diese Struktur soll unabhängig von Unterricht sein und flexibel an beliebige Situationen angepasst werden können.

Durch die Verwendung von AIR³ und Adobe Flex⁴ für die Realisierung des gewünschten Players ist der bisher für die Umcodierung der Videos notwendige Serverteil nicht mehr erforderlich. Stattdessen wird eine Umcodierung in FLV⁵-Dateien nötig. Für diese Codierung soll eine BAT-Datei mit entsprechenden Parametern erzeugt werden. Bei Ausführung dieser Datei mittels „ffmpeg“⁶ wird ein Prozess zur Umcodierung angestoßen.

³AIR: Adobe Integrated Runtime, plattformunabhängige Laufzeitumgebung

⁴Framework zum Erzeugen von Rich Internet Applications

⁵FLV: Flash Video; Containerformat für Videodateien

⁶*ffmpeg*: Audio- und Video-Converter, näheres siehe Kapitel 6.5

3 „ViLM“ – Anpassung und Überarbeitung

Die Realisierung der neuen Anforderungen erfolgt nach dem Konzept der De-konstruktion. Zuerst fand eine Erkundung der bestehenden Software mit anschließender Analyse des Quellcodes statt. Danach wurde die Software entsprechend der neuen Anforderungen angepasst.

Die Systemanforderungen für „ViLM“ haben sich im Vergleich zur bisherigen Version nicht verändert. Die Aufteilung des Programms in Synchronisation und Annotation wurde beibehalten, wobei die Synchronisation nun die für die Umcodierung notwendige Datei erzeugt und deshalb in „ViLM – VideoVorbereitung“ umbenannt wurde. Wegen des neuen Players entfällt die aufwändige Umcodierung (Serverteil) nun, sodass es nur noch eine Zweiteilung des Projekts gibt.

3.1 Design-Entscheidung

Aufgrund der neuen Anforderungen (siehe Abschnitt 2.4) muss das bestehende Layout beider Programme überarbeitet werden. Das derzeitige Design kann nicht ohne Weiteres ein oder mehr als zwei Videos verarbeiten. Deshalb wurde die starre Layoutvorgabe, die bisher durch einen Splitcontainer umgesetzt war, entfernt. Nun muss eine Entscheidung für das Design der neuen Anwendung getroffen werden. Mögliche Designs, die verwendet werden können sind:

- Top-Level Window Interface
- Tabbed Document Interface
- Single Document Interface
- Multiple Document Interface⁷

Bei einem Top-Level Window Interface können beliebig viele Fenster eines Programms geöffnet werden, auch wenn nur eine Instanz der Anwendung läuft. Die Steuerung wird von einem zentralen Fenster übernommen. Eine große Schwierigkeit ist aber, dass das aktive Fenster bei Verlassen, um die Steuerung zu verwenden, den Fokus verliert. Somit ist dann keine Zuordnung der

⁷entnommen aus: Linux Magazin Online, E. Reitmayr, http://www.linux-magazin.de/heft_abo/ausgaben/2005/10/fensterordnung (Stand: September 2008)

ausgewählten Aktion zu dem Fenster mehr möglich. Da bei der Synchronisation aber die Videos beispielsweise zu unterschiedlichen Zeiten angehalten werden müssen, entfällt diese Möglichkeit.

Durch Tabbed Document Interface werden die Fenster einer Anwendung jeweils in Registerkarten geöffnet. Hierdurch ist die Anwendung sehr übersichtlich, allerdings können die Registerkarten nicht gleichzeitig nebeneinander angezeigt werden. Die parallele Betrachtung von Videos ist auf diesem Wege also nicht möglich.

Werden Single Document Interface-Anwendungen erstellt, wird jedes Fenster als eigenständiges Anwendungsfenster mit entsprechender Steuerung geöffnet. Die geöffneten Fenster können unabhängig von einander verschoben und/oder vergrößert werden. Falls mehrere Fenster geöffnet werden, wird die Darstellung der einzelnen Fenster in der Taskleiste schnell unübersichtlich.

Multiple Document Interface Anwendungen besitzen ein Hauptfenster, in dem beliebige Unterfenster geöffnet werden können. Diese Unterfenster können nur innerhalb des Hauptfensters verschoben und ggf. vergrößert bzw. verkleinert werden.

Die Entscheidung musste also zwischen Single und Multiple Document Interface gefällt werden. Da bei der Synchronisation beliebig viele Videos verwendet und bei der Annotation eine Bemerkung zu allen Videos erfolgen soll, fiel die Entscheidung auf eine Multiple Document Interface-Anwendung⁸. Ferner ist über ein zentrales Hauptfenster die Weitergabe von Informationen zwischen den Fenstern leicht umzusetzen. Mit einer MDI-Anwendung können beliebig viele Unterfenster erzeugt werden, die innerhalb des Hauptfensters beliebig verschoben oder in ihrer Größe angepasst werden können. Um die Übersichtlichkeit zu bewahren, sind einige Einstellungen für die Fenster (anordnen, Größe anpassen) vorgesehen. Ferner verliert der Benutzer durch ein Hauptfenster nicht so schnell die Orientierung.

3.2 „ViLM – VideoVorbereitung“

Die VideoVorbereitung beinhaltet, wie schon erwähnt, die Synchronisation der Videos. Die neue Oberfläche ist ähnlich zur bisherigen gestaltet. Dies soll besonders den Benutzern den Wechsel zur neuen Version erleichtern, die mit der bisherigen Version bereits vertraut sind. Für Benutzer, die „ViLM – Synchronisation“ noch nicht benutzt haben, soll die Verwendung einfacher gestaltet werden. Aus Gründen der Benutzerfreundlichkeit ist es in der neuen Version nicht mehr möglich, Button(s) zu betätigen, die in der derzeitigen Situation nicht sinnvoll sind. Beispielsweise muss der Benutzer ein geöffnetes Projekt schließen, bevor ein anderes erstellt oder geöffnet werden kann. Zum jetzigen Zeitpunkt ist

⁸nähere Informationen zu Multiple Document Interface siehe Kapitel 6.4

die Anzahl der Videos, die mit „ViLM“ bearbeitet werden können, auf vier beschränkt. Diese Beschränkung ist aber durch eine Variable im Quellcode vorgegeben, so dass eine Erweiterung ohne großen Aufwand beliebig möglich ist.

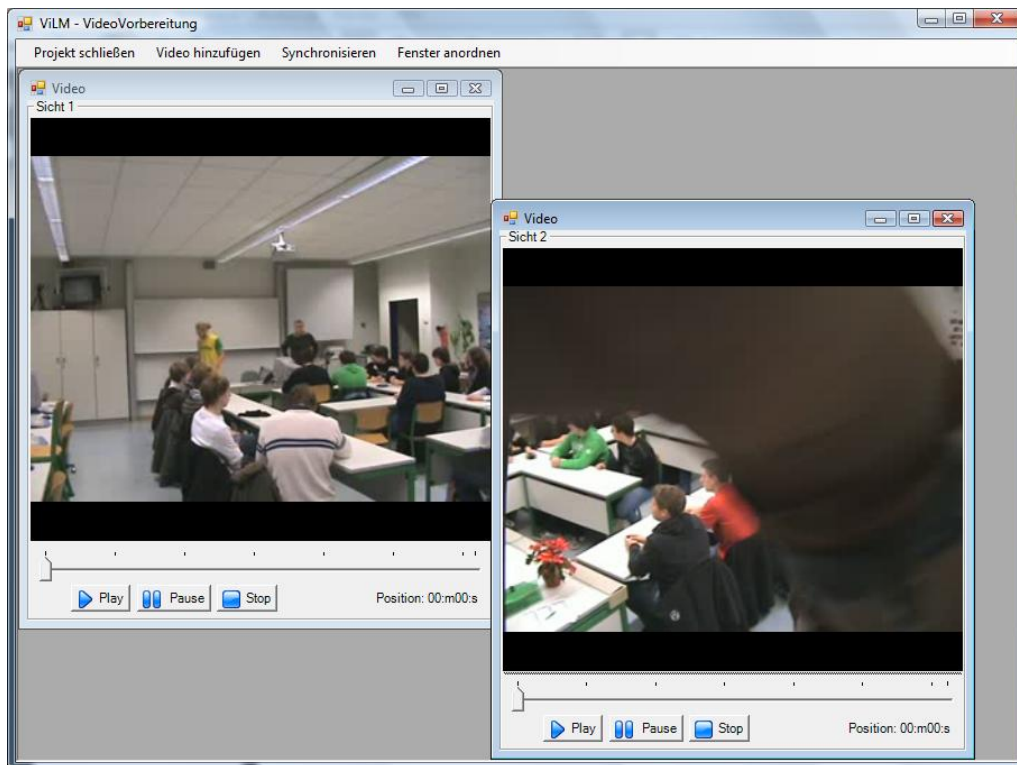


Bild 3.1: Überarbeitete „ViLM - Synchronisation“ Oberfläche

Nach dem Programmstart kann der Benutzer zwischen NEUES PROJEKT und PROJEKT ÖFFNEN wählen oder die Anwendung direkt wieder schließen. Wird ein neues Projekt erstellt, muss dem angezeigten Fenster ein Video zugeordnet werden. Erst nach diesem Schritt besteht die Möglichkeit zum Speichern oder Hinzufügen weiterer Videos. Beendet werden kann ein Projekt nach dem Neuerstellen oder Öffnen jederzeit.

Jedes der Videos hat standardmäßig die Bezeichnung „Sicht“ gefolgt von einer Zahl. Dieser Name kann vom Benutzer durch Anklicken umbenannt werden. Der gewählte Name erscheint bei der späteren Annotation erneut.

