



Starting Guide

RELEASE 1.3

13. Juni 2006

Frank Lützenkirchen (Essen/Duisburg)

Jens Kupferschmidt (Leipzig)

Detlev Degenhardt (Freiburg)

Kathleen Krebs (Hamburg)

Vorwort

An deutschen Universitäten werden zunehmend multimediale Daten sowohl für die Forschung als auch zur Unterstützung der Lehre in elektronischer Form erstellt. Beispiele hierfür sind Digitalisate von historischen Unikaten wie altertümlichen Handschriften, Partituren, Papyri oder Münzen aber auch Lehrvideos, Animationen und Simulationen. Diese Bestände müssen in einer geeigneten Umgebung systematisch erschlossen und auch langfristig zugänglich gemacht werden, eine Aufgabe, die von einer digitalen Bibliothek geleistet werden kann. Dabei werden die multimedialen Daten mit Metadaten versehen, um ein späteres Auffinden oder Zusammenstellen bestimmter Dokumente zu erleichtern.

In diesem Kontext entstand auch Ende 1997 an der Universität Essen das Projekt MILESS¹ (Multimedialer Lehr- und Lernserver Essen). Dieses Open Source Projekt wurde zu einer solchen Reife geführt, dass andere Universitäten, die vor ähnlichen Aufgaben standen dieses Produkt nachgenutzt haben. Aus dem Kreis der Nachnutzer hat sich die ‚MILESS Community‘ entwickelt, die aktiv Erfahrungen austauscht und zusammenarbeitet. Mit der Nachnutzung haben sich auch viele Flexibilisierungswünsche und zusätzliche Anforderungen ergeben, insbesondere im Bereich von digitalen Sammlungen und Archivlösungen. MILESS war ursprünglich nur auf die lokalen Bedürfnisse in Essen zugeschnitten, so dass eine Erweiterung und Flexibilisierung des Metadatenmodells und die Anpassbarkeit des User Interfaces erforderlich wurden.

Aus diesen Anforderungen heraus entstand Anfang 2001 das MyCoRe-Projekt². Ziel von MyCoRe ist es, aufbauend auf den MILESS-Erfahrungen gemeinsam ein neues System als Softwarebasis lokaler digitaler Bibliotheken zu entwickeln. Das Präfix ‚My‘ steht dabei für die Ursprünge in der MILESS Community, das ‚CoRe‘ für Content Repository oder auch für Kern (Core). Die MyCoRe-Initiative ist für alle Interessierten offen und hat sich so organisiert, dass einzelne Mitglieder die Entwicklung bestimmter Funktionsbereiche übernehmen, für die sie die Analyse, das Design und die Implementierung durchführen. Ein Architektur-Board von 4-5 Personen koordiniert dabei die gemeinsame Entwicklung, legt Standards, Richtlinien und Schnittstellen fest, sammelt die Anforderungen aller Mitglieder der Gemeinschaft und stellt die Integrationsfähigkeit der einzelnen Komponenten sicher.

Das vorliegende Dokument gibt eine grundsätzliche Einführung in MyCoRe. Es richtet sich an alle, die sich einen Überblick über die Funktionalität und den Leistungsumfang des MyCoRe-Softwaresystems verschaffen wollen. Leserinnen und Leser werden darüber hinaus über die grundsätzlichen Hard- und Softwarevoraussetzungen sowie über die Nutzungs- und Lizenzbedingungen informiert. Auch werden sie erfahren, welche Funktionalitäten für MyCoRe zukünftig geplant sind und in welchem Zeitrahmen die Entwicklungsschritte zu erwarten sind. Eine Liste von Ansprechpartnern und Mitwirkenden am Projekt schließt diese Dokumentation ab.

Eine ausführliche Dokumentation über die Installation des Softwaresystems sowie über die Installation und Konfiguration einer auf MyCoRe basierenden

¹<http://miless.uni-essen.de>

²<http://www.mycore.de>

Beispielanwendung DocPortal mit einem Dublin Core Datenmodell findet sich in dem *MyCoRe User Guide*. Kern- als auch Anwendungsentwicklerinnen und -entwickler finden sehr detaillierte Informationen zu der verwendeten Softwarearchitektur und zu den einzelnen Komponenten im *MyCoRe Programmer Guide*. Weiterhin finden Sie im Dokumentationsverzeichnis des MyCoRe-Kernes noch weitere Schriften zu speziellen Themen wie IBM Content Manager. Das DocPortal Quick Installation Guide beschreibt, wie Sie in 15 min. zu einer ersten DocPortal-Installation gelangen.

Der besseren Lesbarkeit halber wird in diesem Text hauptsächlich die männliche Form verwendet (z.B. ‚Benutzer‘ anstelle von ‚Benutzerinnen und Benutzer‘). Weibliche Personen sind aber selbstverständlich in gleicher Weise mit angesprochen.

Inhaltsverzeichnis

1	Leistungsumfang von MyCoRe.....	1
1.1	Allgemeines.....	1
1.2	Hierarchisches Klassifikationssystem.....	2
1.3	Metadaten und Suche.....	3
1.3.1	Das MyCoRe-Datenmodell.....	3
1.3.2	Suche.....	4
1.3.3	Volltextindizierung und Suche.....	4
1.3.4	Vererbung.....	5
1.4	Internes Dateispeicher-System.....	5
1.5	Verteilte Suche und Schnittstellen.....	6
1.5.1	Verteilte Suche.....	6
1.5.2	Schnittstellen zu OAI, Z39.50, Webservices.....	6
1.6	Benutzer- und Zugriffsrechteverwaltung.....	7
1.7	Datenpräsentation.....	7
1.8	Autoreninterface.....	7
1.9	Zusätzliche Komponenten.....	8
2	Hard- und Software-Voraussetzungen.....	9
3	Nutzungs- und Lizenzbedingungen.....	10
4	Roadmap.....	11
4.1	Funktionsumfang des Releases 1.3 – 8. Juni 2006.....	11
4.1.1	Klassifikationen.....	11
4.1.2	Metadaten-Modell.....	11
4.1.3	Internes File System.....	12
4.1.4	Benutzer- und Rechtesystem.....	12
4.1.5	Datenpräsentation.....	13
4.1.6	Backend.....	13
4.1.7	Autorenwerkzeug.....	13
4.1.8	Remotezugriff und Services.....	14
4.1.9	zusätzliche Funktionalitäten.....	14
4.1.10	Beispielanwendung.....	14
4.1.11	Dokumentation.....	14
4.2	Funktionsumfang des Releases 1.4 (Ende 2006).....	14
4.2.1	Allgemeines.....	14
4.2.2	Metadaten-Modell.....	15
4.2.3	Internes File System.....	15
4.2.4	Benutzer- und Rechtesystem.....	15
4.2.5	Datenpräsentation.....	15
4.2.6	Backend.....	15
4.2.7	Beispielanwendung.....	15
4.2.8	Dokumentation.....	15
4.2.9	Sonstiges.....	15
5	Mitwirkende am Projekt und Ansprechpartner.....	16
5.1	Architekturboard.....	16
5.2	Projektmitwirkende.....	16
5.3	Ansprechpartner.....	17

