PS Software Engineering

NC-vvv-yyyy 1st February 2001

ArchäoDB

Report

Christopher Rettenbacher
cretten@cosy.sbg.ac.at

Schmidbauer Josef
jschmid@cosy.sbg.ac.at

Markus Spitzlinger
mspitzl@cosy.sbg.ac.at

Gerald Stieglbauer
gstiegl@cosy.sbg.ac.at

Gerald Wiesbauer
wiesi@cosy.sbg.ac.at

Academic Supervisor Mag. Dipl.-Ing. Schwaiger Roland

Department of Computer Science
University of Salzburg

Correspondence to:
Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Title

Christopher Rettenbacher, Josef Schmidbauer, Markus Spitzlinger,
Gerald Stieglbauer, and Gerald Wiesbauer
Department of Computer Science
University of Salzburg
A–5020 Salzburg, Austria
cretten@cosy.sbg.ac.at, jschmid@cosy.sbg.ac.at, mspitzl@cosy.sbg.ac.at,
gstiegl@cosy.sbg.ac.at, gwies@cosy.sbg.ac.at

1st February 2001

Abstract

Ein allgemeines System zur Erfassung, Verwaltung, und Darstellung von archäologischen Daten über ein Web Frontend soll im folgenden entwickelt werden. Teile des Datenbestandes, welche unter einem solchen System verwaltet werden, sollen in einem öffentlich zugänglichen virtuellen Museum dargestellt werden können.

Contents

Chapter 1

Aufgabe 1

1.1 Problembeschreibung

1.1.1 Einleitung

Das System soll die Erfassung, Verwaltung, und Darstellung von archäologischen Daten über ein Web Frontend ermöglichen. Teile des Datenbestandes sollen in einem öffentlich zugänglichen virtuellen Museum dargestellt werden können.

1.1.2 Datenbestand

Die Daten beziehen sich auf archäologische Fundstücke, die dem Museum zur Verfügung stehen. Es werden Daten unmittelbar zu den Artefakten, aber auch zu Fundorten (Ausgrabungsstätten,Siedlungen,Gräber) oder Entdeckern erfaßt. Diese Objekte werden miteinander in Beziehung gesetzt, um eine annähernd hierarchische Strukturierung, in Hinblick auf historische bzw. archäologischen Zusammenhänge zu erreichen. Außerdem können Audio-,Video-, und Bilddateien gespeichert, und von in Zusammenhang stehenden Objekten referenziert werden.

Weiters müssen relevante Daten(Schautafeln zu einzelnen Räumen) für das Virtuelle Museum gespeichert werden, die mit den übrigen Daten nicht unmittelbar in Kontakt stehen.

Die Haltung der Daten erfolgt zentral.

Lese- und Schreibrechte sind von der Benutzerklasse abhängig.

1.1.3 Benutzergruppen

Prinzipiell wird zwischen Erfassern und Internetbesuchern unterschieden.

Wissenschaftler (wissenschaftliche Erfasser), sowie mit der Datenerfassung beauftragte Museumsmitarbeiter (reguläre Erfasser) können Datensätze erfassen, und bearbeiten, sowie entscheiden, welche der Informationen für Internetbesucher freigegeben werden. Weiters können sie Beziehungen zwischen den Daten definieren. Die regulären Erfasser beschränken sich auf die Erfassung der Grunddaten, während sich die wissenschaftlichen Erfasser für einen erweiterten Datenbestand (wissenschaftliche Abhandlungen, Interpretationen) verantwortlich zeigen.

Auch die Erstellung virtueller Museen ist den Erfassern möglich.

Internetbesucher haben lediglich eingeschränkten Lesezugriff auf die archäologischen Daten.

1.1.4 Benutzerschnittstelle

Über spezielle Sicherheitsanforderungen ist noch nichts bekannt.

Web Frontend

Der Zugriff auf den Datenbestand erfolgt für sämtliche Benutzer über ein zu entwickelndes Web Frontend. Je nach Benutzer oder Benutzergruppe kann das Aussehen des Web Frontends variieren (verschiedene Masken) um den unterschiedlichen Anforderungen Rechnung zu tragen. Die Funktionalität hängt also vom eingelogten Benutzer ab.