Falls mehr als ein Video geöffnet wurde, kann der Benutzer die Fenster anordnen lassen oder die Videos synchronisieren. Die Anordnung der Fenster ist eine neue Funktionalität, die für die Übersichtlichkeit hilfreich ist. Die Anordnung erfolgt nach der Reihenfolge, in der die Fenster geöffnet wurden. Das Fenster, welches zuerst geöffnet wurde, erscheint links oben, darunter das zweite usw. Diese Funktionalität war bei der bisherigen Version durch die starre Oberflächenstruktur nicht notwendig.

Durch das Synchronisieren wird automatisch eine BAT-Datei erzeugt, welche die notwendigen Informationen für „ffmpeg“ zur Umcodierung in FLV-Videos

beinhaltet. Diese Datei wird in dem Videoverzeichnis des Projektordners gespeichert. Sie muss allerdings aus dem Stammverzeichnis des Programms (meistens ist der Ordnername „ffmpeg“) heraus gestartet werden, da sonst die Umcodierung nicht ausgeführt wird.

Die Navigation innerhalb des Videos wird im jeweiligen Fenster vorgenommen. Die Funktionalität der einzelnen Button ist analog zu denen der bisherigen Version (siehe Kapitel 2.2).

3.3 „ViLM – Annotate“

Das Aussehen von „ViLM – Annotate“ ist an das bisherige Design angelehnt. Allerdings ist die Größe des Bereichs für die Annotationen nicht mehr veränderbar. Auch bei diesem Programm werden nur noch die Buttons angezeigt, welche verwendet werden können.

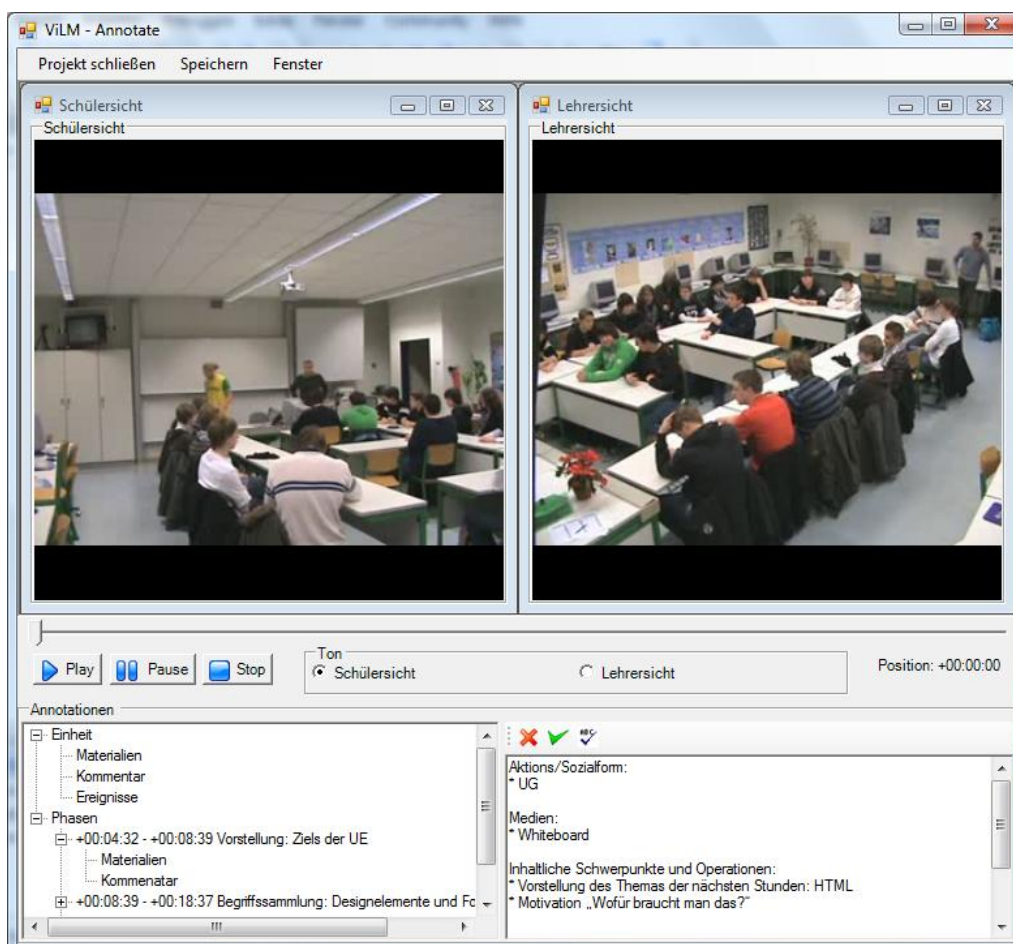


Bild 3.1: Neu gestaltete Oberfläche von „ViLM – Annotate“

Für das Kommentieren der gezeigten Situation ist es meist wichtig, alle Videos im Blick zu haben. Hierzu wird auch bei Annotate eine Möglichkeit zur Anordnung der Fenster geboten. Des Weiteren können die Fenster in einigen

vorgegebenen Formaten (siehe Bild 3.2) angezeigt werden, womit ein Verzerren des Bildes beseitigt werden kann. Ferner werden alle geöffneten Fenster aufgelistet, wie man es aus anderen Anwendungen kennt. Auf diese Weise sind sie für den Benutzer schnell zugänglich.

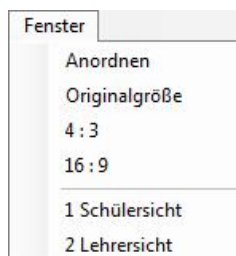


Bild 3.2: Menü FENSTER

Die Tonausgabe kann nun nicht mehr als Überlagerung der einzelnen Videos erfolgen. Der Benutzer wählt den Ton eines beliebigen Videos aus. Die Regelung der Lautstärke muss der Benutzer über seinen PC bzw. über die Lautsprecher vornehmen.

Wichtige Änderungen haben sich im Annotationsbereich ergeben: Bei der Auswahl von *Einheit* oder einer erstellten Phase wird jetzt direkt die Möglichkeit zur Texteingabe geboten. Dieser Eintrag entspricht dem früheren Knoten *Beschreibung*. Durch diese Änderung entfällt der Bereich, der das Setzen der Start- und Endzeit sowie das Aufnehmen in den Schnitt beinhaltete. Das Setzen von Start- und Endzeit ist über das Menü, welches mittels Klick mit rechter Maustaste auf eine Phase angezeigt wird, weiterhin möglich. Der Button IN SCHNITT AUFNEHMEN wird nicht mehr benötigt, weil er sich auf den Serverteil bezieht.

Das Kontextmenü entspricht weitgehend dem bisherigen. Allerdings können derzeit keine Unterphasen erzeugt werden. Da „ViLM“ zukünftig in den unterschiedlichsten Bereichen eingesetzt werden soll, ist auch die Beschreibungsstruktur möglichst flexibel zu gestalten. Deshalb wird dem Benutzer nun über das Kontextmenü die Möglichkeit gegeben, beliebig viele weitere Knoten innerhalb von Phasen hinzuzufügen, die er selbst benennen kann. Das Aussehen der Knoten entspricht dem der *Kommentar*-Knoten. Der standardmäßig eingefügte *Kommentar*-Knoten sowie alle selbst erzeugten Knoten können über das Kontextmenü umbenannt oder auch gelöscht werden.

Bei der Beschreibung von Events können eine Rechtschreibprüfung vorgenommen und Events ab dieser „ViLM“-Version abgespeichert werden. Analog zu der VideoVorbereitung ist die Navigation innerhalb der Videos umgesetzt.

3.4 Architektur

Da „ViLM“ in seiner jetzigen Version als MDI-Anwendung umgesetzt ist, gibt es in jedem Teilprojekt (VideoVorbereitung und Annotate) jeweils zwei Klassen, die Fenster erzeugen. In diesen Klassen sind bisher alle Informationen gespeichert, die für ein Projekt notwendig sind. Wird eine Projektdatei gespeichert (verläuft bei beiden Programmteilen analog), werden die Daten an eine Klasse weitergegeben, welche die XML- bzw. ANN-Datei erzeugt. Da für die Steuerung der Videos die Daten von Objekten der Oberfläche ausgelesen und an diese weitergegeben werden (z.B. Verschieben des *Silders*), wurde entschieden, keine ausgelagerte Klasse zu erzeugen, in der alle Informationen gespeichert sind. Somit konnte das häufig verwendete Konzept des Model-View-Controller⁹ nicht eingesetzt werden. Die jeweils notwendigen Informationen können über entsprechende *set-* und *get-*Methoden gesetzt bzw. abgerufen werden.