6	Anhang.....	19
6.1	Abbildungsverzeichnis.....	19
6.2	Tabellenverzeichnis.....	19

1 Leistungsumfang von MyCoRe

1.1 Allgemeines

MyCoRe ist ein Kernsystem, das alle Grundfunktionen von digitalen Bibliotheken und Sammlungen abdeckt. Es kann als Basis für eine eigene Bibliotheksanwendung, aber auch für die Implementierungen einer ganz anderen Applikation, wie zur Verwaltung multimedialer Objekte verwendet werden. Dabei sollen eigene spezialisierte Lösungen insbesondere hinsichtlich der Datenmodellierung relativ schnell und einfach realisiert werden können. MyCoRe gestattet auch die Kopplung einzelner Installationen zu einem Verbund, was die Attraktivität erheblich erhöht. Beispiele dafür sind u. a. das DocPortal oder das Papyrus-Projekt der Universitäten Halle, Jena und Leipzig³.

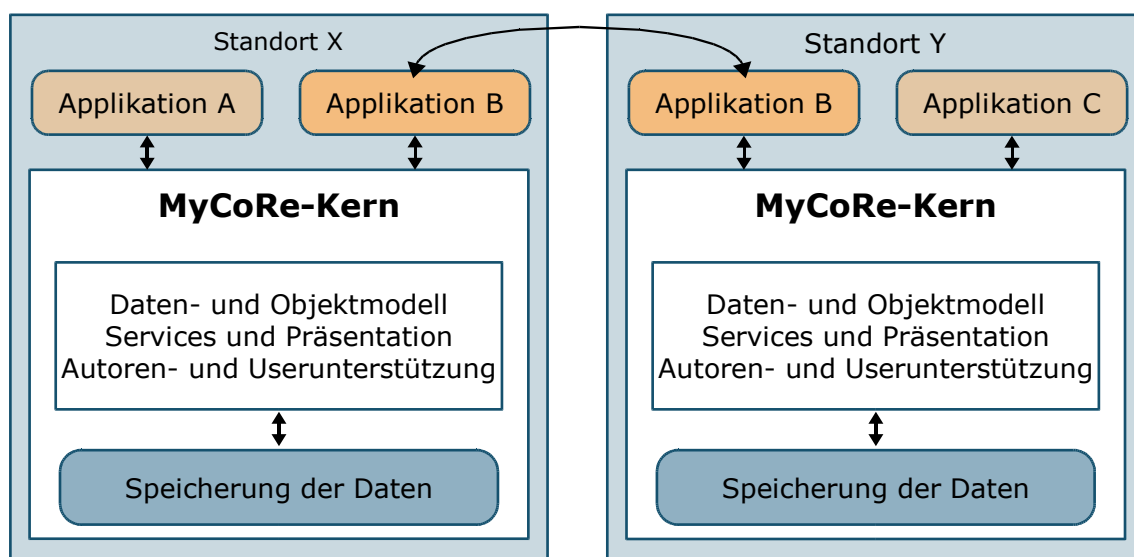


Abbildung 1.1: Grobes Schichtenmodell von MyCoRe

Auf Abbildung 1.1 ist dargestellt, wie MyCoRe-Anwendungen innerhalb eines Betreibers und aus der Sicht mehrerer Betreiber angelegt sind. In einer Einrichtung X gibt es einen MyCoRe-Kern pro Server, auf welchen mehrere Applikationen aufsetzen können. Ebenso können mehrere Anwendungen auf einem System der Einrichtung Y existieren. Dabei ist es auch möglich, dass mehrere physische Anwendungen eine gemeinsame logische Anwendung (virtuelle Datenbasis) darstellen. Dies ermöglicht gemeinsame Projekte verschiedener Einrichtungen auf der gleichen Basis.

Das MyCoRe-Projekt gliedert sich zum einen in den Kern, welcher grundlegende Funktionalitäten und eine Programmierschnittstelle (API) bereitstellt und in die darauf aufsetzenden Anwendungen, welche neben einer eigenen Präsentation auch die Kern-Komponenten erweitern und/oder umgestalten können. Somit ist gewährleistet, dass MyCoRe in vielen Projekten als Datenbasis eingesetzt werden kann. Abbildung 1.2 zeigt die wesentlichen Hauptkomponenten von MyCoRe und deren Zusammengehörigkeit. In den folgenden Abschnitten sollen dann die Leistungen der einzelnen Projektbereiche näher beschrieben werden.