Durch die zwischen den Objekten definierten Beziehungen sollte es möglich sein, durch die Datenbank zu 'surfen'.

Es ist möglich, mittels geeigneter Kriterien nach bestimmten Daten zu suchen.

Virtuelles Museum

Zusätzlich soll es möglich sein, aus einer Auswahl der erfaßten Daten ein virtuelles Museum zu erstellen. Der Zugriff erfolgt ebenfalls über das Web Frontend.

Artefakte sollen gruppiert (virtuelle Vitrinen) und beliebig kombiniert werden können (virtuelle Räume). Es soll möglich sein, über den bisherigen archäologischen Datenbestand hinaus, Informationen zu den Schauräumen und ausgestellten Objekten anzugeben und darzustellen.

Das System soll dem Entwickler Hilfsmittel zur Verfügung stellen und Gestaltungsmöglichkeiten offen lassen, um ein augenfälliges Design zu ermöglichen.

Die virtuellen Museen sollen auf CD-ROM gespeichert und in Katalogform dargestellt werden können.

1.1.5 Anforderung an die Entwicklungsumgebung

- Rational Unified Prozess wird als Grundlage zur Systementwicklung verwendet.
- XML soll eingesetzt werden.
- Geeignete Hilfsmittel zur Wartung und Weiterentwicklung von Datenbanken und Programmen sollen entwickelt werden. Die SE AG soll nach Entwicklungabschluß als Application Service Provider auftreten.

Chapter 2

Aufgabe 2

2.1 Ausgangssituation und Zielsetzung

2.1.1 Einleitung

Das System soll die Erfassung, Verwaltung, und Darstellung von archäologischen Daten über ein Web Frontend ermöglichen. Teile des Datenbestandes sollen in einem oder mehreren öffentlich zugänglichen virtuellen Museen dargestellt werden können. Aus diesen virtuellen Museen sollen auch CD-ROMs und ein gedruckte Kataloge automatisch erstellt werden können. Das System soll weiters auch in real existierenden Museen als Internet-Terminal installiert werden.

2.1.2 Datenbestand

Beschreibung

Die Daten beziehen sich auf archäologische Fundstücke, die dem Museum zur Verfügung stehen. Es werden Daten unmittelbar zu den Artefakten, aber auch zu Fundorten (Ausgrabungsstätten,Siedlungen,Gräber) oder Entdeckern erfaßt. Diese Objekte werden miteinander in Beziehung gesetzt, um eine annähernd hierarchische Strukturierung, in Hinblick auf historische bzw. archäologischen Zusammenhänge zu erreichen. Außerdem können Audio-,Video-, und Bilddateien gespeichert, und von in Zusammenhang stehenden Objekten referenziert werden.

Logisch gesehen gehören diese Daten zu verschiedenen Gruppen und können getrennt verwaltet werden.

Der Datenbestand enthält auch Informationen darüber ob Objekte real vorhanden sind (Lager, Museum) und wenn dies der Fall ist Informationen über deren Lokalisation.

Anforderungen

Die Haltung der Daten erfolgt zentral.

Das System dient zur Erfassung folgender *archäologischer Objekte*:

- Artefakte

- Gräber
- Siedlungen

Diese genannten Objekte können mit Entdeckern, Literatur, anderen archäologischen Objekten,... in Beziehung gesetzt werden. Zu den jeweiligen Bezugsobjekten müssen zusätzliche Daten abgelegt werden können (z.B. Lebenslauf,... bei Entdeckern oder Erfassern, ISBN Nummern bei Literaturangaben, etc.). Dabei werden übliche Daten wie z.B. Name, Geb. Dat, Verheiratet, Wohnhaft, E-Mail etc. gespeichert. Der Kunde überlässt es der SE AG geeignete Attribute vorzuschlagen.

Folgende *Attribute* müssen zu jedem archäologischen Objekt gepeichert werden können:

- Inventarnummer
- Bezeichnung
- Dimension
- Versicherungswert
- Lagerinformation

Weitere Attribute der Objekte und Attribute von anderen Objekten wie z.B. Audio oder Videodaten sind derzeit noch nicht festgelegt.

