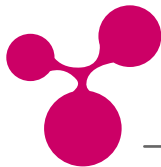


Technische Universität Dresden – Fakultät Informatik
Professur für Multimediatechnik, Privat-Dozentur für Angewandte Informatik

Prof. Dr.-Ing. Klaus Meißner
PD Dr.-Ing. habil. Martin Englien
(Hrsg.)



GENeME '09

GEMEINSCHAFTEN IN NEUEN MEDIEN

an der
Fakultät Informatik der Technischen Universität Dresden

mit Unterstützung der

3m5. Media GmbH, Dresden
GI-Regionalgruppe, Dresden
Communardo Software GmbH, Dresden
Kontext E GmbH, Dresden
Medienzentrum der TU Dresden
nubix Software-Design GmbH, Dresden
objectFab GmbH, Dresden
SALT Solutions GmbH, Dresden
Saxonia Systems AG, Dresden
T-Systems Multimedia Solutions GmbH

am 01. und 02. Oktober 2009 in Dresden

<http://www-mmt.inf.tu-dresden.de/geneme/>
geneme@mail-mmt.inf.tu-dresden.de

B.7 Eine service-orientierte Grid-Infrastruktur zur Unterstützung medienwissenschaftlicher Filmanalyse

Dominik Seiler¹, Ralph Ewerth², Steffen Heinz^{1,2}, Thilo Stadelmann²,
Markus Mühling², Bernd Freisleben², Manfred Grauer¹

¹Universität Siegen, Institut für Wirtschaftsinformatik

²Philipps-Universität Marburg, FB Mathematik und Informatik

Abstract

Innerhalb des von der DFG (SFB/FK 615) geförderten Projektes „Methoden und Werkzeuge zur rechnergestützten medienwissenschaftlichen Analyse“ werden Verfahren und Werkzeuge entwickelt, die Medienwissenschaftler bei der Analyse von audiovisuellen Daten unterstützen sollen. In diesem Beitrag wird die Videoanalyse-Software Videana vorgestellt, die eigens entwickelte Verfahren und Konzepte zur Analyse von audio-visuellen Daten beinhaltet. Weiterhin werden Forschungsarbeiten bezüglich einer Grid-Infrastruktur zur Unterstützung medienwissenschaftlicher Forschungsarbeiten sowie zur service-orientierten Analyse von Multimediadaten präsentiert. So verfügt Videana beispielsweise über eine integrierte Grid-Anbindung und erlaubt Mitgliedern einer virtuellen Organisation, rechenintensive Analyseaufgaben auf externen Rechenressourcen auszuführen.

1 Einleitung

Das Teilprojekt „Methoden und Werkzeuge zur rechnergestützten medienwissenschaftlichen Analyse“ des von der Deutschen Forschungsgemeinschaft geförderten kulturwissenschaftlichen Forschungskollegs SFB/FK 615 „Medienumbrüche“ entwickelt und erforscht unter anderem Verfahren zur medienwissenschaftlichen Analyse digitaler Bild- und Videodaten, die in dem Video- und Filmanalysewerkzeug *Videana* [5] integriert sind. Diese Verfahren dienen der automatischen Erschließung und Erstellung von deskriptiven Metadaten und Annotationen für Videomaterial. Sie sollen die medienwissenschaftliche Filmanalyse durch rechnergestützte Verfahren quantitativ unterstützen und den ansonsten sehr zeitaufwändigen und teuren Prozess der manuellen Annotation von einzelnen Kameraeinstellungen weitgehend ersetzen. Die qualitative Analyse und Interpretation von Filmen wird jedoch auf absehbare Zeit ausschließlich Menschen vorbehalten sein. Nichtsdestotrotz können Rechner den Menschen bei einigen typischerweise sehr zeitraubenden Prozessen unterstützen bzw. entlasten. Diese Entlastung steht jedoch in einem engen Zusammenhang mit dem steigenden Ressourcenbedarf der verwendeten

Computer. Ein weiterer Teil unserer Forschungsarbeiten erstreckt sich daher auf die Auslagerung rechenintensiver Analysen auf freie Rechenressourcen eines Grids, das einer Gemeinschaft von virtuellen Organisationen zur Verfügung steht. Hierzu werden in diesem Beitrag eigene Lösungen vorgestellt: (a) zur Vereinfachung des Job-Schedulings in einem Grid, (b) zur Übertragung von großen Datenmengen in einer service-orientierten Grid-Umgebung, und (c) zur einfachen Komposition von Workflows zur Medienanalyse. Die Nutzung ausgelagerter Dienste durch den Endnutzer in Form von Web Services ist ein weiterer Aspekt unserer Forschung; hierzu wird der von uns entwickelte Mashup Editor präsentiert.