³<http://papyri.uni-leipzig.de>

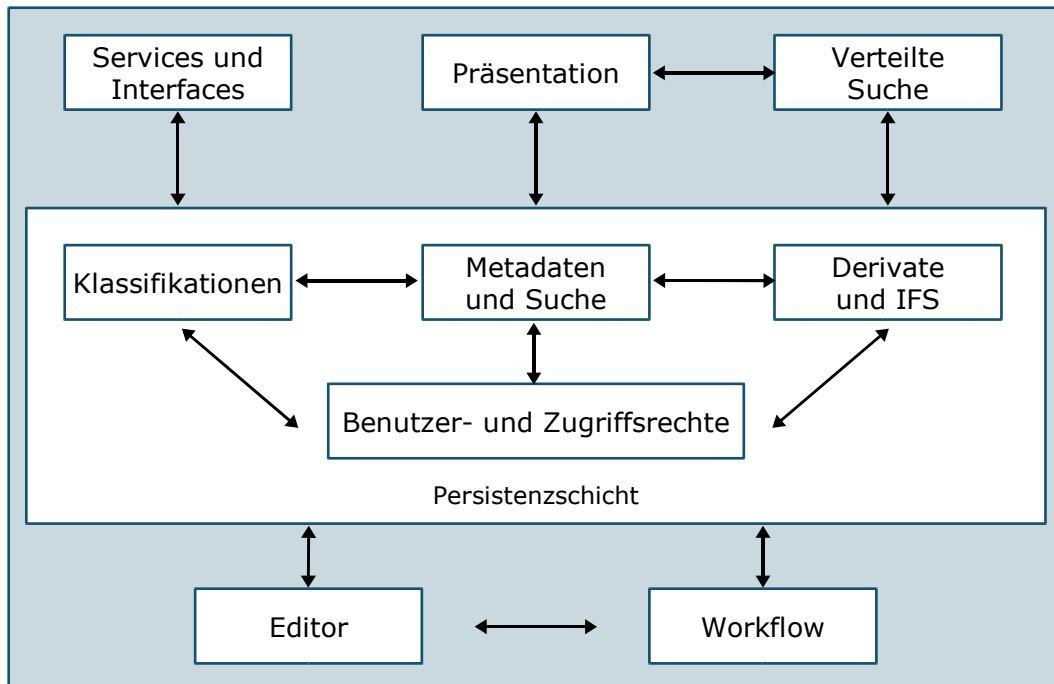


Abbildung 1.2: Komponentenmodell von MyCoRe

1.2 Hierarchisches Klassifikationssystem

Um eine bessere Wiederauffindbarkeit von Informationen, besonders in großen Datenbeständen, zu erreichen, wurden in allen Bereichen der Informationsverarbeitung Klassifikationen eingeführt. Dabei kann es sich um sehr allgemeine, wie die einer Herkunft oder eines Dateiformat-Typs, oder aber fach- bzw. anwendungsspezifische, Klassifikationen z. B. eng begrenzte Wissenschafts- oder Sachgebiete, handeln. In beiden Fällen sollte es jedoch möglich sein, hier auch eine Strukturierung der einzelnen Begriffe abbilden zu können.

MyCoRe implementiert ein Datenmodell mit folgenden Eigenschaften:

- Abbildung von hierarchischen Strukturen einer Klassifikation.
- Eindeutige Identifizierung einer Kategorie über mehrstufige Identifier.
- Speicherung der Kategorie-Bezeichner in mehreren Sprachen.
- Speicherung von Text-Beschreibungen zu den einzelnen Kategorien.
- Speicherung einer URL zu jeder einzelnen Kategorie.
- Jede Klassifikation ist eine XML-Datei und kann mit den für XML vorhandenen Werkzeugen bearbeitet werden. Eine Überprüfung der Syntax erfolgt über XML-Schema.

Auf die im MyCoRe-Projekt gespeicherten Klassifikationen kann von den Metadaten aus via einer Referenz aus Klassifikations- und Kategorie-ID zugegriffen werden.

1.3 Metadaten und Suche

1.3.1 Das MyCoRe-Datenmodell

Das MyCoRe-Datenmodell soll eine flexible Modellierung von Metadaten bei möglichst geringem Arbeitsaufwand für die Einzelanwendung gestatten. Um dies zu realisieren wurden die Ansätze implementiert:

- Die Notation der Daten erfolgt in XML-Form.
- Die eigentlichen Metadaten werden in ein Framework zur Objektverwaltung integriert.
- Es erfolgt eine Gestaltung eines XML-Schemas von Typen für die allgemein gültigen und oft gebrauchten Metadaten.

Ein MyCoRe-Metadaten-Objekt besteht aus den drei Teilen

`structure` - hier werden Informationen zur Einordnung des Datensatzes in eine logische Struktur festgehalten. Dies umfasst Verweise zu Eltern- und Kind-Metadaten-Sätzen. Weiterhin werden in dem Abschnitt Referenzen zu den Metadaten der eigentlichen digitalen Objekte gespeichert. Diese Referenzen erlauben den Zugriff auf die eigentlichen multimedialen Objekte.

`metadata` - in diesem Bereich werden die eigentlichen Metadaten gemäß des Datenmodells abgelegt. Jeder MyCoRe-Metadaten-Typ kann hier als eine Liste von Elementen angegeben werden. Jeder Metadaten-Typ hat mindestens folgende Eigenschaften:

- Er hat einen Schalter, ob er parametrisch, d. h. als Feld, suchbar sein soll.
- Er hat einen Schalter, ob er in einem Volltext suchbar sein soll.
- Er hat einen Schalter, ob er vererbt werden darf.
- Er hat einen Schalter, ob er selbst erben darf.
- Bei jedem Element kann die zugehörige Sprache spezifiziert werden.
- Jedes Element kann einen Attribut 'type' haben.

`service` - der letzte Abschnitt enthält Informationen zur Verwaltung des Metadaten-Satzes wie Erstellungsdatum, Zugriffsrechte (ACL's) oder Verwaltungs-Flags.

Sollten die im MyCoRe-Kern angebotenen Datentypen zur Gestaltung eines Projektes nicht ausreichen, so besteht die Möglichkeit, innerhalb der Anwendung eigene Typen zu kreieren und diese mit einzubinden. Somit ist MyCoRe auch für Nutzungsbereiche offen, von denen die Autoren bisher nicht ausgegangen sind. Der Datenmodellierung sind in MyCoRe kaum Grenzen gesetzt. Auch externe Stores für die Speicherung von Daten in anderen Suchmechanismen (z. B. Polygonsuche geographischer Daten) lassen sich problemlos integrieren.

1.3.2 Suche

Eine MyCoRe Anwendung sucht nicht direkt in den Metadaten von Objekten und Dateien, sondern in daraus abgeleiteten Suchfeldern. Die Abbildung von Metadaten auf Suchfelder erfolgt über eine Konfigurationsdatei. So wird z. B. das XML-Element `/metadata/titles/title` auf ein Suchfeld `title` abgebildet. Der Datentyp eines Suchfeldes (ID, Name, Text, Zahl, Datum etc.) bestimmt die bei der Suche einsetzbaren Operatoren (z.B. Phrasensuche, Trunkierung, `<`, `>`, ...). Es kann in den Metadaten von Objekten, in den Metadaten von Dateien, in gespeicherten XML-Dateien und im Volltext von Dateien gesucht werden.

Die Nutzung verschiedener Backend-Implementierungen für Indizierung und Suche ist möglich. Die Referenzimplementierung für die Suche in Metadaten und Volltexten verwendet die Apache Lucene Suchmaschine. Eine andere Implementierung verwendet ausschliesslich Strukturen im Hauptspeicher des Servers (JDOM Searcher) und ist zu Test- und Entwicklungszwecken gedacht. Alternative Implementierungen verwenden einen Suchindex basierend auf relationalen Datenbanktabellen (SQL mit/ohne Hibernate). Eine Implementierung der Suche basierend auf dem IBM Content Manager ist in Planung. Die verschiedenen Implementierungen unterstützen ggf. weitere, nicht-standard Suchoperatoren, beispielsweise ist eine unscharfe Suche mit dem „fuzzy“ Suchoperator der Apache Lucene Implementierung möglich.

Die Resultate einer Anfrage sind die IDs der gespeicherten XML-Metadaten-Objekte, welche in einem sie umgebenden XML-Container dem fragenden Prozess zurückgeliefert werden. Diese Resultatsliste kann nun zur Datenpräsentation mit XSLT aufbereitet werden.