Weiters müssen relevante Daten (Schautafeln zu einzelnen Räumen) für das Virtuelle Museum gespeichert werden, die mit den übrigen Daten nicht unmittelbar in Kontakt stehen.

Zusatzdaten wie Stichwörter zu den Videos sollen nicht abgespeichert werden. Allerdings wurde vom Kunden die Möglichkeit gewünscht z.B. Videodaten mit Personen oder Standortsangaben zu verknüpfen um nach ihnen suchen zu können.

Umfang

Der Anfangsbestand an Daten ist null.

Der Datenbestand wird sich jährlich wie folgt erweitert:

1. 400 Bilder (diese Zahl entspricht auch den neu hinzugefügten Artefakten)
2. 400 Handzeichnungen
3. 10 Filme a 10 sec und 10 Audiodaten
4. Texte (genaue Anzahl schwer zu erfassen)

Sicherheit

Lese- und Schreibrechte sind von der *Benutzergruppe* abhängig. Die Rechte sollen bis auf die Attributebene (d.h. für jedes einzelne Attribut) genau festgelegt werden. Beispielsweise soll, wenn ein besonderes Artefakt gefunden worden ist, für jedes Attribut die benötigten Rechte festgelegt werden können. können. Wenn keine Rechte explitzit vergeben werden so wird eine Default-Angabe angenommen. Diese kann vom Erfasser geändert werden.

Format

Graphiken, Photos sollen dynamisch bei der Verwendung in ein geeignetes Format umgerechnet werden. Dabei wird vorgeschlagen intern Bilder als JPEG oder GIF zu speichern.

Videos und Audiodaten werden in der höchst möglichen Qualität, sowie in zwei verschiedenen Qualitätsstufen (Hohe Qualität, Niedrige Qualität) abgespeichert. Die beiden Qualitätsstufen dienen vor allen für die Internetanwender und sollen in einem gängigen Format abgelegt werden. Der Kunde hat hierbei RealAudio und MP3 (falls ein Freeware-Encoder verfügbar ist) bzw. RealVideo als Format festgelegt.

Austausch

Der Kunde erwartet daß das System die Formate JPEG und GIF bei Bildern, sowie RealVideo und RealAudio (bzw. MP3, falls ein Freeware-Encoder verfügbar ist) bei den Multimedia Daten verarbeiten kann (Interoperabilität).

Zusätzlich muß der Datenaustausch mit Schwestersystemen möglich sein. Diese verwenden exakt die selbe Software und werden auch von der SE AG betreut - allerdings für einen anderen Kunden. Beide Systeme sind Soft und Hardwareseitig voneinander unabhängig. Es muß möglich sein das Objekte von einem System auf das andere System transferiert werden können (Ausleihe von Ausstellungsstücken). Diese haben aber nur eine bestimmte Gültigkeitsdauer. Diese muß vom Ausleiher definiert werden können. Dannach erfolgt auch die Abrechnung für ein Leihstück. Das vermietete Objekt wird als physikalische Kopie auf dem Schwestersystem gespeichert. D.h. insbesondere auch das sich das Leihstück auch nicht ändern darf, selbst wenn die ursprünglichen Daten geändert wurden.

Löschen

Von Seiten des Kunden wird ein “Multilevel-Undo” gewünscht. Alle Änderungen (Deltas) sollen gespeichert werden. Damit ist ein beliebiges zurücknehmen von Aktionen möglich.

2.1.3 Benutzer

Auf das Systemen greifen *Benutzer* zu. Ein Anwender kann sich beim System *anmelden* und bekommt dann die, ihm zugeordneten Benutzergruppen zugewiesen. Solange dieser Benutzer das System verwendet verfügt er über die Rechte die in den entsprechenden Benutzergruppen zugeordnet sind. Wenn sich ein Anwender nicht anmeldet (Internetbenutzer) bekommt er automatisch alle eingerichteten Benutzergruppen zugeordnet für die keine Anmeldung erforderlich ist. Diese Gruppen werden zweckmässigerweise auf Lesezugriff auf das System beschränkt sein.