Der Beitrag ist wie folgt strukturiert. In Abschnitt 2 wird zunächst *Videana* und dessen Grid-Anbindung vorgestellt. Abschnitt 3 behandelt die auf den Kernalgorithmen von *Videana* beruhenden Analyseservices, sowie deren Komposition zu komplexen Analyseabläufen. Außerdem wird der sich noch in Entwicklung befindende Mashup Editor vorgestellt. Schließlich wird in Abschnitt 4 der Beitrag zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben.

2 Videana

In diesem Abschnitt wird das Video- und Filmanalysewerkzeug *Videana* kurz vorgestellt. Ein großer Vorteil der Rechnerunterstützung ist in der Automatisierung formaler, zeitintensiver Analyseschritte zu sehen. Zu nennen ist etwa die zeitliche Segmentierung eines Videos in Kameraeinstellungen und hierbei die Identifikation der Montageart (harter Schnitt, Aufblende, Abblende, Überblende etc.), das Finden und Erkennen von eingeblendetem Text, das Erkennen von Objekt- und Kamerabewegung (horizontale und vertikale Kamerabewegung durch Translation oder Rotation der Kamera sowie Zoom), das Erkennen der verwendeten Einstellungsgröße, Informationen über die Präsenz der Akteure, die Art der auditiven Signale etc. Die graphische Benutzerschnittstelle von *Videana* ermöglicht es, Videos anzuzeigen, abzuspielen und bildgenau an eine bestimmte Position des Videos zu gelangen (siehe Abb. 1).

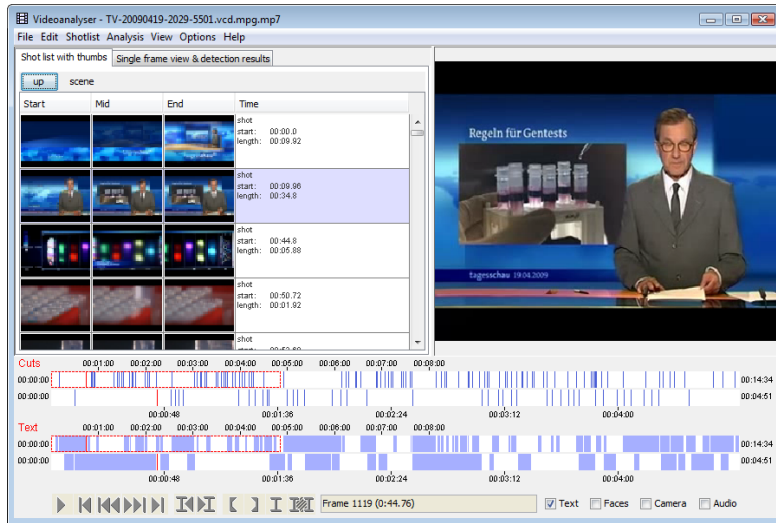


Abbildung 1: Screenshot des Videana-Hauptfensters

Weiterhin wurden Algorithmen zur Schnittdetektion [4], Textdetektion und -segmentierung (notwendig für eine anschließende Zeichenerkennung mit einer „Optical Character Recognition“ (OCR) Software, [8]), Gesichtsdetektion [14] und Gesichtserkennung und der Bestimmung der Kamerabewegung [6] integriert, die mittels eines Plug-In-Konzepts auf einfache Art und Weise ergänzt, entfernt oder ausgetauscht werden können. Automatisch erzeugte Analyseergebnisse können jederzeit von Benutzerseite manuell korrigiert werden. Bei den Verfahren aus [4], [6] und [8] handelt es sich um Eigenentwicklungen des Teilprojektes.

Die Ergebnisse der verschiedenen Detektoren werden auf einer jeweils eigenen Zeitleiste visualisiert. Einzelne Kameraeinstellungen, Ereignisse wie graduelle Übergänge und Text- oder Gesichtsobjekte können mit beliebigen Kommentaren und mit Schlüsselwörtern annotiert werden. Sowohl die extrahierten Multimediametadaten als auch die während des medienwissenschaftlichen Arbeitsprozesses manuell ergänzten Kommentare werden dem MPEG-7 Standard [12] konform gespeichert. Der MPEG-7 Standard formalisiert die Darstellung solcher Daten und ermöglicht den Datenaustausch zwischen unterschiedlichen Anwendungen auf Basis von XML-Dokumenten.