1.3.3 Volltextindizierung und Suche

Neben der Suche in den Metadaten interessiert oft auch eine Volltextindizierung der digitalen Dokumente und die Suche nach den indizierten Begriffen. Die Volltextsuche ist inzwischen unabhängig vom Ort der Speicherung der Dateien. Sie können z.B. Ihre Dateien in einem herkömmlichen Dateisystemverzeichnis ablegen, oder auch im IBM Content Manager. Bei der Speicherung von Dateiinhalten werden die Volltexte unterstützter Datentypen über Textfilter-Plugins extrahiert und in einem Index abgelegt. Hierfür kann zur Zeit die Apache Lucene Implementierung verwendet werden, die sehr effektiv auch große Dateimengen indizieren kann und dabei eine Normalisierung und Stammwortreduktion (Stemming) von Wörtern durchführt. Eine zukünftige Implementierung der Volltextsuche wird die Stärken des IBM Content Manager nutzen, der durch die Verwendung der NSE eine hochwertige Volltextsuche anbietet. Ausserdem unterstützt der Content Manager alle gängigen Dateiformate. Derzeit ist eine Indizierung über die Apache Lucene Suchmaschine integriert. Die Erzeugung von Textextrakten für eine Durchsuchung hängt aber stark von den eingesetzten zusätzlichen Komponenten (installierten Textfilter-Plug-ins) ab. Möglich ist die Suche in PDF, OpenOffice swx, odt, Microsoft Word und PowerPoint, Text, XML, HTML und anderen Plain-Text-Formaten, sofern die ggf. erforderliche Zusatzsoftware (etwa OpenOffice) installiert ist.

1.3.4 Vererbung

In einigen Projekten ist es erforderlich, die Strukturierung der zu speichernden Objekte mit abzulegen. Eine hierarchische Anordnung der Teilobjekte, z. B. der Kapitel eines Buches, soll zu einem effizienteren Zugriff auf die gewünschten Objekte führen. Dabei sollen aber alle Metadaten-Informationen möglichst nicht redundant erfasst werden.

Um dies zu realisieren, wurde in das MyCoRe-Projekt ein Vererbungsmodell integriert, welches eine Verzahnung von Metadaten gleichen Datenmodelltyps zulässt. Dabei können jedem Metadaten-Element Informationen mitgegeben werden, ob es von Elementen eines Kind-Objektes geerbt werden kann oder diese von den Eltern erben soll. Auf diese Weise lassen sich auch komplexere Strukturen relativ leicht abbilden. Die Informationen über den Strukturzusammenhang einzelner Metadaten-Objekte werden im Frame-Bereich structure gespeichert und verwaltet. Abbildung 1.3 stellt das Vererbungsmodell nochmal grafisch dar.

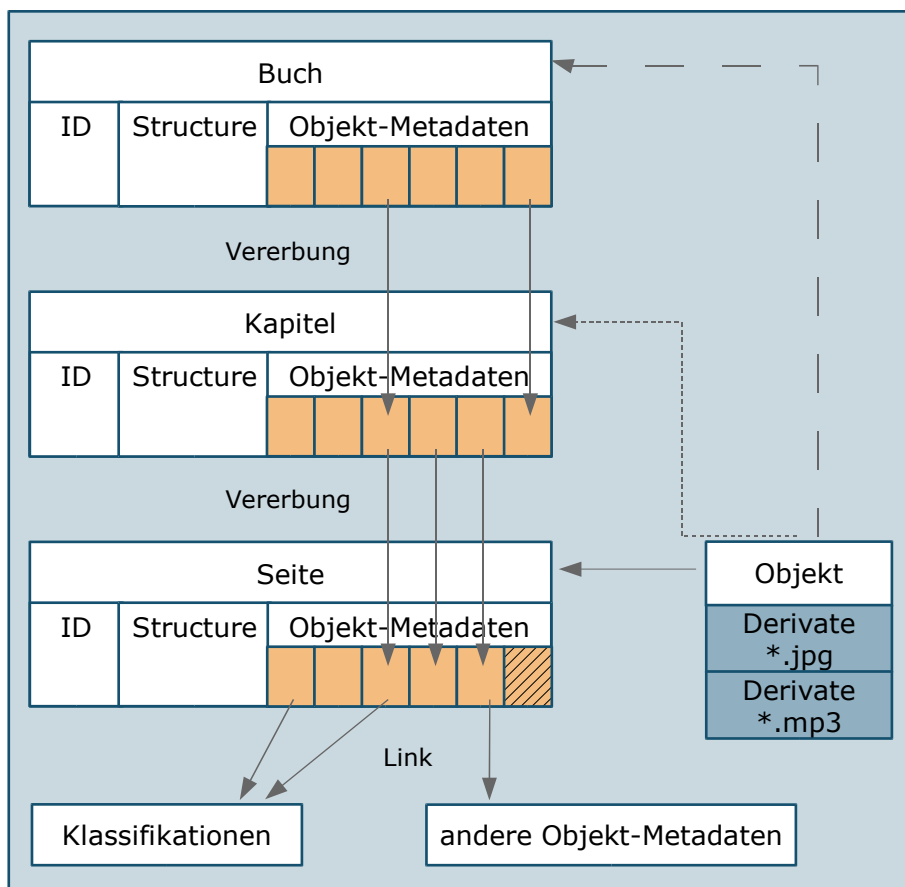


Abbildung 1.3: Vererbungsmodell von MyCoRe

1.4 Internes Dateispeicher-System

Viele Systeme ermöglichen die Pflege und Bearbeitung von beschreibenden Daten (Metadaten) zu Dokumenten und verwalten darüber hinaus einen Link auf die eigentlichen Dokumentvolltexte auf einem Webserver. MyCoRe hingegen verwaltet

nicht nur Metadaten, sondern auch die dazugehörigen Dateien selbst. Das bedeutet, dass die zu einem MyCoRe Objekt dazugehörigen Dateien (Derivate des Objektes), wie z. B. die PDF- oder HTML-Version eines Dokumentes oder das Foto einer Person bzw. Autors in das System importiert werden und dort verwaltet werden.

MyCoRe besitzt zu diesem Zweck ein Internes Dateispeicher-System (Internal Filesystem, IFS), das Dateien, Verzeichnisse und ihre Strukturen untereinander abbildet. Dieses Dateisystem verwaltet die Dateiinhalte (Content oder Derivate) und beschreibende Informationen wie Dateiname, MD5 Prüfsumme, Datum der letzten Änderung, Verzeichnisstrukturen usw. Schnittstellen (Command Line Interface, JavaServlets und API) erlauben es, Dateien zu importieren, zu verändern oder zu löschen, Dateien anhand von Pfadausdrücken in einer intern abgebildeten Verzeichnisstruktur aufzufinden oder Verzeichnisinhalte zu sortieren.