Jeder Benutzer ist für die von ihm eingegebenen Daten bzw. die erstellten Objekte, Museem, Räume, Vitrinen vorantwortlich. Dies gilt sowohl für die Rechtevergabe (Änderung der vorgegebenen Defaultrechte im ihm erlaubten Rahmen), als auch dafür das er rechtlich für den Inhalt verantwortlich ist. Dies gibt insbesondere für die öffentliche Darstellung des Museums im Web oder auf CD-ROM.

Der Kunde erwartet das maximal 100 Internetbenutzer und 5 Bearbeiter das System gleichzeitig verwenden. Dabei beschränkt das System die Anzahl der Benutzer, falls die Antwortzeit zu hoch wird.

Festzulegen sind eine genaue Unterscheidung zwischen Benutzern und Benutzergruppen bzw. deren Rechten und Attributen.

2.1.4 Benutzergruppen

Es können verschiedene Benutzergruppen erstellt werden welchen individuell Rechte zugeordnet werden können. Die vorhergehend vorgesehene Unterteilung in Erfasser, wissenschaftliche Erfasser und Internetbesucher wird aufgelassen. Diese Benutzer können aber im neuen Schema als Benutzergruppen angelegt werden (z.B. Internetbenutzer mit sehr beschränkten Rechten und Notwendigkeit einer Anmeldung).

Durch die Benutzergruppen erreicht man eine sehr hohe Flexibilität und man kann auch nach der endgültigen Realisierung des Projektes diese Gruppen problemlos und einfach verwalten.

Folgende Aktionen können in diesem Zusammenhang durchgeführt werden:

- Benutzer und Benutzergruppen anlegen
- Benutzer und Benutzergruppen löschen
- Benutzer und Benutzergruppen ändern
- Benutzern Benutzergruppen zuordnen

Einzelne Benuztergruppen können verschiedene Rechte haben. Diese Rechte werden als Attribut eines Benutzers gespeichert.

Diese Rechte können beispielsweise sein:

- Eintragen von neuen archäologischen Objekten
- Ändern dieser Daten (bei für die jeweilige Benutzergruppe freigegebenen Attributen)
- Löschen/Entfernen von Objekten aus der Datenbank
- Daten zu suchen
- Rechtevergabe für andere Benutzergruppen
- Anmeldung notwenig

Rechte können bis Attributebene genau vergeben werden (z.B. Durchsuchen). Ebenfalls kann festgelegt werden wer das Recht hat Zugriffsrechte für eine Attribut zu ändern.

Sinnvoll wird es sein eine Benutzergruppe "Administratoren" einzuführen welche über die Rechte verfügt alle anderen Rechte zu ändern (auch die Default Rechte). Der Kunde soll dabei selber Administrator sein können.

Weiters wird es notwendig sein eine Benutzergruppe "Internetanwender" einzuführen für welche keine Anmeldung erforderlich ist. Es wird sinnvoll sein die Rechte dieser Benutzergruppe stark einzuschränken.

Benachrichtungungen

Es wird vorgeschlagen das Benutzer aus einzelnen Benutzergruppen benachrichtigt werden wenn ein Benutzer der selben oder einer anderen Benutzergruppe Daten welche vom erstgenannten Benutzer eingegeben wurden ändert oder löscht. Diese Benachrichtigung könnte über E-Mail erfolgen. Die E-Mail Adresse ist hierbei ein *Attribut* eines Benutzers. Eine solche Meldung könnten z.B. den Namen der Person welche die Daten geändert hat und die Daten selber (in geeigneter Form, z.B. durch die Bezeichnung, der Inventarnummer,...) enthalten.

Wenn ein Benutzer beim Einloggen ein falsches Passwort zu seiner Benutzerkennung eingibt soll sowohl an den Administrator als auch den Benutzer selber eine Mail gesendet werden. Der Benutzer soll jedoch nicht gesperrt werden.