2.1 Grid-Anbindung

Grid Computing [7] konnte sich in den letzten Jahren zunehmend als eine der wesentlichen Technologien etablieren, um Mitgliedern einer virtuellen Organisation (VO) einen einheitlichen Zugriff auf unterschiedliche, verteilte Rechner- und Datenressourcen zu erlauben.

Das vorrangige Ziel der Grid-Anbindung in *Videana* ist es, die rechenintensiven audiovisuellen Analyse-Jobs der medienwissenschaftlichen Nutzer auf externe Ressourcen auszulagern und ihnen den Zugang zu High-Performance Computern und Clustern zu ermöglichen. Zudem soll dem Anwender eine möglichst einfach zu handhabende Schnittstelle zum Grid geboten werden, die sich nahtlos in die Arbeitsumgebung *Videana* integriert.

Die Anbindung basiert auf unseren Forschungsarbeiten [10] zu Omnivore, einem dezentralen System zum Job-Scheduling, welches auf einem Peer-to-Peer (P2P) basierten Meta-Scheduler [1], genannt PPM, aufbaut. Omnivore arbeitet nahtlos mit dem GridWay Meta-Scheduler zusammen und ist insbesondere in die Grid-basierte Systemlandschaft des Globus Toolkit 4 (GT4) integriert. Dieses System bietet auf Basis des verwendeten P2P-Ansatzes eine skalierbare, ausfallsichere und effiziente Umgebung zur Jobausführung. Das dabei entstehende Netzwerk pflegt und repariert sich selbst, ohne dass administrative Schritte an den einzelnen Knoten notwendig sind. Während mit GridWay Jobs über die Grid-Middleware bekannter Cluster auf diesen ausgeführt werden, kann mit Omnivore zusätzlich auch auf Cluster zurückgegriffen werden, die nicht über eine Grid-Middleware verfügen. Dabei wird der lokale Scheduler des Hauptknotens direkt genutzt. Durch die Verwendung des PPM kann Omnivore sowohl mehrere Cluster als auch Desktops in einem so genannten Omnivore-Pool zusammenfassen und verwalten. Ein solcher Pool kann dabei dynamisch wachsen und schrumpfen.

Die Omnivore-Architektur selbst setzt sich aus drei Komponenten zusammen, die alle auf dessen P2P-System aufsetzen. Die *Informationskomponente* dient dem Sammeln von Informationen über das Gesamtsystem, z.B. wie viele Knoten gerade zur Verfügung stehen. Um die Daten zu beziehen und die Kommunikationskosten zu reduzieren, werden die Knoten in einer dynamischen Baumstruktur verwaltet. Die gesammelten Informationen werden dem Scheduling-Algorithmus von GridWay dann als Entscheidungsbasis weitergereicht. Die *Speicherkomponente* speichert sowohl Daten, die Omnivore selbst zum Betrieb benötigt, als auch Daten für die eigentliche Jobausführung, d.h. Ein- und Ausgabedateien. Alle Daten werden redundant im Netzwerk gespeichert und dezentral verwaltet. Schließlich behandelt die *Ausführungskomponente* die Jobbeschreibung und die Weiterleitung der Jobs an den PPM, der wiederum auf verschiedene Scheduler der darunterliegenden Grid-Middleware bzw. auf lokale Scheduler, wie Fork, LSF, SGE oder PBS, zur eigentlichen Jobausführung zurückgreift.

Jede *Videana*-Distribution verfügt über einen Client, der den Meta-Scheduler über eine Service-Schnittstelle anspricht. Dieser ist nahtlos in *Videana* integriert und lässt sich über einen einfachen Konfigurationsdialog aktivieren. Innerhalb dieses Dialogs wird der Benutzer aufgefordert ein sogenanntes Proxy-Zertifikat zu erstellen. Hierzu benötigt er ein gültiges Grid-Zertifikat, welches ihn als legitimes Mitglied einer VO ausweist. Zudem kann der Benutzer festlegen, ob jede gestartete Analyse auf einen externen Knoten ausgelagert wird bzw. kann er einen Schwellenwert für die Dateigröße festlegen, ab der die Verarbeitung ausgelagert wird.