Der Aufbau von „ViLM – VideoVorbereitung“ und „ViLM – Annotate“ im Zusammenspiel mit dem neuen Player kann man sich in etwa so vorstellen: Die

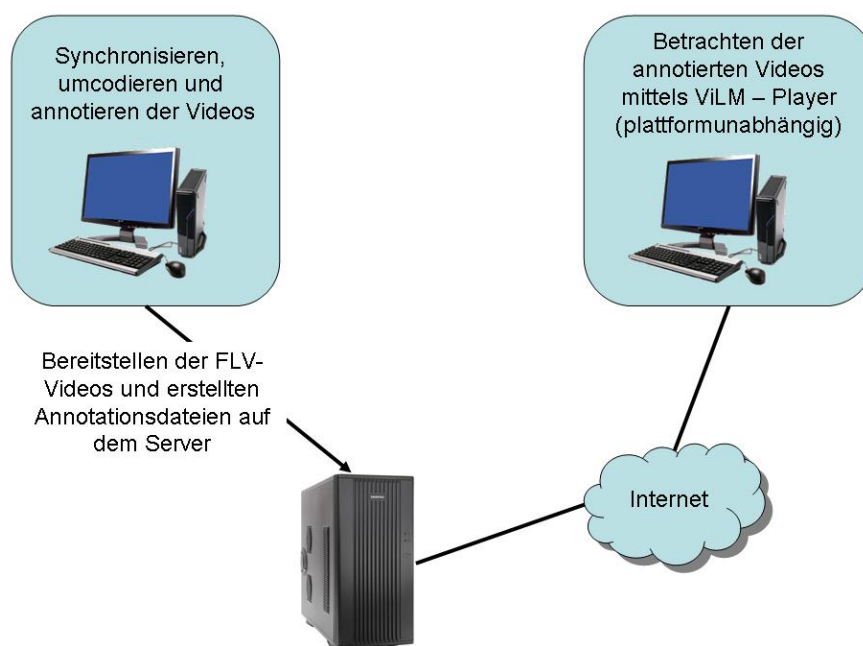


Bild 3.1: Kooperation von „ViLM“ mit dem neu erstellten Player

erstellten Annotationsdateien und die umcodierten Videos werden auf einem Server zur Verfügung gestellt. Über den lokal installierten Player wird eine auf dem Server liegende ANN-Datei geöffnet und die gespeicherten Informationen angezeigt. Die Videos werden als Stream geladen und können direkt abgespielt werden.

⁹Model-View-Controller ist ein Architekturmodell, welches die Einteilung eines Softwareprojekts in Daten (Model), Präsentation (View) und Steuerung (Controller) vorsieht.

4 Mögliche Einsatzszenarien

Um mögliche Einsatzgebiete für die neue „ViLM“-Software zu verdeutlichen, folgen nun einige ausgesuchte Situationen, in denen die Software sinnvoll und unterstützend eingesetzt werden kann.¹⁰ Für alle Situationen mit mehr als einer Kamera, in denen „ViLM“ eingesetzt wird, ist es zwingend erforderlich, dass sich die Bildbereiche der Videokameras überschneiden, damit ein Zeitpunkt zur Synchronisation gesetzt werden kann. Natürlich können die Kameras anschließend beliebig geschwenkt werden. Die Platzierung im Raum sollte allerdings so erfolgen, dass sie den Ablauf der Situation nicht (oder möglichst wenig) beeinflussen.

4.1 Der Vortrag – Ein-Kamera-Perspektive

Jeder Vortrag bezieht sich auf ein mehr oder weniger begrenztes Thema. Für die Zuhörer ist es wichtig, dass sie dem Vortrag gut folgen können. Hierzu ist eine klare Gliederung und ein roter Faden sehr sinnvoll. Ferner kann man die Art betrachten, wie jemand seinen Vortrag gestaltet:

- Ist die Wortwahl und der Ausdruck angemessen?
- Unterstützen Gestik und Mimik das Gesagte?
- Wie ist die Argumentation des Vortragenden aufgebaut?
- Wird schlüssig argumentiert?
- Werden für den Vortrag verschiedene Medien (Beamer, Overhead-Projektor, Tafel o. Ä.) eingesetzt?
- Erfolgt der Einsatz der Medien sinnvoll?
- Wie wird auf Zwischenfragen reagiert?
- Wie geht der Redner mit Zwischenfragen um? Kann er sie kompetent und verständlich beantworten?

¹⁰Vor der Aufzeichnung sollte man sich das Einverständnis der gefilmten Personen einholen, um das Material auch in einem eingeschränkten, vorher bekanntgegebenen Benutzerkreis zu verwenden. Ein Beispiel für eine Zustimmungserklärung findet sich im Anhang A.

Die genannten Aspekte bieten eine Auswahl an Kriterien, die berücksichtigt werden können. Für eine genauere Auswertung können beliebige Aspekte ergänzt oder andere ausgelassen werden. Ohne den Einsatz von „ViLM“ kann sich ein Beobachter während des Vortrags Notizen machen oder ggf. eine Videoaufzeichnung des Vortrags zur Unterstützung heranziehen. Der Vortrag und die Anmerkungen stehen dann jedoch unverbunden nebeneinander. Natürlich kann man einzelne Sätze genau analysieren, allerdings ist eine gleichzeitige Betrachtung des Videos mit digitalen Bemerkungen nur eingeschränkt möglich. Werden solche Notizen anderen Personen zur Beurteilung vorgelegt, können sie die beschriebenen Aspekte oft nur schwer nachvollziehen.

Um die Nachvollziehbarkeit zu vereinfachen, kann „ViLM“ eingesetzt werden. Für die Verwendung ist eine Aufzeichnung des Vortrags (auf Video) erforderlich. Hierbei sollte die Kamera auf den Redner ausgerichtet sein. Für die Auswertung/ Analyse kann eine Einteilung des Vortrags in Phasen, z.B. anhand der Gliederung, durchgeführt werden. In den jeweiligen Phasen kann der Beobachter dann die Aspekte als Knoten aufnehmen, die er für relevant hält und diese mit entsprechenden Bemerkungen versehen. Durch die Zuordnung von Phasen innerhalb des Videos ist eine Strukturierung in Abschnitte möglich, zwischen denen man beliebig hin und her springen kann. Es können unterschiedliche Aspekte in den verschiedenen Phasen berücksichtigt oder gleiche Aspekte von unterschiedlichen Phasen verglichen werden. So kann man leicht nachvollziehen, ob und wie sich das Verhalten des Vortragenden im Verlauf geändert hat. Ferner bietet „ViLM“ die Möglichkeit, dass auch weitere Personen die eigenen Notizen nachvollziehen können. Sie sehen die Situation im Video und können die entsprechenden Kommentare dazu nachlesen. Über diese Beschreibungsmöglichkeit kann anschließend diskutiert werden. Gegebenenfalls können auch Anpassungen oder Ergänzungen des Geschriebenen vorgenommen werden.

4.2 Die Filmanalyse – Ein-Kamera-Perspektive

Für die Filmanalyse ist keine eigene Aufzeichnung erforderlich. Der gewünschte Film sollte jedoch zur Annotation digital vorliegen. Eine Filmanalyse beurteilt unterschiedliche Kriterien, die nachfolgend beispielhaft aufgeführt sind.

- *Bildgestaltung durch die Kamera:*
 - *Welche Einstellungsgrößen werden wie verwendet?*
 - *Welche Kameraperspektiven werden wie verwendet?*
 - *Welche Arten der Kamerabewegung werden wie verwendet?*
- *Bildgestaltung nach der Aufnahme:*
 - *Dominieren kurze oder lange Einstellungen?*

- *An welchen Stellen wird wie geschnitten?*
- *Welche Tricks werden an welchen Stellen verwendet?*
- *Komposition einzelner Stilelemente - Montage:*
 - *Wie werden einzelne Bildgestaltungsmittel miteinander verbunden?*
 - *Wie werden Bildgestaltungsmittel mit Tongestaltungsmitteln verbunden?*
 - *Welche der analysierten Gestaltungsmittel werden wie miteinander verbunden (Montage), um den Gesamteindruck des Films zu erzielen?*
- *Vertonung des Films:*
 - *Welche Sprache (bzw. Schrift bei Einblendung) wird verwendet (Kommentar, Dialog, Erzähler ...)?*
 - *Wie sprechen die Darsteller (Tonfall, Dialekt, Hochsprache, Stimme sympathisch oder unsympathisch ...)?*
 - *In welchem Verhältnis steht die Sprache zum Bild (unwichtig, erklärt die Bilder, hat nichts mit den Bildern zu tun ...)?*
 - *Wie wird welche Art von Musik eingesetzt?*
 - *Wie werden welche Töne und Geräusche eingesetzt?*
- *Ausstattung:*
 - *Welches Dekor (Bauten, Landschaften, Requisiten) wird wie verwendet?*
 - *Wie sind die Darsteller durch Masken, Kostüme, Mimik, Gestik usw. gekennzeichnet?*
 - *Wie wird Farbe als Gestaltungsmittel eingesetzt?*
 - *Wie werden Licht/Schatten (Beleuchtungseffekte) als Gestaltungsmittel eingesetzt?¹¹*

Bei einer schriftlichen Filmanalyse besteht keine direkte Verknüpfung zwischen einzelnen Notizen und der beschriebenen Situation. Man kann sich zur Orientierung zwar die aktuelle Spieldauer notieren, allerdings kann man keine Markierung auf dem Video setzen oder spezielle Punkte direkt ansteuern.

Durch die Verwendung von „ViLM“ zur Analyse kann man einzelne Situationen als *Events* kennzeichnen und entsprechende Notizen zu einzelnen Aspekten notieren. Welcher Aspekt wo beschrieben ist, kann durch eine entsprechende Benennung der Knoten erfolgen.

¹¹entnommen aus: Tiroler Bildungsserver,
<http://www.bildungsservice.at/faecher/be/start.htm> (Stand: September 2008)

Da beliebig viele Knoten hinzugefügt werden können, kann zu jeder Situation eine beliebig genaue Analyse vorgenommen werden. Durch die Phaseneinteilung kann man auch Abschnitte des Films genauestens auswerten. Um an die gewünschte Positionen innerhalb des Videos zu gelangen, können die entsprechenden Start- und Endzeiten angesprungen werden, so dass das Auffinden sehr leicht und ohne großen Aufwand möglich ist. Ferner können auch Ergänzungen eingepflegt werden. Falls verschiedene Personen eine Analyse desselben Films mit „ViLM“ durchführen, können z.B. Phaseneinteilungen und festgelegte Events leicht verglichen werden.