Über die MyCoRe-Kommandozeile (Command Line Interface) können Derivate, d. h. Einzeldateien oder Dateibündel, die zu einem Dokument gehören, importiert oder aktualisiert werden. Über ein Servlet werden Dateien und Verzeichnisse im Browser angezeigt und ausgeliefert. Es gibt auch die Möglichkeit, Dateien interaktiv über den Browser in das System hochzuladen.

Die beschreibenden und technischen Daten zu Dateien und Verzeichnissen werden in einem FileMetadataStore gespeichert. Prinzipiell kann es für diesen Store verschiedene Implementierungen geben, derzeit ist nur eine Implementierung für relationale Datenbanken realisiert, die sich z. B. mit MySQL oder DB2 nutzen lässt.

Die Dateiinhalte dagegen werden getrennt davon in einem FileContentStore gespeichert und verwaltet. Derzeit gibt es eine Reihe von Implementierungen eines FileContentStores, die je nach Konfiguration des Systems auch gleichzeitig eingesetzt werden können. Eine Liste finden Sie unter Punkt 4. dieser Dokumentation.

1.5 Verteilte Suche und Schnittstellen

1.5.1 Verteilte Suche

Wie bereits in Abschnitt 1.1 erwähnt, ist es möglich, mehrere gleichartige MyCoRe-Anwendungen, z.B. Dokumenten-Server, miteinander zu verknüpfen, so dass sie virtuell als Ganzes für den Benutzer erscheinen. Dies wird u.a. beim DocPortal demonstriert. Hier erfolgt eine Verbindung der einzelnen Installationen über eine Webservice-Schnittstelle. Der Ausbau der Webservice Funktionalitäten ist geplant.

1.5.2 Schnittstellen zu OAI, Z39.50, Webservices

Eine Schnittstelle für den Zugang via OAI (Open Archive Initiative) und eine erste Z39.50-Umsetzung sind bereits in MyCoRe integriert. Der Ausbau der OAI und Z39.50-Schnittstelle ist geplant bzw. in Entwicklung. Über die DocumentServlet-Schnittstelle lassen sich darüber hinaus die gespeicherten Daten abfragen. Diese können dann beliebig, z. B. mit XSLT, verarbeitet werden.

1.6 Benutzer- und Zugriffsrechteverwaltung

Im Subsystem Benutzermanagement wird die Verwaltung derjenigen Personen geregelt, die mit dem System umgehen (zum Beispiel als Autoren Dokumente einstellen). Dazu bietet die Benutzerverwaltung insbesondere die Möglichkeit, dass sich Benutzer am System authentifizieren können. Die Geschäftsprozesse einer Benutzerverwaltung wie zum Beispiel das Anlegen neuer Benutzer, das Setzen von Passwörtern, die Aktivierung/Deaktivierung von Benutzern usw. erfordern unterschiedliche Permissions. Diese können je nach Aufgaben festgelegt werden.

Die Benutzerverwaltungskomponente des MyCoRe-Projekts ermöglicht eine detaillierte Rollenübernahme einzelner Personen auf Basis von Benutzergruppen, denen bestimmte Rechte zugewiesen werden können. Die gesamte Rechteverwaltung funktioniert ab MyCoRe-Version 1.3 über eine integrierte Access-Controll-List (ACL). Diese ist eine eigenständige Komponente, welche Informationen über die vergebenen Zugriffs- und Änderungsrechte jedes einzelnen Datenobjektes enthält. Zu Import/Export-Zwecken können diese Rechte in XML-Form transformiert und an die Metadaten angehängt werden. (Detaillierte Informationen hierzu finden Sie im *MyCoRe Programmer Guide*.)

Weiterhin bietet MyCoRe den Zugang zum ACL- und User-System für Web-Applikationen via Servlet-Schnittstellen.

1.7 Datenpräsentation

Ziel einer MyCoRe-Anwendung ist es, die Daten des Systems dem Anwender in verschiedenster Art zu präsentieren. Da MyCoRe konsequent XML-basiert ist, bietet sich hier vor allem die Möglichkeit mittels XSLT eine Datenaufbereitung und -darstellung zu betreiben. Eine sehr zentrale Komponente ist hierbei das MCRLayoutServlet. Es bietet in eleganter Art und Weise die Möglichkeit einer Transformation der XML-Daten via Anwendungs-Stylesheets. So kann beispielsweise bei einem Verbund mehrerer Partner-Applikationen jede ihr eignes Layout zeigen. Der Funktionsbereich der Datenpräsentation erstreckt sich in MyCoRe von einfachen Web-Seiten über Trefferlisten und Resultats- bzw. Dokumentanzeigen bis hin zu Komponenten der Nutzerverwaltung.

1.8 Autoreninterface

Das MyCoRe-System bietet eine Schnittstelle für die Dateneingabe durch berechnigte Anwender. Hier ist das zentrale Glied das MCEditorServlet und die zugehörigen XSL-Stylesheets. Das System ist eine Mischung aus Konfigurationsdateien und anwendungsbezogenen Servlets. Dem Ideenreichtum bei der Gestaltung des Editors und der Prüfung der Daten entsprechend des Datenmodells sind kaum Grenzen gesetzt. So lassen sich für die MyCoRe Datenmodelle ansprechende Eingabewerkzeuge gestalten, welche harmonisch in die Anwendung integriert werden können. Auch das Einspielen von Dokumenten und multimedialen Objekten ist vorgesehen. Derzeit wird eine einfache Workflow-Implementation als Modul angeboten, welche die zu bearbeitenden Metadaten und Objekte auf einem Plattenbereich zwischenspeichert. Im Test hat sich diese Lösung

für einfache Aufgaben gut bewährt. Eine Anbindung an bestehende Workflow-Engines wie JBPM ist in Arbeit.

1.9 Zusätzliche Komponenten

MyCoRe ist kein statisches Produkt. Mit dem Entstehen einer jeden Anwendung besteht die Möglichkeit, allgemein interessierende neue Komponenten in den Kern mit aufzunehmen. Darüber hinaus werden eine Reihe von Applikationen zur Nachnutzung und Modifikation angeboten, so dass hier oft wenig Eigenentwicklung betrieben werden muss.

2 Hard- und Software-Voraussetzungen

Das MyCoRe-Projekt verwendet für die Speicherung der Objekte (multimediale Daten, Metadaten, Benutzerinformationen usw.) sowie für die Suche in Texten Softwarekomponenten von Drittanbietern (Third Party). Dabei werden zwei grundlegend verschiedene Ansätze unterstützt, und zwar die Verwendung der kommerziellen Software wie des IBM Content Manager⁴ der Firma IBM sowie die Verwendung frei verfügbarer Software-Komponenten. Jeder Ansatz hat seine eigenen Vor- und Nachteile, die an dieser Stelle nur kurz angerissen werden sollen. Bei der Entscheidung wird sicherlich die Berücksichtigung des bereits bestehenden IT-Umfeldes eine große Rolle spielen. In den folgenden Tabellen finden Sie die wesentlich eingesetzten Softwarekomponenten entsprechend des gewählten Ansatzes.