2.2 Architektur-Überblick

Abbildung 2 gibt einen Überblick über die Grid-Architektur von *Videana*. Mittels des *Videana*-Systems in der *Interaktionsschicht* kann der Benutzer an die darunter liegende *Middlewareschicht* Analysejobs absetzen. Die *Middlewareschicht* ist in zwei Ebene unterteilt. Auf der ersten Ebene befinden sich GridWay und Omnivore, die sich um die Weiterleitung und die Verwaltung der Knoten im gesamten Grid kümmern. Darunter befindet sich als Kern der Grid-Middleware das GT4, welches sich unter anderem um die eigentliche Jobausführung kümmert. In der letzten Schicht, der *Infrastrukturschicht*, befinden sich die eigentlichen Rechenressourcen in Form von Clustern bzw. Omnivore-Pools.

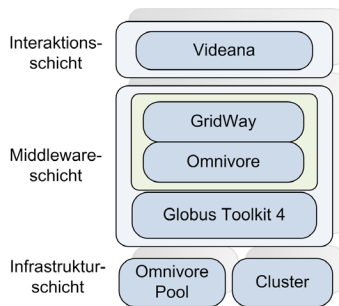


Abbildung 2: Die Videana Grid-Architektur

3 Eine service-orientierte Architektur für die Multimedia-Analyse

Ein weiterer Teil der Forschungsarbeiten des Projektes beschäftigt sich mit der Entwicklung von Werkzeugen und Web Services für eine service-orientierte Architektur (SOA), die speziell auf die Anforderungen der multimedialen Analyse abgestimmt ist. Abbildung 3 zeigt die grundlegende Architektur der vorgeschlagenen service-orientierten Architektur für die Analyse von Multimediadaten. Sie besteht aus vier Schichten: Die *Interaktionsschicht* stellt die Schnittstelle zur Kommunikation mit der SOA da. Ein Entwickler kann mit Hilfe eines BPEL (Business Process

Execution Language) Workflow Editors auf Basis existierender Web Services und Workflows aus der entsprechenden Schicht neue Workflows definieren und diese als Web Service exponieren. Ein service-unterstützender Mashup Editor, der ebenfalls in dieser Schicht platziert ist, erlaubt es, Web Service-Nutzern neue Anwendungen als Kombination einer gegebenen Menge von vordefinierten Services (wie etwa ein Service zur Detektion von Gesichtern) und Webanwendungen (wie z.B. Flickr oder YouTube) zu erstellen. Die *Service- und Workflowschicht* besteht aus einer Menge von Workflows und Services, die hierarchisch angeordnet sein können. In der *Middleware-schicht* befinden sich sowohl der Apache Tomcat als auch das Globus Toolkit 4 als Laufzeitumgebung für die Workflows bzw. Services. Beide nutzen für den Datentransport die Flex-SwA-Plattform [11]. Flex-SwA wurde entwickelt, um in service-orientierten Umgebungen einen effizienten Transport von großen Datenmengen zu ermöglichen. Abschließend folgt die *Infrastrukturschicht*, die die eigentlichen Rechenressourcen bereitstellt.

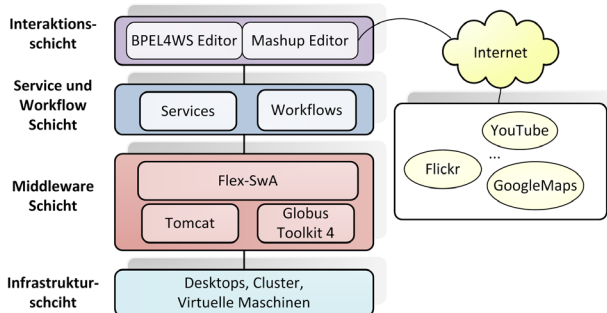


Abbildung 3: Service-orientierte Architektur für die Multimedia-Analyse

Mit Hilfe der service-orientierten Architektur für die Analyse von Multimediadaten sollen Anwender in die Lage versetzt werden, eigene Analyseverfahren auf Basis bereits existierender „Bausteine“ zu erstellen und zu deren Ausführung auf externe Rechenressourcen zurückgreifen zu können. Eine (Re-)Integration der neu generierten Analyseverfahren in *Videana* soll in Zukunft ebenfalls möglich sein. Die folgenden Unterabschnitte stellen die wichtigsten Komponenten der service-orientierten Architektur für die Multimedia-Analyse vor.