4.3 Das Bewerbungsgespräch – Zwei-Kamera-Perspektive

Das Bewerbungsgespräch ist ein klassisches Zwiegespräch: meist ein bis zwei Vertreter¹² der Firma (Abteilungsleiter o. Ä.) und ein Bewerber für den Arbeitsplatz. Ein denkbarer Verlauf eines Bewerbungsgesprächs kann folgender sein (Diese Phasen können aber beliebig benannt oder auch ausgelassen werden):

- *Vorlauf/Anreise*
- *Begrüßung*
- *Smalltalk zum Aufwärmen*
- *Darstellung des Unternehmens*
- *Kurze Selbstpräsentation des Bewerbers*
- *Fragen nach persönlichen Kompetenzen*
- *Fragen nach fachlichen Kompetenzen*
- *Klärung von Gehaltsfragen, Eintrittstermin usw.*
- *Gelegenheit, selbst Fragen zu stellen*
- *Abschluss, Klärung von Terminfragen, weiteres Vorgehen usw.*¹³

Die Anzahl der Phasen deutet bereits an, dass diese Gespräche sehr umfangreich sind. Für die Auswertung des Gespräches ohne Videoaufzeichnung ist der subjektive Eindruck der Anwesenden ausschlaggebend. Sie geben an, ob das gezeigte Verhalten des Bewerbers angemessen war und ob sich der Bewerber gut präsentieren konnte.

Durch den Einsatz von „ViLM“ können sich auch weitere Personen, die nicht selbst an dem Gespräch beteiligt waren, einen Eindruck von den Bewerbern machen. Hierzu ist die Aufzeichnung auf Video erforderlich. Die eingesetzten

¹²um nicht im Singular und Plural zu schreiben, gehe ich im folgenden Text von zwei Vertretern der Firma aus

¹³entnommen aus: <http://www.wiwi-treff.de/home/index.php?mainkatid=3&ukaid=3&sid=303&artikelid=517&pagenr=0> (Stand: September 2008)

Kameras sollen das Verhalten des Bewerbers auf Fragen des Firmenvertreters festhalten. In einigen Gesprächen wird durch die Reaktion auf gegebene Antworten mit Mimik und Gestik versucht, den Bewerber zu verunsichern. In solchen Situationen zeigt sich jedoch im Besonderen, wie sicher der potentielle neue Mitarbeiter auftritt. Das Verhalten des Bewerbers kann bei gewöhnlichen Bewerbungsgesprächen (ohne Videokamera) nur von den anwesenden Personen beurteilt werden. „ViLM“ ermöglicht (auf Basis der beiden Videos) eine genaue Analyse des Gespräches von unterschiedlichen Personen.

Nach einer Gliederung des Gesprächs besteht die Möglichkeit, jeweils eine Beschreibung und Kommentare zu den einzelnen Situationen festzuhalten. Durch das Abspielen, Anhalten und Zurückspringen innerhalb der synchronen Videos kann jede Szene genau analysiert werden. Man kann hiermit das Verhalten des Bewerbers in Stresssituationen und die Reaktion auf bestimmte Fragen noch einmal nachvollziehen und auf Dinge achten, die während des Gesprächs selbst vielleicht nicht wahrgenommen oder bereits vergessen wurden (z.B. Blickrichtung, Gestik, Mimik, zögernde Antworten). Besonders bei einer großen Anzahl von Bewerbern kann es hilfreich sein, sich einzelne Situationen noch einmal ins Gedächtnis zu rufen. Durch die Vergleichsmöglichkeit mit anderen Bewerbern kann auf diesem Weg ein für die Firma passender Kandidat bestimmt und eingestellt werden.

4.4 Die Unterrichtsanalyse – Zwei-Kamera-Perspektive

Die Verwendung von „ViLM“ zur Unterrichtsanalyse und -reflexion war bisher das hauptsächliche Einsatzumfeld. Die Verwendung in diesem Bereich besteht natürlich weiterhin. Allerdings haben die Benutzer jetzt die Möglichkeit die Struktur ihrer Beschreibung selbst zu beeinflussen. Da sich Unterricht in bestimmte Phasen einteilen lässt, werden auch hier eine mögliche Einteilung angefügt:

- *Problemstellung*
- *Zielvereinbarung und Bedeutsamkeit*
- *Verständigung über das Vorgehen*
- *Erarbeitung von Grundlagen für die Problemlösung*
- *Durchführung der Problemlösung*
- *Vergleich und Zusammenfassung*
- *Anwendung*
- *Weiterführung und Bewertung*¹⁴

¹⁴S. Blömeke, B. Herzig, G. Tulodziecki: Gestaltung von Unterricht - Eine Einführung in die Didaktik. S. 105ff. Bad Heilbrunn, 2004.

Die genannten Gesichtspunkte können als Phaseneinteilung des Unterrichts dienen. Es steht den Nutzern allerdings frei, beliebige andere Einteilungen vorzunehmen.

Durch den Einsatz von „ViLM“ in der Lehrerbildung kann den angehenden Lehrern ihr eigenes Verhalten bewusst vor Augen geführt werden. Vor allem die Reaktion auf unerwartete Situationen im Unterricht kann auf diese Weise deutlich gemacht werden. Ferner kann anhand der Videos mit Unbeteiligten darüber diskutiert werden, welches Verhalten evtl. angemessener gewesen wäre.

Bei zwei vorliegenden Videos besteht immer auch die Möglichkeit, diese getrennt anzuschauen und auszuwerten. Eine parallele Sicht, in der der direkte Vergleich von Aktion und Reaktion zu sehen ist, kann die Verhaltensweisen beider Seiten deutlich herausstellen. Der Lehrperson wird auf diesem Weg das eigene Verhalten aus einer anderen Perspektive vor Augen geführt, die man sonst nicht betrachtet. Des Weiteren ist mittels „ViLM“ das Springen innerhalb der Videos parallel möglich, so dass man nicht versuchen muss, beide Videos in derselben Situation zur selben Zeit zu starten.

4.5 Die Gruppenarbeit – Vier-Kamera-Perspektive

Eine beliebte Sozialform im Unterricht ist die Gruppenarbeit. Die Anzahl der Gruppen und die der Gruppenmitglieder ist abhängig von der Klassenstärke. Dieses Szenario soll sich auf vier Gruppen beziehen, die jeweils von einer Kamera beobachtet werden. Für diese Beschreibung der Gruppenarbeit ist die Anzahl der Mitglieder sowie die Vorbereitung (Einleitung der Gruppenarbeit, Einteilung der Gruppen usw.) nicht relevant. Die Aufzeichnung der Videos beginnt also direkt mit dem Start der Gruppenarbeitsphase.

Damit alle Gruppenmitglieder wissen, was ihre Aufgabe ist, soll ein klarer Arbeitsauftrag mit einem vorgegebenen Ziel bekannt gegeben werden. Zur Erreichung des Ziels müssen die Mitglieder zusammenarbeiten. Da die Gruppenarbeit ein Ergebnis liefern soll, ist eine abschließende Präsentation sinnvoll.

Durch die Aufzeichnung der Gruppenarbeiten kann man nachvollziehen, welche Schüler aktiv mitgearbeitet haben bzw. wer sich eher zurückgehalten hat. Ferner kann man einen Vergleich durchführen, welche Reihenfolge für die Bearbeitung der Aufgabe gewählt wurde, wie unterschiedlich die Herangehensweise an die Aufgabe waren und vieles Andere mehr. Mit „ViLM“ können die Videos so kommentiert werden, dass leicht erkennbar ist, wann in welcher Gruppe welche Aktivität vorgenommen wurde. Eine Einteilung in Phasen (so weit es diese bei vier unterschiedlichen Gruppen gibt) kann ebenfalls vorgenommen werden. Bei der Verwendung von „ViLM“ in diesem Umfeld ist das Setzen von Ereignissen wahrscheinlich eine praktikable Möglichkeit, bestimmte Szenen der einzelnen Gruppen zu kennzeichnen und zu beschreiben.

Über die Videoaufzeichnung bietet sich dem Lehrer die Möglichkeit, die Einsatzbereitschaft jedes Einzelnen genauer zu beurteilen. Zwar ist es bei Gruppenarbeiten nicht immer möglich, dass jeder Schüler den gleichen Arbeitsanteil leistet, aber die geleisteten Beiträge spiegeln meist die Bereitschaft wider. Durch „ViLM“ wird dem Lehrer ein besserer Einblick in die einzelnen Gruppen gegeben. Bei einer erneuten Gruppenarbeitsphase kann er dann gewonnene Informationen (z.B. über Gruppeneinteilung, Zeiteinteilung) zur besseren Planung und Durchführung berücksichtigen. Außerdem kann er den Schülern über das Vorspielen der parallelen Videos mit „ViLM“ vermitteln, wie unterschiedlich die Arbeitsweisen der Gruppen waren. Dies kann einen Anreiz für die nächste Gruppenarbeitsphase bieten, so dass jeder einzelne seine Einsatzbereitschaft verbessert.

4.6 Die Pyramidendiskussion – *n*-Kamera-Perspektive

Der Einsatz von „ViLM“ soll für möglichst viele Situationen sichergestellt werden. Um eine Vorstellung davon zu haben, welche Kontexte mit der Verwendung der Software abgedeckt werden können, wird hier ein Beispiel für den Einsatz beliebig vieler Kameras aufgezeigt.

Bei einer Pyramidendiskussion wird eine Frage- oder Problemstellung zuerst an jeden Einzelnen gestellt. Nachdem jeder seine Lösung notiert hat, werden Zweiergruppen gebildet. In diesen Gruppen werden die eingebrachten Lösungen diskutiert und anschließend eine gemeinsame Lösung festgehalten. Nun werden zwei Zweiergruppen zu einer Vierergruppe zusammen gefasst und die Diskussion beginnt erneut. Dieser Vorgang wird so lange wiederholt, bis es nur noch eine Gesamtgruppe gibt.