2.1.5 Benutzerschnittstelle

Web Frontend

Der Zugriff auf den Datenbestand erfolgt für sämtliche Benutzer über ein zu entwickelndes Web Frontend. Je nach Benutzer oder Benutzergruppe kann das Aussehen des Web Frontends variieren (verschiedene Masken) um den unterschiedlichen Anforderungen Rechnung zu tragen. Die Funktionalität hängt also vom eingelogten Benutzer ab. Dies ist auch deshalb sinnvoll weil verschiedene Benutzer im Allgemeinen verschiedenen Benutzergruppen angehören und damit bestimmten Einschränkungen hinsichtlich der erlaubten Operationen unterliegen. Die Masken können so variert werden, daß Operationen welche für eine Benutzergruppe nicht erlaubt sind gar nicht mehr angezeigt werden. Damit wird auch das Risiko der Fehlbedienung reduziert und der Benutzer unterstützt.

Ein Internetbenutzer kann sich registrieren und erhält dadurch einen neuen Status (mit Username und Passwort). Es könnten z.B. mehr Informationen (oder in besserer Qualität) für die registrierten Internetbenutzer angezeigt werden. Welche Informationen oder Rechte der Internetbenutzter durch die Registrierung zusätzlich hat kann dabei von Kunden durch die individuelle Rechtevergabe der Benutztergruppen regeln. Dies ermöglicht dem Kunden Werbemails bzw. Newsletters an die registrierten Benutzer zu verschicken. Ausserdem denkt der Kunde an die Einführung eines Online Fanklubs.

Das Web Frontend verfügt über Funktionalität welche es erlaubt über die, zwischen den Objekten definierten Beziehungen durch die Datenbank zu 'surfen'. Dabei werden zu jedem Objekt die zugehörigen Beziehung zu anderen Objekten als auswählbarer Link dargestellt.

Es werden zwei verschiedene Suchmodi (vom Kunden gefordert) unterstützt:

- EINFACHE SUCHE

  Der Benutzer gibt einen Suchbegriff vor und das System durchsucht alle Attribute auf welche der Benutzer zugreifen kann nach diesem Begriff. Alle gefundenen Objekte werden in einer Liste angezeigt. Der Benutzer kann aus dieser Liste Objekte auswählen und sich nähere Informationen (d.h. die Attribute des Objektes) anzeigen lassen.

- ERWEITERTE SUCHE

> Der Benutzer kann mehrere Suchbegriffe eingeben. Jeder dieser Suchbegriffe kann auf ein oder mehrere Attribute von wählbaren Objekten eingeschränkt werden. Die Suchbegriffe können untereinander mit Boolschen Operatoren verknüpft werden. Alle Attribute von Objekten für die der Benutzer Rechte hat werden dabei durchsucht. Die Ergebnisse werden in einer Liste angezeigt. Der Benutzer kann aus dieser Liste Objekte auswählen und sich nähere Informationen (d.h. die Attribute des Objektes) anzeigen lassen. Zu jedem Ergebniss wird der Teil der Erweiterten Suche angezeigt der zu duesem Ergebniss führte (z.B. bei Suche nach "ROM oder ITALIEN" Anzeige das das Ergebniss gefunden wurde weil in der Beschreibung "ROM" vorkommt). Dies führt bei komplexeren Suchen zu einer verbesserten Orientierung für den Anwender.

Bei der Suche soll es auch möglich sein nach abstrakten Daten (Meta Begriffen, Schlagwörtern) wie z.B. Epochen, Stilrichtungen, etc. zu suchen. Jedem Objekt können mehrere Schlagwörter zugeordnet werden (z.B. "Mittelalter"). Eine Meta Suche soll sich darauf beschränken diese Schlagwörter zu durchsuchen.

Es gibt keinerlei festgelegte Anforderung an das Web Frontend hinsichlich der verwendeten Graphiken und des Aussehens. Der Kunde wünscht sich jedoch das Aussehen des Web-Frontends verändert werden kann. Dabei übergibt der Kunde der SE AG die gewünschten Anforderungen und die SE AG führt die notwendigen Arbeiten über die Wartungsschnittstelle durch.

Graphiken, Audio und Videodokumente werden in gängigen Formaten angezeigt (JPEG, RealAudio, RealVideo, MP3). Dabei können verschiedene Qualitäten je nach Bandbereitenkapazität gewählt werden.