3.1 Datentransport mit Flex-SwA

Unsere Entwicklung von Flex-SwA [11, 13] gestattet es, große Datenmengen in einer service-orientierten Umgebung flexibel zu handhaben. Anstatt die Nutzdaten, wie im SOA-Umfeld üblich, direkt bei einem Serviceaufruf in einer SOAP-Nachricht

mitzuschicken, wird nur eine Referenz auf diese an den aufzurufenden Service gesendet. Dabei kann es sich sowohl um lokal vorliegende Daten als auch um Daten handeln, die sich auf einem entfernten Datenserver befinden. Die Flex-SwA-Architektur kümmert sich um den eigentlichen Datentransport von der angegebenen Datenquelle zum Ziel und ermöglicht damit einen direkten Transport, der mittels verschiedener Protokolle, wie z.B. TCP, FTP oder GridFTP, realisiert werden kann. Zusätzlich werden verschiedene Kommunikationsmuster unterstützt, die den Zeitpunkt der eigentlichen Datenübertragung festlegen und eine Überlappung von Serviceausführung und Datentransport erlauben. Die Verzahnung von Übertragung- und Ausführungszeit ermöglicht eine weitere Optimierung der Gesamtlaufzeit des Service.

3.2 Grid Services für die Analyse von Multimediatdaten

Parallel zur integrierten Grid-Anbindung von *Videana* wurden die wesentlichen Analysealgorithmen als Grid Services exponiert. Im Gegensatz zu Web Services sind diese zustandsbehaftet und verwalten ihren Zustand entsprechend des „Web Service Resource Framework“ (WSRF, [9]). Dadurch eignen sie sich besonders für langlaufende Prozesse, wie dies bei der Analyse von multimedialen Daten der Fall sein kann. Momentan verfügbare Grid Services sind unter anderem:

- Decoder: Der Decoderservice dekomprimiert MPEG-1 und MPEG-2 Videos und liefert als Ausgabe einzelne Bilder.
- Video Splitter: Dieser Service dient der Zerlegung von MPEG-1 und MPEG-2 Videos in einzelne Sequenzen.
- MPEG-7 Konverter: Der MPEG-7 Konverter sammelt während einer Analyse die Teilergebnisse und liefert als Ausgabe eine MPEG-7 Datei.
- Schnitt-Detektor: Mit der Detektion der Einstellungswechsel („Schnitte“, [4]) wird ein Video in seine grundlegenden Einheiten, die Einstellungen („shots“), unterteilt. Viele andere Analyseverfahren setzen eine solche zeitliche Segmentierung des Videos voraus.
- Kamerabewegungsdetektor: Service, der die Kamerabewegungen in einem Video detektiert [6].
- Textdetektor: Dieser Service implementiert das Verfahren, um eingeblendeten Text in Videos zu finden [8].
- Textsegmentierung: Dieser Service nutzt die Ergebnisse des Textdetektors, um Schwarz-/Weiß-Bilder zu erzeugen, die nach Möglichkeit nur noch den Text auf einem einheitlichen Hintergrund enthalten [8]. Dies ist wichtig für eine nachfolgende OCR-Verarbeitung.
- Gesichtsdetektor: Der Gesichtsdetektor basiert auf der Implementierung der Open Source Computer Vision Library (OpenCV: <http://www.intel.com/technology/computing/opencv>) von Intel und erkennt frontal zu sehende Gesichter in Videos.

Der Gesichtsdetektor ist die einzige Komponente, die nicht auf einem eigens entwickelten Ansatz basiert. Alle Services verfügen über eine Schnittstelle die eine „Streaming“-orientierte Verarbeitung von Videos mit Hilfe von Flex-SwA umsetzen [13].

3.3 Workflow Editor

Um auf Basis der vorgestellten Grid Services komplexe Workflows entsprechend des BPEL-Standards auf einfache Art generieren zu können, wurde im Rahmen unseres Projekts der Visual Grid Orchestrator (ViGO) entwickelt [3]. Zur Ausführung dieser Workflows kann eine um WSRF-Funktionen angereicherte BPEL-Engine verwendet werden.