Mit Hilfe von Kameras in den jeweiligen Gruppen können unterschiedliche Herangehensweisen und Argumentationsketten dokumentiert und analysiert werden. Besonders in den kleinen Gruppen können sehr unterschiedliche Ergebnisse ermittelt werden, die es dann auf einen Nenner zu bringen gilt. Ferner ist so das unterschiedliche Verhalten Einzelner in den verschiedenen Gruppenkonstellationen beobachtbar.

5 Anwendungsfälle

Die Anwendungsfälle sollen beispielhaft beschreiben, wie der Einsatz von „ViLM“ aussehen kann. Zu jeder Kameraanzahl der zuvor beschriebenen Szenarien wird nachfolgend ein Beispiel skizziert.

5.1 Rhetorik-Schulung – Ein-Kamera-Perspektive

In einem mehrtägigen Rhetorik-Seminar für Führungskräfte soll der Vortragsstil der Teilnehmer verbessert werden. Hierzu werden im Seminarverlauf unterschiedliche Aspekte angesprochen und erklärt, die für einen gelungenen Vortrag wichtig sind. Um den schrittweisen Erfolg des Seminars zu dokumentieren und den Vortragenden ihr eigenes Verhalten bewusst zu machen, wird am Ende eines Seminartages ein kurzer Vortrag (etwa 15 Minuten) von einem der Teilnehmer gehalten. Der Vortrag wird auf Video aufgezeichnet und auf einen PC gespeichert um später mit „ViLM“ genau analysiert zu werden. Am nächsten Tag soll eine Analyse der Aufzeichnung in der gesamten Gruppe unter Berücksichtigung einiger Kriterien erfolgen:

- Angemessene Wortwahl und verständlicher Ausdruck
- Unterstützende Gestik und Mimik
- Aufbau der Argumentation; Ist die Argumentation schlüssig?
- Medieneinsatz (Beamer, Overhead-Projektor, Tafel o. Ä.)
- Reaktion auf Zwischenfragen

Zu Beginn wird ein Teilnehmer festgelegt, der für die Bedienung des Computers zuständig ist. Dieser ist für die Steuerung des Videos (starten, anhalten usw.) sowie für die Notation von Anmerkungen zuständig. Bevor mit der Annotation begonnen werden kann, wird eine vilm.xml-Datei mittels „ViLM – Video-Vorbereitung“ erstellt (Dieser Schritt ist derzeit noch erforderlich). Anschließend wird ein neues Annotationsprojekt angelegt, welches die zuvor erstellte vilm.xml-Datei verwendet.

Zur besseren Übersicht wird der Vortrag in Phasen eingeteilt, die sich an den Abschnitten des Vortrags orientieren. Diese Einteilung sollte recht schnell durchgeführt werden können, da der Vortrag relativ kurz gehalten ist. An den entspre-

chenden Zeitpunkten wird ein neuer Phasenknoten im Annotationsbereich erzeugt. Nun kann eine allgemeine Beschreibung des Inhalts der jeweiligen Phasen erfolgen. Anschließend wird der Kommentar-Knoten umbenannt in „Wortwahl und Ausdruck“ und vier weitere Knoten erzeugt, die mit den oben aufgeführten Kriterien benannt werden. Diese Aspekte sollen in allen Phasen berücksichtigt werden, so dass diese Knoten entsprechend bei den übrigen Phasen ergänzt werden müssen.

Nach dieser Vorbereitung kann das Video gestartet werden. Wird das Ende einer Phase erreicht, sollte das Video angehalten werden, damit über die Umsetzung der Kriterien gesprochen werden kann. Für die Diskussion ist es sinnvoll, die Kriterien der Reihe nach zu besprechen um entsprechende Notizen im Annotationsbereich machen zu können. Die Teilnehmer beginnen also mit ihren Eindrücken über die Wortwahl und den Ausdruck des Vortragenden. Damit keine Informationen verloren gehen, sollten die genannten Aspekte in Stichworten festgehalten werden. Nachdem die genannten Gesichtspunkte besprochen wurden, muss eine entsprechende Notiz im Annotationsbereich vorgenommen werden. Dieser Vorgang wiederholt sich für alle Aspekte und bei allen Phasen. Abschließend kann dann ein Gesamteindruck des Vortrags gegeben werden.

5.2 Die Unterrichtsanalyse – Zwei-Kamera-Perspektive

Im Rahmen des Seminars MIU¹⁵ plant ein Student eine Unterrichtsstunde. Die Stunde soll am darauf folgenden Tag in der Schule stattfinden. Damit eine genaue Analyse der Unterrichtssituation möglich ist, werden zwei Kameras samt Stativen für die Aufzeichnung der Stunde bei der Fachgruppe „Didaktik der Informatik“ ausgeliehen.

Vor Unterrichtsbeginn werden die Kameras im Klassenraum platziert: eine im hinteren Bereich des Raumes um die Aktionen des angehenden Lehrers¹⁶ festzuhalten, die zweite im vorderen Bereich um die Schüler zu beobachten. Damit beide Videos später synchron abgespielt werden können, müssen sich die Bildbereiche während der Aufzeichnung etwas überschneiden und ein Zeichen (z.B. in die Hände klatschen) gegeben werden. Falls möglich, sollte jemand an der auf den Lehrer gerichteten Kamera stehen um diese bei Bedarf der Position des Lehrers anzupassen.

Nach dem die Unterrichtsstunde beendet ist, werden die Kameras zurück in die Fachgruppe gebracht, wo die Videos von der Kamera auf einen Server kopiert werden. Die zur Verfügung gestellten Videos werden von dem Studenten für die Bearbeitung mit „ViLM“ verwendet. Mit „ViLM – VideoVorbereitung“ werden die beiden Videos geladen und auf das angegebene Zeichen synchronisiert.

¹⁵MIU: Methoden des Informatikunterrichts in Theorie und Praxis

¹⁶zur Verkürzung später nur als Lehrer bezeichnet

Anschließend wird mit „ViLM – Annotate“ ein Annotationsprojekt erzeugt. Die Einteilung in Phasen, die an die geplanten Unterrichtsphasen angelehnt sind, kann bereits vorgenommen werden, ebenso eine allgemeine Beschreibung der Einheit sowie der Phasen. Da während des Seminars die Videoaufzeichnungen noch diskutiert werden, sollte mit der Analyse erst nach der Besprechung begonnen werden.

Während der Diskussion über die gehaltene Unterrichtsstunde im Seminar sollte sich der Student, der den Unterricht geleitet hat, Notizen machen, um alle angesprochenen Aspekte später aufgreifen zu können. Hier werden Situationen angesprochen, die gut verlaufen sind oder auch solche, in denen man sich besser anders verhalten hätte.

Nun sollte sich der Student an die genaue Analyse seiner Unterrichtsstunde begeben. Einige Anhaltspunkte wurden bereits im Seminar aufgezeigt, die jetzt aufgegriffen und genau ausgewertet werden müssen. Er sollte sich die einzelnen Phasen genau anschauen und auch auf Einflüsse achten, die man vielleicht erst beim zweiten Hinschauen wahrnimmt. Die von ihm festgestellten Aspekte sollen kommentiert und analysiert werden. Um einen Gesamteindruck der Einheit zu vermitteln, kann ein allgemeiner Kommentar zur Unterrichtseinheit (unter *Einheit*) notiert werden.

Die während der Unterrichtsstunde verwendeten oder erstellten Schaubilder, Folien usw. können, falls sie digital vorhanden sind, der entsprechenden Phase zugeordnet werden. So erhält der Student am Ende eine vollständige Zusammenstellung seines Unterrichts mit einer allgemeinen Beschreibung, Einteilung in Phasen und einer genauen Analyse. Diese Stunde kann dann auch von anderen Studenten oder Interessierten angeschaut und nachvollzogen werden.

5.3 Gruppenarbeit zum Thema „Mauerfall“ – Vier-Kamera-Perspektive

Auch die hier geschilderte Situation kann im Rahmen einer Unterrichtsreihe behandelt werden. Hier ist aber die Aufmerksamkeit auf die unterschiedlichen Gruppen zu richten, nicht auf die Lehrperson. In diesem Einsatzbereich von „ViLM“ möchte die Lehrperson das Verhalten von Schülern in kleinen Gruppen untersuchen.

Für die Gruppenarbeit wird die Klasse in vier möglichst gleich starke Gruppen eingeteilt. Jede Gruppe soll von einer Kamera beobachtet werden, allerdings muss nach dem Start (oder ggf. auch am Ende) eine Überschneidung der Bildbereiche erzeugt werden. In diesem Bereich wird ein Zeichen zur Synchronisation gegeben. Anschließend werden die Kameras auf die Gruppen ausgerichtet.

Alle Gruppen bekommen die Aufgabe: Sie sollen positive und negative Gesichtspunkte zum Thema „Mauerfall“ aus einem bereitgestellten Text ermitteln.

Um ihre Ergebnisse später der gesamten Klasse mitteilen zu können, bekommt jede Gruppe ein Plakat und einige dicke Stifte. Eine Person aus der Gruppe soll am Ende der Stunde die Ergebnisse vorstellen. Damit den Schülern die Bearbeitungszeit bekannt ist, wird eine Uhrzeit vorgegeben, zu der die Plakate fertig gestellt sein sollen. Sie sollen ihre Ergebnisse möglichst übersichtlich auf den Plakaten darstellen. Zur Präsentation der Gruppenarbeit können die Kameras auf die präsentierenden Schüler gerichtet werden. Für eine abschließende Analyse ist die Art der Plakate und der Darstellung vielleicht auch interessant.

Nachdem der Unterricht beendet ist, muss eine Auswertung der erstellten Videos erfolgen. Hierzu werden die Videos auf einen PC gespeichert und mittels „ViLM – VideoVorbereitung“ benannt und synchronisiert. Anschließend kann eine Analyse der Gruppenarbeit vorgenommen werden.