Allg. Basis	Mögliche Komponenten
Metadaten Store	Feldsuche mittels des Apache-Lucene-Projektes oder Nutzung des IBM Content Manager 8.3 mit integrierter Text-Suchmaschine IBM NSE.
Volltextsuche	Auch die Volltextsuche geht über das Apache-Lucene-Projekt oder die Nutzung des IBM CM Ressource Managers ist voll integriert.
Datenbank	Durch die Nutzung der Hibernate-API können alle darüber unterstützten Datenbanken genutzt werden. Getestet sind: DB2, MYSQL, HSQLDB
Dokument Stores	Filesystem mit VFS-Funktionen (SFTP usw.), IBM CM Ressource Manager, IBM Video Charger, Helix Server
Servlet-Engine	Jetty, Apache Tomcat, IBM WebSphere
OS	IBM AIX, Sun Solaris, Linux, Microsoft Windows

Tabelle 2.1: MyCoRe Komponentenübersicht

Die Hardware-Anforderungen hängen natürlich auch von der erforderlichen Performance, der Ausfallsicherheit usw. ab. Ein aktuell (Juni 2006) verfügbarer PC mit 2 GB Hauptspeicher oder mehr ist aber in der Lage, ein performantes System auf Basis der frei verfügbaren Software zu realisieren. Soll der IBM Content Manager zum Einsatz kommen, so können die minimalen Hardwareanforderungen für die einzelnen Komponenten (Video Charger, ContentManager, DB2) auf den Webseiten der IBM eingesehen werden. Auch sollten Sie Kontakt mit dem in Kapitel 5.3 genannten IBM-Ansprechpartner aufnehmen. Prinzipiell kann man aber sagen, dass durch den Einsatz des IBM WebSphere Application Server höhere Anforderungen an den Hauptspeicher gestellt werden. Der für die Content Manager Prozesse verfügbare Hauptspeicher sollte nicht unter 2 GB liegen.

⁴www.ibm.com/data/software/cm

3 Nutzungs- und Lizenzbedingungen

Das MyCoRe-Softwaresystem ist als Open Source unter der GNU General Public License (GPL) verfügbar. Damit darf MyCoRe von jedem verändert und weitergegeben werden, vorausgesetzt die Bedingungen der GPL werden befolgt.

Die GNU General Public License ist im Original unter <http://www.gnu.org/copyleft/gpl.html> nachzulesen, in vollständiger deutscher Übersetzung unter <http://www.gnu.de/gpl-ger.html>.

Die wichtigsten Bedingungen der GPL werden im Folgenden kurz zusammen gefasst. Diese Darstellung der GPL dient nur der groben Orientierung und ist kein Ersatz für die oben angegebene vollständige GPL.

§ 1. Die unveränderte GPL-Software darf vervielfältigt und weitergegeben werden, wenn jeder Kopie ein entsprechender Haftungsausschluss- und Copyright-Vermerk beigefügt wird.

§ 2. Die Verbreitung veränderter GPL-Software bedarf der zusätzlichen Vermerke, die darüber Auskunft geben, wer wann was geändert hat. Außerdem muss jede neue Software, die ganz oder zum Teil auf GPL-Software Dritter basiert, ebenfalls unter der GPL stehen.

§ 3. Sie dürfen Software unter den Bedingungen von § 1 und § 2 nur dann weitergeben, wenn Sie den zugehörigen Quellcode mitausliefern.

...

§ 6. Jedem Empfänger der Software müssen die gleichen Rechte und Pflichten zugestanden werden, wie Ihnen selbst. Einschränkungen sind nicht erlaubt.

...

§ 11. Gegenüber GPL-Software bestehen keine Gewährleistungsansprüche bezüglich Funktion, Qualität oder Verwendbarkeit.

§ 12. Kein Copyright-Inhaber oder Co-Autor kann für Schäden die aus der Benutzung oder Nichtbenutzbarkeit der Software entstehen, haftbar gemacht werden.

Entsprechend der Empfehlung in der GNU General Public License wird der Kopf jeder Javaklasse in MyCoRe um einen Hinweis zur GPL ergänzt. Darüber hinaus wird der Text der GPL als Textdatei `license.txt` mitgeliefert.

4 Roadmap

Die folgende Aufstellung soll interessierte Nachnutzer des MyCoRe-Projektes darüber informieren, welcher Funktionsumfang für die einzelnen Releases von den Entwicklern vorgesehen ist. Da die Entwicklung des Projektes durch das Developer-Team bedingt durch anderweitige Aufgaben nicht immer kontinuierlich betrieben werden kann, ist eine Mitwirkung neuer Anwender in höchstem Maße erwünscht.

4.1 Funktionsumfang des Releases 1.3 – 8. Juni 2006

4.1.1 Klassifikationen

- Das Klassifikationsdatenmodell ist implementiert und funktioniert.
- Klassifikationen werden in XML-Form gespeichert.
- Kategorien können hierarchisch angeordnet werden, die Kategorien bekommen dann mehrstufige ID's.
- Die Label und Descriptions der Kategorien können mehrsprachig sein.
- Die für das MyCoRe-Beispiel benötigten Klassifikationen sind vorhanden.
- Für die Web-Anwendung steht ein Servlet zur Verfügung, welches das Browsen in Klassifikationen ermöglicht.
- Kategorien von Klassifikationen können einen Verweis auf eine URL enthalten.
- Im MyCoRe-Beispiel sind weitere allgemeingültige Klassifikationen aus dem Bibliotheksumfeld abgelegt (z.B. PACS). Diese können für eigene Entwicklungen genutzt werden.
- Klassifikationen können per Commandline-Tools oder mittels eines Web-Editors erstellt und bearbeitet werden.

4.1.2 Metadaten-Modell

- Das allgemeine Datenmodell für die Metadaten ist implementiert.
- Es sind 16 komplexe Grunddatentypen zur Gestaltung von Anwendungen implementiert (siehe User Guide Kapitel 5.1.2.4). Eine Erweiterung seitens der Anwendungsentwickler ist möglich.
- Das Modell für die Datentypen gestattet eine mehrsprachige Speicherung.
- Die Metadaten können per Commandline-Tools und über ein Web-Interface verwaltet werden.
- Es gibt einen Migrations-Pfad von MILESS/MyCoRe 1.2 zu MyCoRe 1.3.
- Für einige Backends sind je nach Leistungsfähigkeit Textsuchfunktionen möglich.