Innerhalb eines Workflows erfolgt die Kommunikation zwischen den einzelnen Services üblicherweise über den Knoten, der die ausführende BPEL-Engine bereitstellt. Dieser stellt somit einen Flaschenhals für die Kommunikation dar, über den alle Daten fließen. Dies ist insbesondere in dem hier betrachteten Anwendungsfall mit großen Datenmengen kritisch. Wir konnten unter anderem zeigen [13], dass dieser Flaschenhals im Datenfluss mit Hilfe von Flex-SwA entschärft werden kann, indem die Daten von Service zu Service ausgetauscht werden. In einer aktuellen Arbeit [2] wurde zudem dieses Workflowsystem um Komponenten zur Lastverteilung auf freie Ressourcen sowie zur dynamischen Akquirierung von zusätzlichen Rechenressourcen innerhalb der Elastic Compute Cloud (EC2) von Amazon in Zeiten hoher Auslastung erweitert.

3.4 Mashup Editor

Aktuelle Arbeiten unseres Projektes befassen sich mit der Entwicklung eines Mashup Editors, dessen Konzept und aktueller Stand hier kurz vorgestellt wird. Mashup Editoren wie etwa Yahoo Pipes bieten Anwendern webbasierte Schnittstellen, mittels deren Hilfe sie auf Daten von verschiedenen populären Webanwendungen wie z.B. Flickr oder Google Maps zugreifen können, um aus deren Kombination eigene Webanwendungen zu erzeugen. Ziel des hier vorgestellten Mashup Editors ist es darüber hinaus, eine einfach zu bedienende webbasierte Möglichkeit zu schaffen, mit deren Hilfe auf leistungsstarke Rechenressourcen zugegriffen werden kann. Zu diesem Zweck unterstützt der Mashup Editor Aufrufe von Web und Grid Services. Damit ist er in der Lage, die Vorteile von Grid-Ressourcen zu nutzen. Der Editor wird über den Browser des Benutzers von einem Server geladen, der diesen als Rich Internet Application (RIA) bereitstellt. Zusätzlich bietet dieser Server einen Proxy an, der serverseitige Aufgaben, wie Aufrufe von Web Services, Zugriffe auf Datenbanken und die Verwaltung von Grid- bzw. Proxy-Zertifikaten übernimmt.

Der Benutzer arbeitet im Editor auf Basis von Komponenten. Dabei handelt es sich um flexible grafische Bausteine, die sich jeweils durch Datenschnittstellen

für Ein- und Ausgabe sowie der eigentlichen Anwendungslogik definieren. Die Anwendungslogik kann dabei lokal oder entfernt, z.B. in Form eines Web Services, vorliegen. Der Editor unterscheidet zwei Ansichten: die Entwickler- und die Benutzersicht. Dieses Merkmal grenzt ihn ebenfalls von anderen Editoren ab, die oftmals eine solche Unterscheidung nicht berücksichtigen. Innerhalb der Benutzersicht kann der Anwender neue Anwendungen auf Basis existierender zusammensetzen. Die Entwickleransicht dient der Erzeugung neuer Komponenten. Dabei kann eine Anwendung, die innerhalb der Benutzersicht erzeugt wurde, zur späteren Wiederverwendung als grafische Komponente gespeichert werden. Eine erste prototypische Umsetzung der grafischen Schnittstelle des Mashup Editors wurde mit Hilfe des Flex 3 SDK von Adobe umgesetzt.

Im Gegensatz zur Orchestrierung von Workflows auf Basis von BPEL zielt das Mashup-Konzept vor allem auf die Unterstützung technisch unerfahrener Nutzer ab. Hierzu gibt es eine breitgefächerte Menge an Werkzeugen zur Datenintegration und zur Erstellung von Webanwendungen. Dieser Ansatz bietet zu diesem Zweck in Bezug auf die Erzeugung neuer Dienste eine nicht so generische Lösung wie dies etwa BPEL bietet, vermeidet dabei aber weitgehend die Nutzungskomplexität von BPEL-Workflows.

4 Zusammenfassung

In diesem Beitrag wurden Arbeiten vorgestellt, die einer Gruppe von medienwissenschaftlich arbeitenden Personen einerseits Verfahren zur inhaltlichen Analyse von audiovisuellen Daten bereitstellen und ihnen andererseits einen einfachen Zugang zu externen und/oder leistungsstarken Rechenressourcen ermöglichen. Dazu wurde zunächst das Video- und Filmanalysewerkzeug *Videana* und dessen Grid-Anbindung präsentiert. Weiterhin wurde das Modell einer von uns vorgeschlagenen und entwickelten service-orientierten Architektur für die Analyse von Multimediadaten dargestellt. Dabei wurden einzelne Analyseservices, der Datentransport innerhalb der SOA, ein Workflow Editor sowie ein sich noch in der Entwicklung befindender Mashup Editor zur Komposition der Analyseservices präsentiert.