Mit „ViLM – Annotate“ werden nun alle Videos betrachtet. Unter *Einheit* kann eine allgemeine Beschreibung der gezeigten Tätigkeiten und falls gewünscht ein Kommentar notiert werden. Da jede Gruppe eine andere Herangehensweise an die Aufgabenstellung hat und auch das Arbeitstempo verschieden ist, ist eine Einteilung in Phasen wenig sinnvoll. Besser ist in einer solchen Situation die Verwendung von *Events*. Falls in einer der Gruppen etwas Auffälliges passiert, kann ein Event erzeugt werden, zu dem dann eine entsprechende Notiz gemacht wird. Zur besseren Übersicht der verschiedenen Events ist es vielleicht hilfreich, den Titel des Videos in der Bezeichnung des Events mit aufzunehmen.

Bei der Betrachtung der Videos kann die Lehrperson auf Gesichtspunkte achten, die sonst im Unterrichtsverlauf nicht wahrgenommen werden. So wird z.B. deutlich, wer sich während einer Gruppenarbeitsphase stark engagiert oder wer andere Arbeiten erledigt. Durch eine so genaue Beobachtung können Arbeitsweisen gut ermittelt werden. Falls eine besonders gute Arbeitsweise gefunden wurde, kann diese möglicherweise für eine spätere Gruppenarbeitsphase als Orientierung oder Strukturierung vorgegeben werden. Ferner kann die Beurteilung der Arbeitsleistungen durch die Videos sehr genau vorgenommen werden.

Natürlich kann eine solche Zusammenschau der Videos (evtl. nur in Ausschnitten) den Schülern gezeigt werden. Vielleicht dient dies bei einer nachfolgenden Gruppenarbeit als Motivation. Falls eine besonders gute Unterrichtsstunde auf diese Weise entstanden ist, kann es vielleicht auch für Lehrerkollegen interessant sein, sich die Videos anzuschauen.

5.4 Pyramidendiskussion zur Staatsgründung – *n*-Kamera-Perspektive

Da ein konkretes Beispiel für *n* Kameras schwierig zu formulieren ist, wurde eine Situation herausgesucht, die sich auf beliebig viele Sichten erweitern lässt. Ähnlich wie die zuvor dargestellte Situation ist eine Diskussion zur Staatsgründung

ebenfalls ein mögliches Unterrichtsthema. Wie bereits in Kapitel 4.6 beschrieben, verlaufen Pyramidendiskussionen zu Beginn in kleinen Gruppen, die dann nach und nach zu einer gesamten Gruppe zusammengeschlossen werden. Da eine Doppelstunde im Unterricht nicht ausreicht, um mit Einzelarbeit zu beginnen, werden für diese Diskussion sechs Arbeitsgruppen gebildet (natürlich kann bei größerem Zeitumfang auch mit kleineren Gruppen begonnen werden). Jede dieser Gruppen wird von einer Kamera beobachtet. Zu Beginn (oder zum Schluss) der Aufzeichnung wird ein Synchronisationszeichen gegeben, welches von allen Kameras aufgezeichnet werden muss.

Die sechs Gruppen stellen kleine Ausschüsse im Parlament dar, die über die Dringlichkeit von Fragen der Grundrechts- und Verfassungsdiskussion beraten. Hierzu erhält jede Gruppe jeweils fünf verschiedene Themen. Diese müssen ihrer Dringlichkeit nach innerhalb der Gruppen eingestuft werden. Eine Begründung für die jeweilige Reihenfolge sollen die Schüler schriftlich festhalten. Danach werden drei Gruppen zu jeweils einem erweiterten Verfassungsausschuss gebildet. Die Kameras müssen nun so positioniert werden, dass sie die neue Gruppenkonstellation widerspiegelt.

In dem neuen Ausschuss trägt jeweils ein Sprecher der drei Unterausschüsse die ermittelten Ergebnisse vor. Nun müssen die Schüler in dieser Gruppe die vorgetragenen Themen ihrer Wichtigkeit nach ordnen. Aus der ermittelten Reihenfolge wird gemeinsam ein Dringlichkeitsantrag an das Parlament formuliert.

An dieser Stelle werden die Gruppen zu einem Plenum zusammengefasst, so dass die Kameras ggf. wieder neu ausgerichtet werden müssen. Hier werden die begründeten Entscheidungen der Verfassungsausschüsse vorgestellt und der erstellte Antrag zur Abstimmung vorgetragen¹⁷.

Nachdem die Videos auf dem Computer gespeichert und synchronisiert wurden, kann mit der Annotation begonnen werden. Bei diesem Szenario ist eine Einteilung in Phasen möglich. Diese Phasen beginnen/enden jeweils mit der Neubildung der Gruppen. Zu diesen kann dann jeweils eine Beschreibung notiert werden (z.B. bei der ersten Phase: Ausschüsse im Parlament). Falls an dieser Stelle kein Kommentar gewünscht ist, kann der standardmäßig vorhandene Knoten gelöscht werden.

Auch für diese Form der Gruppenarbeit ist die Verwendung von *Events* sinnvoll. Allerdings sollte eine Benennung so erfolgen, dass man eine Zuordnung erkennen kann. Eine ausführliche Beschreibung kann wie gewohnt durchgeführt werden.

¹⁷Die Informationen für dieses Unterrichtsbeispiel stammen von <http://www.erinnerungsstaette-rastatt.de/pyramidendiskussion.htm> (Stand: September 2008)

6 Verwendete Technologien und Konzepte

In diesem Kapitel werden kurz die verwendeten Technologien und Konzepte vorgestellt, die für „ViLM“ relevant sind. Dabei wird an einigen Stellen mit Code-Beispielen gezeigt, wie die Umsetzung innerhalb des Programms realisiert wurde.

6.1 Objektorientierung

Dieses Projekt ist nach dem Konzept der Objektorientierung umgesetzt worden. Bei diesem Konzept werden Informationen und Eigenschaften, die zueinander in Beziehung stehen, in Klassen zusammengefasst. Dabei muss der Programmierer entscheiden, inwieweit Objekte (die zur Laufzeit aus einer Klasse instanziiert werden) bzw. entsprechende Informationen sichtbar sind. Hierüber wird beeinflusst, ob die Daten vom Benutzer verändert werden können.

6.2 Die Programmiersprache C#

C# ist eine von Microsoft entwickelte Programmiersprache, die die Mächtigkeit des .NET Frameworks ausnutzt. Sie ähnelt der Programmiersprache JAVA oder auch C++, ist aber mächtiger. Die bisherige „ViLM“-Version wurde unter der Programmierumgebung Visual-Studio 2005 mit der Programmiersprache C# implementiert. Da die Visual-Studio 2005-Umgebung sehr übersichtlich gestaltet ist und große Teile des bestehenden Quellcodes lediglich angepasst werden mussten, wurde die Entwicklungsumgebung und die Programmiersprache beibehalten. C# ist durch Kenntnisse in JAVA relativ leicht verständlich und erlernbar. Bei Schwierigkeiten steht eine große Literaturliste zu diesem Thema zur Verfügung.

6.3 XML - Extensible Markup Language

XML ist eine Datenbeschreibungssprache, die vom „World Wide Web Consortium“ (W3C) entwickelt wurde. Ähnlich wie bei der Hypertext Markup (HTML)

Language wird eine Baumstruktur erzeugt. Allerdings beschränkt sich XML auf die Struktur von Daten, das Layout wird nicht berücksichtigt.

Die Struktur enthält entweder ein Start-Tag (z.B. <ROOT>), welches mit einem entsprechenden End-Tag (z.B. </ROOT>) beendet wird oder ein leeres Tag (z.B. <Materialien />). Zum Speichern von Informationen können dem Start- oder leeren Tag Attribute hinzugefügt („Attributname = Attributwert“) oder Text zwischen Start- und End-Tag gespeichert werden. Durch die einfache Syntax und klare Struktur können XML-Dateien sowohl von Menschen als auch von Maschinen gelesen werden.

Innerhalb des Projekts wird XML in beiden Bereichen „ViLM – VideoVorbereitung“ und „ViLM – Annotate“ genutzt. „ViLM – VideoVorbereitung“ erzeugt eine XML-Datei (vilm.xml), die die Anzahl der Videos, die Dateinamen der Videos, die vom Benutzer gewählte Beschreibung der Videos, sowie die Start- und Endzeitpunkte enthält. Die Datei wird folgendermaßen erzeugt:

```
//XML-Deklaration
xmlnode = xmlDoc.CreateNode(XmlNodeType.XmlDeclaration,
    "", "");
xmlDoc.InsertBefore(xmlnode, xmlDoc.DocumentElement);

//Root-Element hinzufügen
xmlRoot = xmlDoc.CreateElement("ROOT");
xmlDoc.AppendChild(xmlRoot);

xmlelem = xmlDoc.CreateElement("anzahl");
xmlelem.SetAttribute("anzahl", anzahlVideos.ToString());
xmlRoot.AppendChild(xmlelem);

int countIndex = 0;
for (int i = 0; i < aktAnzahl; i++)
{
    xmlelem = xmlDoc.CreateElement("view" + (countIndex+1));
    // nur den Dateinamen (ohne Verzeichnis) speichern
    string file = fileName[i].ToString().Substring(fileName[i]
        ].ToString().LastIndexOf("\\"));
    file = file.Substring(1);
    xmlelem.SetAttribute("File", file);
    xmlelem.SetAttribute("videoname", videoName[i].ToString()
        );
    xmlelem.SetAttribute("StartTime", startTime[countIndex].
        ToString());
}
```

```

xmlelem.SetAttribute("EndTime", endTime[countIndex].
    ToString());
xmlRoot.AppendChild(xmlelem);
countIndex = countIndex + 1;
}

try
{
    xmldoc.Save(filespec);
    MessageBox.Show("Das Projekt wurde gespeichert.");
}
catch (Exception e)
    MessageBox.Show(e.Message);

```

Listing 6.1: Speichern der synchronisierten Videodaten

Eine erzeugte XML-Datei mit zwei Videos hat dann beispielsweise folgendes Aussehen:

```

<?xml version="1.0"?>
<ROOT>
  <anzahl anzahl="2" />
  <view1 File="2007_12_06_Elsen-11.flv" videoname="Schülersicht"
    StartTime="0" EndTime="2216,6986806" />
  <view2 File="2007_12_06_Elsen-s1.flv" videoname="Lehrersicht"
    StartTime="28,2643194" EndTime="2244,963" />
</ROOT>

```

Listing 6.2: Erzeugte vilm.xml

Die Struktur der vilm.xml-Datei wurde von der bisherigen Version übernommen. Es sind allerdings kleine Veränderungen vorgenommen worden: Die Anzahl sowie die vom Benutzer gewählten Namen der Videos werden gespeichert. Um die Struktur flexibel zu halten, wurde statt „teacherView“ und „pupilView“ eine allgemeine Bezeichnung („view“ mit Nummerierung) gewählt.