- Es wurde ein Suchkonzept auf Basis einer effizienten, XML-basierten Syntax implementiert.
- Die Suche in den Daten kombiniert die Suche in den Metadaten mit einer Volltextsuche in den Dokumenten.
- Ein Event-Handler zur Speicherung und Indizierung der Daten wurde implementiert. Dadurch ist es möglich beliebige Abläufe beispielsweise beim Speichern von Daten zu definieren.
- Metadaten können über konfigurierbare Formulare per Web eingegeben werden. Eine Fehlerausgabe bei den Web-Formularen zur Pflege der Inhalte durch Eingabevalidierung im Editor-Framework erleichtert die Eingabe der Daten.

4.1.3 Internes File System

- Es sind Funktionalitäten zur Anbindung der eigentlichen Dokumente und multimedialen Objekte an die Metadaten implementiert.
- Es ist eine zentrale Verwaltungseinheit zur Klassifizierung und Verteilung der einzelnen Objekte auf die Stores vorhanden.
- Es sind folgende Stores implementiert:
 - FileSystem Store als Standardspeicher. Dieser integriert einen Lucene Store zur Speicherung der Textindizierung von *.txt, *.pdf, *.doc, *.swx usw. Dokumenten.
 - IBM Content Manager 8.3 Store inklusive Volltextindizierung der Dokumente
 - Helix Server Store für multimediale Objekte
 - IBM VideoCharger Store für multimediale Objekte

4.1.4 Benutzer- und Rechtesystem

- Die Benutzerdaten bestehen aus Nutzern welche in Gruppen angeordnet sind.
- Die Benutzerdaten können per Commandline-Tools und über ein Web-Interface verwaltet werden.
- Das Benutzersystem hat eine API Schnittstelle (MCRUserManager) zu den anderen Komponenten von MyCoRe.
- Das System bietet eine einheitliche Login-Komponente. Für die Integration eines Login-Dialogs in die Datenpräsentation stehen entsprechende Servlets und XSLT Stylesheets bereit.
- MyCoRe verwaltet Access-Controll-Lists (ACL's) für die Datenobjekte.
- Via Servlet können die ACL-Permissions einzelner Datenobjekte für die jeweiligen Nutzer abgefragt werden.
- Die ACL's regeln die Zugriffsrechte für Nutzer bzw. Nutzergruppen, Einstelldaten und IP's/Netzwerke.

4.1.5 Datenpräsentation

- Die Datenpräsentation basiert auf XSLT 1.0.
- In der Beispielanwendung werden folgende Möglichkeiten der Darstellung gezeigt:
 - einfache Web-Dokumente und Seiten
 - Suche in der Datenbasis
 - Anzeige der Resultatsliste
 - Anzeige der Objekt-Metadaten
 - Classifikation browsing
 - Index browsing
 - Anzeige der Attribute des Objektes und des Objektes selbst
 - Integration eines Image-Viewers
 - Umschalten der Präsentation auf eine andere Sprache
 - spezielle Seiten für den Zugang von Robots
- Eine Schnellsuche wird in die Startseite integriert.
- Es gibt eine Expertensuche zur Demonstration der Suchmöglichkeiten.
- Die Arbeit der implementierten Servlets läuft sitzungsorientiert (Sessions), d.h. alle Servlets kennen den konkreten Stand der einzelnen Sitzungen.
- Die Daten können wahlweise auch gepackt als zip-Datei heruntergeladen werden.

4.1.6 Backend

- Neben Klassen für den Zugriff auf nativ SQL-Backends wie MySQL und DB2 steht eine Hibernate-Implementierung für einen transparenten Zugriff zur Verfügung.
- HSQLDB wird für die Beispielanwendung mitgeliefert. Eine separate Installation eines RDBMS ist somit für kleine Anwendungen nicht mehr zwingend notwendig.

4.1.7 Autorenwerkzeug

- Zum Erstellen neuer Datensätze steht ein Autorenwerkzeug zur Verfügung.
- Für das Einstellen neuer Objekte und Ändern des Bestandes kann mit Hilfe des SimpleWorkflow-Modules ein einfacher Arbeitsablauf realisiert werden.
- Objektdateien und -verzeichnisse können wahlweise mit einem Java-Applet oder in eingeschränkter Form mittels Servlet-Anwendung in das System eingebracht werden.
- Es steht ein direkter Zugang zu Bearbeitungsfunktionen aus der Detailansicht zur Verfügung.

- Das Editor-System bietet eine einfache Form der direkten Datenvalidierung.

4.1.8 Remotezugriff und Services

- Es wurde via WebServices eine Möglichkeit der Remote-Abfrage von Systemen geschaffen.
- Es wurde ein OAI Client gemäß Standard 2.0 implementiert.
- Es existiert ein Z3950 Client.

4.1.9 zusätzliche Funktionalitäten

- Es wurde ein Tool zur Generierung von NBN's implementiert.
- Über ein Index-Servlet werden die Daten für Search Robots aufbereitet angeboten.
- Das WCMS-Modul ermöglicht eine Web-basierte Administration der statischen Web-Inhalte.

4.1.10 Beispielanwendung

- Es existiert eine Beispielanwendung, welche auf einfache Weise installiert werden kann.
- Die Beispielanwendung dokumentiert den derzeitigen Stand eines Dokumenten-Servers.
- Es sind alle Funktionalitäten der Datenpräsentation vorhanden.
- Für die Beispielanwendung wird separat eine große Menge von Datensätzen zum Test angeboten.
- Es gibt eine Binärdistribution, die eine schnelle und einfache Installation der Beispielanwendung ermöglicht.

4.1.11 Dokumentation

- Die Dokumentation gestattet die Installation des MyCoRe-Beispiels.
- Mit ihrer Hilfe kann ein eigener Dokumenten-Server aufgebaut werden.
- Sie vermittelt einen Einblick in das Design des Projektes und gestattet dem Anwender die Implementation eigener Komponenten und Erweiterungen.

4.2 Funktionsumfang des Releases 1.4 (Ende 2006)

4.2.1 Allgemeines

Mit Release 1.4 werden eine Reihe alter Klassen des Release 1.2 aus dem MyCoRe-Kern verschwinden. Parallel dazu werden die Strukturen der Beispielanwendung DocPortal entsprechend gestrafft.

4.2.2 Metadaten-Modell

- Die Metadaten-Typen für die ISBN und die NBN werden überarbeitet und ergänzt.
- Alle MyCoRe-Tags und Attribute erhalten den Namespace mcr.

4.2.3 Internes File System

- Weitere Dateitypen können textindiziert werden.
- Eine Suche in Scorm-Daten ist möglich.

4.2.4 Benutzer- und Rechtesystem

- Nutzung bzw. Anbindung an externe Indentifikationssysteme (Shiboleth, LDAP)

4.2.5 Datenpräsentation

- ACL's für Klassifikationen
- Implementation von Warenkorbfunktionen in das Beispiel.
- Weitere WebServices sind in Produktion nutzbar.
- Es soll eine Inventar-Funktion möglich sein.