Es gibt mehrere Themen für zukünftige Arbeiten. Zunächst steht noch die Fertigstellung des Prototyps des vorgeschlagenen Mashup Editors aus. Dabei sollen Komponenten entwickelt werden, die ein filmanalytisches Arbeiten erlauben, so wie es gegenwärtig durch *Videana* ermöglicht wird. Zusätzlich soll eine Reintegration von definierten Mashups bzw. Workflows in *Videana* ermöglicht werden. Weiterhin werden sich zukünftige Arbeiten mit dem Einsatz von virtuellen Ressourcen in dem hier vorgestellten Kontext sowie mit deren nahtloser Integration in die vorhandenen Werkzeuge beschäftigen.

Danksagung

Diese Arbeit wurde finanziell von der Deutschen Forschungsgemeinschaft (SFB/FK 615) unterstützt.

Literatur

- [1] Dörnemann, K., Prenzer, J., Freisleben, B., A Peer-to-Peer Meta-Scheduler for Service-Oriented Grid Environments, Proc. of First Int. Conference on Networks for Grid Applications, 1–8, 2007
- [2] Dörnemann, T., Juhnke, E., Freisleben, B., On-Demand Resource Provisioning for BPEL Workflows Using Amazon's Elastic Compute Cloud, Proc. of the 9th IEEE/ACM Int. Symposium on Cluster Computing and the Grid (CCGrid '09), 140–147, 2009
- [3] Dörnemann, T., Friese, T., Herdt, S., Juhnke, E., Freisleben, B., Grid Workflow Modelling Using Grid-Specific BPEL Extensions, Proc. of German e-Science Conference (GES), 1–8, 2007
- [4] Ewerth, R., Freisleben, B., Video Cut Detection without Thresholds, Proc. of the 11th Int. Workshop on Systems, Signals and Image Processing, 227–230, 2004
- [5] Ewerth, R., Mühlh, M., Stadelmann, T., Gllavata, J., Grauer, M., Freisleben, B., Videana: A Software Tool for Scientific Film Studies, Digital Tools in Media Studies, Transcript Verlag Bielefeld, 101–116, 2008
- [6] Ewerth, R., Schwalb, M., Tessmann, P., Freisleben, B., Estimation of Arbitrary Camera Motion in MPEG Videos, Proc. of 17th Int. Conf. on Pattern Recognition, Vol. 1, 512–515, 2004
- [7] Foster, I., Kesselman, C., The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers Inc., 2003
- [8] Gllavata, J., Ewerth, R., Freisleben, B., A Text Detection, Localization and Segmentation System for OCR in Images, Proc. of the 6th IEEE Int. Symposium on Multimedia Software Engineering, 310–317, 2004
- [9] Graham, S., Karmarkar, A., Mischkinsky, J., Robinson, I., Sedukhin, I., Treadwell, J., Srinivasan, L., Banks, T., Maguire, T., Snelling, D., Liu, L. and Meder, Web Services Resource Framework (WSRF), 2006, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
- [10] Heidt, M., Dörnemann, T., Dörnemann, K., Freisleben, B., Omnivore: Integration of Grid Meta-Scheduling and Peer-to-Peer Technologies, Proc. of the 8th IEEE Int. Symposium on Cluster Computing and the Grid (CCGrid '08), 316–323, 2008
- [11] Heinzl, S., Mathes, M., Friese, T., Smith, M., Freisleben, B., Flex-SwA: Flexible Exchange of Binary Data Based on SOAP Messages with Attachments, Proc. of the IEEE Int. Conf. on Web Services, 3–10, 2006

-
- [12] Martinez, Jose M., MPEG-7 Overview, Technical Report N4980, ISO/IEC JTC1/SC29/WG11, Klagenfurt 2002
 - [13] Seiler, D., Heinzl, S., Juhnke, E., Ewerth, R., Grauer, M. and Freisleben, B., Efficient Data Transmission in Service Workflows for Distributed Video Content Analysis, Proc. of the 6th Int. Conf. on Advances in Mobile Computing & Multimedia, 7–14, 2008
 - [14] Viola, P., Jones, M.J., Robust Real-Time Face Detection, Int. J. Comput. Vision, 57(2), 137–154, 2004