Durch das Erstellen eines neuen Projekts mit „ViLM – Annotate“ werden die Informationen aus vilm.xml ausgelesen und entsprechend verwendet. Durch das Öffnen eines bestehenden Annotationsprojekts werden die Informationen über die Startzeit, Videonamen usw. aus der gewählten ANN-Datei ermittelt. Wie das Auslesen der Videodaten aus einer XML-Datei erfolgt, zeigt das folgende Beispiel:

```

xmldoc = new XmlDocument();
xmldoc.Load(filespec);
xmlelem = (XmlElement) xmldoc.GetElementsByTagName("anzahl").Item
    (0);
anzahlVideos = Convert.ToInt32(xmlelem.GetAttribute("anzahl"));

```

```

for (int i = 0; i < anzahlVideos; i++){
    xmlelem = (XmlElement)xmldoc.GetElementsByTagName("view" + (i + 1))
        .Item(0);
    fileName.Add(xmlelem.GetAttribute("File"));
    videoName.Add(xmlelem.GetAttribute("videoname"));
    startTime.Add(double.Parse(xmlelem.GetAttribute("StartTime")));
    endTime.Add(double.Parse(xmlelem.GetAttribute("EndTime")));
}

```

Listing 6.3: Auslesen einer XML-Datei

Durch die flexible Struktur der Annotationen ist das Laden von bestehenden Annotationsprojekten aufwendiger, daher wurde in dem Beispiel eine Beschränkung auf die Informationen über die Videos vorgenommen. Die von „ViLM – Annotate“ erzeugte ANN-Datei ist eigentlich eine XML-Datei, deren Endung verändert wurde. Daher kann der Zugriff auf den Inhalt in dieser Datei auf die gleiche Art erfolgen, wie bei der Speicherung der synchronisierten Videos.

6.4 MDI-Anwendung

MDI steht für „Multiple Document Interface“. MDI-Anwendungen ermöglichen es, mehrere Fenster innerhalb eines übergeordneten Fensters zu öffnen. Die sogenannten Unterfenster können innerhalb des Hauptfensters beliebig verschoben werden. Die Fenster lassen sich meistens zur besseren Übersicht anordnen.

Wie in der Design-Entscheidung beschrieben wurde, ist „ViLM“ als MDI-Anwendung umgesetzt. Da diese Technik die Definition eines Hauptfensters erfordert, muss in der Datei des Hauptfensters eine entsprechende Variable gesetzt werden. Die untergeordneten Fenster müssen mit einer entsprechenden Anweisung erzeugt werden:

```

Video_1 videoFenster = new Video_1(countVideos, false);
listVideo.Add(videoFenster);
videoFenster.MdiParent = this; //an dieser Stelle wird dem
    aktuellen Fenster das erzeugte untergeordnet
videoFenster.Show();

```

Listing 6.4: Erzeugen des ersten Videofensters bei „ViLM – Videovorbereitung“

Durch diese Einstellung kann aber noch nicht von dem untergeordneten Fenster auf das Hauptfenster zugegriffen werden. Hierzu muss innerhalb des Unterfensters ein Verweis auf das Hauptfenster erstellt werden:

```

Form1 theParent = (Form1) this.MdiParent;

```

Listing 6.5: Zugriff auf das Hauptfenster ermöglichen

Über dieses Objekt ist dann der Zugriff auf öffentliche Methoden des Hauptfensters möglich.

6.5 ffmpeg

„ffmpeg“ ist ein Programm zur Codierung von Videos in unterschiedliche Formate. Zur Auswahl stehen dem Benutzer folgende Formate, in die bestehende Videos codiert werden können: AVI, MOV, MPG, FLV, ASF, WMV. Unter Windows steht dieses Programm mit einer graphischen Benutzeroberfläche zur Verfügung. Zur Codierung in FLV-Dateien werden die Einstellungen so vorgenommen, wie es dem folgenden Bild zu entnehmen ist:

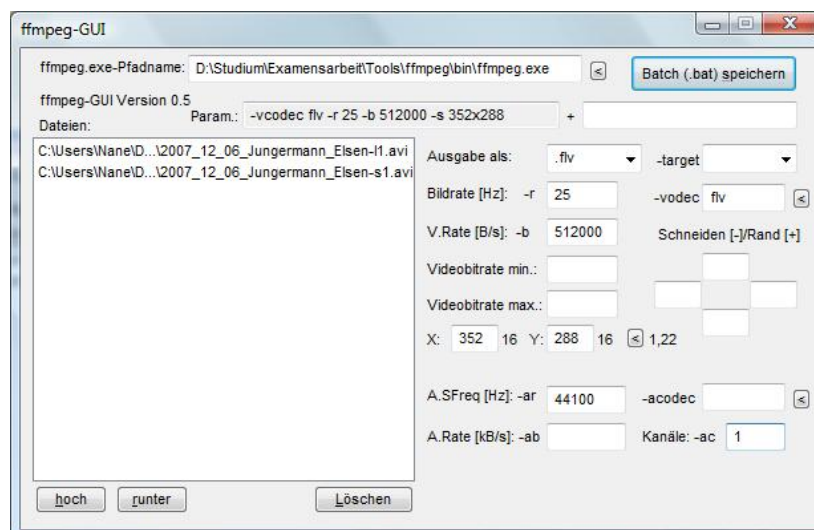


Bild 6.1: Parameter zur Codierung der FLV-Videos

Da dieses Tool Freeware ist, kann jeder Benutzer von „ViLM“ eine entsprechende Codierung seiner Videos vornehmen. Hierzu muss er lediglich die während der VideoVorbereitung erzeugte file4ffmpeg.bat-Datei in den Hauptordner von „ffmpeg“ kopieren und ausführen. Der Vorgang, der hiermit angestoßen wird, benötigt einige Zeit (abhängig von der Größe des Videos). Die erzeugte Datei wird in dem Verzeichnis des ursprünglichen Videos abgelegt.

7 Resümee

Rückblickend kann ich sagen, dass das Projekt sehr umfangreich ist. Zu Beginn hätte ich nicht geglaubt, dass die Umsetzung so viel Zeit in Anspruch nehmen würde.

Da sich während der Bearbeitung einige Schwierigkeiten auftaten, möchte ich diese an dieser Stelle kurz erwähnen. Ferner hat sich gezeigt, dass es noch Verbesserungen und mögliche Erweiterungen gibt, die in einer späteren Version aufgenommen werden können. Diese werden ebenfalls in einem nachfolgenden Abschnitt dargestellt.

7.1 Aufgetretene Schwierigkeiten

Durch die geringen Kommentare im Quellcode fiel die Orientierung anfangs schwer. Es war zwar klar, dass im Projekt Synchronisation innerhalb von Form1.cs die wichtigen Informationen zum Synchronisieren der Videos enthalten sind, aber die Benennung der Button und die entsprechende Zuordnung zu den Oberflächenelementen war schwierig. Nachdem ich mich im Quellcode etwas orientiert hatte, musste ich feststellen, dass es zu den verwendeten Klassen zur Videodarstellung und -steuerung keinen Eintrag in der Klassenbibliothek (unter .NET) gibt. Von Oliver Buschjost wurde ich dann auf entsprechende Internetseiten¹⁸ aufmerksam gemacht. Hierdurch wurden die verwendeten Klassen etwas verständlicher.

Ähnlich verlief die Einarbeitung in „ViLM – Annotate“. Allerdings ergaben sich hier Schwierigkeiten, den Bereich für die Annotationen herauszufinden. Dieser wurde, wie ich später erfuhr, in einem eigenen Formular erstellt und der Quellcode anschließend kopiert. Da Oberflächen für gewöhnlich als Formulare innerhalb von Projekten auftreten, habe ich nach einer entsprechenden Klasse gesucht. Allerdings habe ich nach einiger Zeit herausgefunden, dass dieser Bereich in der Klasse ViLMProjekt.cs implementiert ist. Die notwendigen Änderungen ließen sich dann recht schnell vornehmen.

Durch die Umsetzung der beiden Programme als MDI-Anwendungen ergaben sich einige Schwierigkeiten. Die Zugriffe auf die einzelnen Objekte vom Hauptfenster aus bzw. Anpassungen im Hauptfenster, die von einem der Unterfenster

¹⁸<http://www.codeproject.com/KB/directx/directshowmediaplayer.aspx> und <http://www.codeproject.com/KB/game/froggohop.aspx>

hervorgerufen wurden, mussten ermöglicht werden. Zu Beginn war die Zuordnung der Informationen der Unterfenster schwierig, da diese in einer Liste innerhalb des Hauptfensters vorlagen. Nachdem aber der Zugriff auf die entsprechenden Objekte über „foreach“-Schleifen bzw. über den entsprechenden Index realisiert wurde, mussten get- und set-Methoden zu einzelnen Objekten erstellt werden. Nur so konnten die entsprechenden Werte zwischen den Objekten ausgetauscht werden.