4.2.6 Backend

- Verbesserte Unterstützung des IBM Content Managers 8.3.

4.2.7 Beispielanwendung

- Die Beispielanwendung repräsentiert einen vollständigen Dokumenten-Server und ist in einem Anwendungspaket untergebracht.

4.2.8 Dokumentation

- Es wird ein *MyCoRe Contributors Guide* angelegt.
- Das *MyCoRe Programmer Guide* wird überarbeitet und ergänzt.
- Das *MyCoRe User Guide* wird überarbeitet und ergänzt.

4.2.9 Sonstiges

- Verringern der Komplexität bei Installation und Konfiguration
- Weitere Beispieldaten werden hinzukommen.

5 Mitwirkende am Projekt und Ansprechpartner

5.1 Architekturboard

Das Architekturboard ist während einer der ersten gemeinsamen Sitzungen der Community entstanden. Es umfasst eine kleine Gruppe von wenigen Mitgliedern, die auf lange Sicht am MyCoRe-Projekt beteiligt sind und Entscheidungen treffen können. Das Board soll die Entwicklung koordinieren, einzuhaltende Standards sowie Schnittstellen festlegen und das auch überwachen.

Derzeit zählen zu dieser Gruppe:

- Frank Lützenkirchen (luetzenkirchen@bibl.uni-essen.de)
- Jens Kupferschmidt (kupferschmidt@rz.uni-leipzig.de)
- Thomas Scheffler (thomas.scheffler@uni-jena.de)
- Ulrike Krönert (ulrike.kroenert@uni-jena.de)
- Anja Schaar (anja.schaar@uni-rostock.de)

5.2 Projektmitwirkende

An der Entwicklung des MyCoRe-Kerns sind mehrere deutsche Universitäten beteiligt. Daneben gibt es weitere Hochschulen, die MyCoRe-Anwendungen entwickeln und einsetzen. Nachfolgend sind die aktuell in die Entwicklung involvierten Universitäten aufgeführt (in alphabetischer Reihenfolge der Standorte). Für jeden Standort ist mindestens ein Ansprechpartner genannt, an den Sie sich bei Bedarf wenden können.

- **Universität Duisburg-Essen (Rechenzentrum, Bibliothek)**

Frank Lützenkirchen
Universität Duisburg-Essen, Universitätsbibliothek Campus Essen
Universitätsstrasse 9-11
D-45141 Essen
email: luetzenkirchen@bibl.uni-essen.de
Tel: +49-(0)201-183-2124

- **Albert-Ludwigs-Universität Freiburg (Rechenzentrum)**

Dr. Detlev Degenhardt (Entwicklung und Betrieb)
URZ Universität Freiburg
Hermann-Herder Str. 10
D-79104 Freiburg
email: Detlev.Degenhardt@rz.uni-freiburg.de
Tel: +49-(0)761-203-4655
Fax: +49-(0)761-203-4643

- **Universität Hamburg (Bibliothek, Rechenzentrum)**

Dr. Stefan Gradmann
Regionales Rechenzentrum, Leiter Arbeitsgruppe Virtuelle Campusbibliothek
Schlüterstr. 70
D-20146 Hamburg
email: stefan.gradmann@rrz.uni-hamburg.de
Tel: +49 (0)40-42838-3093
Fax: +49 (0)40-42838-3284

- **Friedrich-Schiller-Universität Jena (Rechenzentrum, Bibliothek)**

Ulrike Krönert
Universitätsrechenzentrum
Am Johannisfriedhof 2
D-07743 Jena
email: zuk@rz.uni-jena.de
Tel: +49-(0)3641-940-534
Fax: +49-(0)3641-940-502

- **Universität Leipzig (Rechenzentrum)**

Jens Kupferschmidt (Entwicklung und Betrieb)
Universitätsrechenzentrum
Augustusplatz 10-11
D-04109 Leipzig
email: kupferschmidt@rz.uni-leipzig.de
Tel: +49-(0)341-97-33348
Fax: +49-(0)341-97-33399

- **Universität Rostock (Informatik, Rechenzentrum, Bibliothek)**

Prof. Dr. Andreas Heuer
Fachbereich Informatik - Lehrstuhl Datenbank- und Informationssysteme
Albert-Einstein-Straße 21
D-18059 Rostock
email: heuer@informatik.uni-rostock.de
Tel: +49-(0)381-498 3404
Fax: +49-(0)381-498 3443

5.3 Ansprechpartner

Hier finden Sie Kontaktinformationen, sofern Sie Fragen zu bestimmten Aspekten des MyCoRe-Projektes haben:

- **Allgemeine Fragen zu MyCoRe**

Kathleen Krebs
Projektkoordination MyCoRe
Regionales Rechenzentrum der Universität Hamburg
Schlüterstr. 70

D-20146 Hamburg
email: kathleen.krebs@uni-hamburg.de
Tel: +49 (0)40-42838-7212
Fax: +49 (0)40-42838-3284

Frank Lützenkirchen
luetzenkirchen@bibl.uni-essen.de
Universität Duisburg-Essen, Universitätsbibliothek Essen
Universitätsstrasse 9-11
D-45141 Essen
Tel: +49-(0)201-183-2124
Fax: +49-(0)201-183-3231

- **Fragen zu DocPortal mit MyCoRe**

Jens Kupferschmidt
kupferschmidt@rz.uni-leipzig.de
Universitätsrechenzentrum Leipzig
Augustusplatz 10-11
D-04109 Leipzig
Tel: +49-(0)341-97-33348
Fax: +49-(0)341-97-33399

- **Fragen zu IBM Content Manager und zur Rolle der IBM in MyCoRe**

Dr. Stefan Gradmann (Koordination der MyCoRe/IBM Kooperation)
stefan.gradmann@rrz.uni-hamburg.de
Regionales Rechenzentrum
Schlüterstr. 70
D-20146 Hamburg
Tel: +49 (0)40-42838-3093
Fax: +49 (0)40-42838-3284

Dr. Ulrich Groh (IBM)
ugroh@de.ibm.com
IBM Deutschland, CUE Cluster Education & Research Germany
Tel: +49-(0)711-785-4853
Fax: +49-(0)711-785-4997

6 Anhang

6.1 Abbildungsverzeichnis

Abbildung 1.1: Grobes Schichtenmodell von MyCoRe.....	1
Abbildung 1.2: Komponentenmodell von MyCoRe.....	2
Abbildung 1.3: Vererbungsmodell von MyCoRe.....	5

6.2 Tabellenverzeichnis

Tabelle 2.1: MyCoRe Komponentenübersicht.....	9
---	---