Das Abspielen einiger Videos verursachte Erstaunen: Manche Videos können auf meinen Systemen (Windows XP bzw. Windows Vista) nicht abgespielt werden. Zuerst dachte ich, es wäre ein Fehler in „ViLM“, aber auch nach der Installation eines entsprechenden Videocodec¹⁹ ergab sich keine Veränderung. Dieses Phänomen liegt aber nicht an „ViLM“, sondern scheint ein Problem des Systems zu sein, denn die entsprechenden Videos können auch nicht mit anderen Programmen unter meinen Systemen abgespielt werden.

Ein weiteres merkwürdiges Verhalten eines verwendeten Videos war, dass dieses Video Sprünge aufwies. Dies ist allerdings erst aufgefallen, nachdem die Videos synchronisiert und später in „ViLM – Annotate“ geöffnet wurden. Nach kurzer Zeit (wenige Minuten) liefen die Videos bereits nicht mehr synchron. Bei genauer Betrachtung fiel dann auf, dass in dem Video kleine Sprünge enthalten sind, die wahrscheinlich während der Aufnahme entstanden sind. Nachdem dieses Verhalten festgestellt wurde, habe ich andere Videos für das Testen der Software verwendet, die korrekt abgespielt wurden.

Beim Testen der neuen „ViLM – VideoVorbereitung“-Software habe ich bemerkt, dass es Probleme bei der Ermittlung der frühesten Startzeit der Videos gab. Zu Beginn wurde in einigen Dateien eine negative Startzeit gespeichert. Dies verursachte dann bei der Annotation eine Fehlermeldung, da kein negativer Zeitpunkt für ein Video gesetzt werden kann. Dieses Problem wurde dann aber schnell über eine zusätzlich eingebaute for-Schleife behoben.

Ferner fiel mir auf, dass bisher keine erzeugten Events oder Unterphasen abgespeichert werden. Das Speichern der Events ist jetzt zwar möglich, leider konnte ich aus Zeitgründen keine Unterphasen mehr realisieren. Dieser Aspekt sollte aber bei einer Erweiterung bzw. Anpassung der Software mit berücksichtigt werden.

7.2 Ausblick

Es hat sich gezeigt, dass es einige Eigenschaften gibt, die ich noch gern umgesetzt hätte. Eine, in meinen Augen wichtige Eigenschaft, ist die Auslagerung des Annotationsbereichs in ein eigenständiges Fenster. Durch diese Umsetzung

¹⁹Programm zum Codieren bzw. Decodieren von Videos

kann der Benutzer den Bereich ausblenden oder verkleinern, falls er nicht benötigt wird.

Ferner sollte der Bereich für die Auswahl des Tons an einer anderen Stelle positioniert werden. Derzeit werden bei langen Videonamen Teile abgeschnitten bzw. über den eigentlich nachfolgenden Radiobutton hinausgeschrieben. Vielleicht ist auch hier eine Auslagerung in ein eigenständiges Fenster sinnvoll.

Da für den Benutzer nicht direkt ersichtlich ist, was im rechten Bereich der Annotationen angezeigt wird, könnte man hier noch eine Art Beschriftung des Bereichs vornehmen. Dies ist leider nicht so einfach umzusetzen, da der Bereich nicht als Formular gestaltet ist. Somit muss man sich hier erst einmal eine sinnvolle Position überlegen und anschließend versuchen, diese im Quellcode zu realisieren.

Wie in der Beschreibung der bisherigen „ViLM“-Software bereits beschrieben, gab es einen Bereich zur Vorschau auf einige Bilder aus den Videos. Auch dieser Bereich sollte in Zukunft wieder aufgenommen werden. In der beschriebenen Variante wurde die Hälfte des Annotationsbereichs durch die Vorschau eingenommen. Denkbar wäre auch ein eigenständiges Fenster, in dem zu jedem Video entsprechende Bilder gezeigt werden. Eine dritte Variante, die ich mir vorstellen kann, ist eine Anzeige in dem Videofenster selbst. Diesen Bereich sollte man dann ein- und ausschalten bzw. die Größe verändern können. Wie die Realisierung letztendlich sinnvoll soll, ist abzuwägen.

Da für die Umwandlung der Videos „ffmpeg“ notwendig ist, wurde über ein Installationspaket von „ViLM“ gesprochen, welches „ffmpeg“ für Windows zur Verfügung stellt. Dieses Paket soll neben „ffmpeg“ natürlich auch die „ViLM“-Software installieren. Wenn eine solche Installationsmöglichkeit besteht, könnte man bei der Synchronisation der Videos die Codierung in FLV-Dateien direkt anstoßen. Unter Visual Studio sollte es möglich sein, auf einfachem Weg eine Installationsdatei (z.B. MSI-Paket) erstellen zu lassen.

Für eine weitere sinnvolle Funktionalität halte eine Hilfefunktion. Zwar sind viele Funktionen von „ViLM“ leicht verständlich, vielleicht bleibt dem Benutzer aber so ein Teil der Funktionalität verborgen. Ich denke an dieser Stelle besonders an die Umbenennung des *Kommentar*-Knoten, der standardmäßig eingefügt wird oder an die Benennung der Videos (in der VideoVorbereitung).

Derzeit ist eine vilm.xml-Datei für alle Annotationen erforderlich. Da ein einzelnes Video nicht synchronisiert werden muss, ist zu überlegen, ob es eine sinnvolle Möglichkeit gibt, diesen Schritt einzusparen.

Anhang

Abbildungsverzeichnis

2.1	Aktuelle „ViLM – Synchronisation“ Oberfläche	4
2.1	Aktuelle „ViLM – Annotate“ Oberfläche nach dem Programmstart .	6
2.2	Fenster zur Material-Verwaltung	8
2.3	Bereich für Annotationen	8
2.4	Anzeige bei Auswahl einer Phase	10
3.1	Überarbeitete „ViLM - Synchronisation“ Oberfläche	15
3.1	Neu gestaltete Oberfläche von „ViLM – Annotate“	16
3.2	Menü FENSTER	17
3.1	Kooperation von „ViLM“ mit dem neu erstellten Player	18
6.1	Parameter zur Codierung der FLV-Videos	37

Listings

6.1	Speichern der synchronisierten Videodaten	34
6.2	Erzeugte vilm.xml	35
6.3	Auslesen einer XML-Datei	35
6.4	Erzeugen des ersten Videofensters bei „ViLM – Videovorbereitung“	36
6.5	Zugriff auf das Hauptfenster ermöglichen	36

Literaturverzeichnis

- [BBM⁺06] BEER, W., D. BIRNGRUBER, H. MÖSSENBÖCK, H. PRÄHOFER und A.WÖSS: *Die .NET-Technologie. Grundlagen und Anwendungsprogrammierung*. dpunkt.verlag, Heidelberg, 2. Auflage, 2006.
- [Cod08] CODEPROJECT, 1999-2008. <http://www.codeproject.com/> (Stand: September 2008).
- [Cor08] CORPORATION, MICROSOFT: *MSDN Library (Deutsch)*, 2008. <http://msdn.microsoft.com/de-de/library/default.aspx> (Stand: September 2008).
- [Gun02] GUNNERSON, E.: *C#. Tutorial und Referenz*. Galileo Press, Bonn, 2. Auflage, 2002.
- [Hin07] HINZBERG, H.: *C# Beispiele*, 2007. <http://www.hinzberg.net/csharp/index.html> (Stand: September 2008).
- [HWG04] HEJLSBERG, A., S. WILTAMUTH und P. GOLDE: *Die C# Programmiersprache. Die komplette Referenz*. Addison-Wesley, München, 2004.
- [Kla08] KLAUS, M.: *CSharp XML - Erzeugen eines XML-Dokuments mit C#*, 2006-2008. http://www.tsq1.de/csharp/csharp_xml_erzeugen.htm (Stand: September 2008).
- [Kue08] KUEHNEL, A.: *Visual C# 2008 - Das umfassende Handbuch*, 2008. http://www.galileocomputing.de/openbook/visual_csharp/ (Stand: September 2008).
- [Mag99] MAGENHEIM, J.: *Proceedings of ED-MEDIA 99 World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Kapitel ViLM: Visualization of Learning and Teaching Strategies with Multimedia in Teacher Education, Seite 1593 f. Seattle, Washington, USA, 19.-24. Juni 1999. <http://ddi.uni-paderborn.de/fileadmin/Informatik/AG-DDI/Veroeffentlichungen/Paper/1999/edmedia99.pdf> (Stand: September 2008).

A Muster einer Zustimmungserklärung

Name

Vorname

Schule

Klasse / Kurs

Zustimmungserklärung

Ich bin damit einverstanden, dass in ausgewählten Schulstunden des Informatik-Unterrichts Video-Aufnahmen gemacht werden und diese auszugsweise (etwa 10% des anfallenden Materials) von einer geschlossenen Benutzergruppe (Studenten, Hochschulmitarbeiter und Referendare) ausschließlich zu Ausbildungs- und Forschungszwecken ausgewertet werden.

Datum und Unterschrift
der Schülerin / des Schülers

Datum und Unterschrift der / des
Erziehungsberechtigten bei
nicht volljähriger Schülerin /
nicht volljährigem Schüler

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit in der vom Landesprüfungsamt festgesetzten Bearbeitungszeit selbständig verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, habe ich in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Das Gleiche gilt auch für die Zeichnungen, Kartenskizzen und Darstellungen.

Dringenberg, den 9. Oktober 2008