Der Kunde stellte fest, daß eine akzeptable Antwortzeit beim Web Frontend schwer festzulegen ist. Es dürfte sich laut Kunden im 2-5 Sekunden Bereich bewegen.

Virtuelles Museum

Zusätzlich soll es möglich sein, aus einer Auswahl der erfaßten Daten ein virtuelles Museum zu erstellen. Der Zugriff erfolgt ebenfalls über das Web Frontend.

Es gibt dazu spezielle Teile bzw. Masken im Web Frontend um Virtuelle Museen zu erstellen und zu bearbeiten. Jeder Benutzer erhält auch hier die Rechte je nach den zugewiesenen Benutzergruppen.

Der Aufbau eines virtuellen Museeums orientiert sich am real exisitieren Vorbild. Dabei wird die Struktur mit Stockwerken, Räumen und Virtinen berücksichtigt. In einem Stockwerk werden, zum Beispiel, nur Fibeln ausgestellt. Die Räume enthalten enthalten die Fibeln bestimmter Epochen, während der genaue zeitliche Ablauf durch die Anordnung der Vitrinen wiedergespiegelt wird.

Artefakte sollen gruppiert (virtuelle Vitrinen) und beliebig kombiniert werden können (virtuelle Räume). Es soll möglich sein, über den bisherigen archäologischen Datenbestand hinaus, Informationen zu den Schauräumen und ausgestellten Objekten anzugeben und darzustellen.

Das System soll dem Entwickler Hilfsmittel zur Verfügung stellen und Gestaltungsmöglichkeiten offen lassen, um ein augenfälliges Design zu ermöglichen. Es ermöglicht das ein berechtigter Benutzer auf einfache Art und Weise ein Virtuelles Museum erstellen kann.

Es ist möglich mehrere virtuelle Mueseen anzulegen und diese über das Web Frontend zu verwalten. Analog zum Anlegen von neuen archäologischen Objekten erhält auch hier der Ersteller bestimmte Rechte (Default Rechte) und kann diese gegebenenfalls abändern. Auch hier wird der Ersteller via Mail verständigt wenn ein anderer, berechtigter Benutzer die Daten ändert.

Das System soll es weiter ermöglichen virtuelle Touren durch ein Museeum zusammenzustellen. Dies kann sowohl manuell als auch automatisch geschehen. Diese Touren werden den Besuchern des virtuellen Museums beim Betreten dieses angeboten. Der Besucher hat nun die Möglichkeit sich eine dieser Touren anzusehen oder sich frei durch das Museeum zu bewegen.

Der Kunde begrüßt die Möglichkeit eines "Guides " (Suchfunktion / Orientierungshilfe) bei einem virtuellen Mussum. Dieser könnte z.B. Informationen zu einembestimmten Artefakt geben oder auf den Museums-Webshop verweisen. Eine Suchmöglichkeit im virtuellen Museum soll eingeführt werden.

Es werden keine Links dargestellt werden welche aus dem Museum verweisen. Dies betrifft insbesondere auch Links welche zu bestimmten Objekten als Attribut abgelegt wurden und aufgrund einer entsprechenden Rechtevergabe angezeigt werden könnten.

Der Ersteller eines Museums, Raumes oder Vitrine ist sowohl von Seiten der Rechtevergabe, als auch legislativ für den Inhalt verantwortlich. Dies gibt insbesondere für die öffentliche Darstellung des Museums im Web oder auf CD-ROM.

Die virtuellen Museen soll sowohl im Web besucht als auch auf CD-ROM gespeichert oder in Katalogform dargestellt werden können.

Der Benutzer kann sich die Form des Virtuellen Museums beim Besuch auszusuchen Z.B. Ansehen im Web, Download als Katalog (z.B. PDF),... .

Sicherheit

Die Verbindung mit dem Web Frontend erfolgt über das HTTPS Protokoll. Alle Daten werden dabei bei der Übertragung verschlüsselt. Sämtliche weiteren Sicherheitsvorkehrungen werden auf Hardware Ebene realisiert und werden als gegeben angenommen.

Die Anmeldung erfolgt über einen entsprechen Dialog im Web Frontend. Auch hier werden die Daten natürlich wie vorher aufgeführt verschlüsselt übertragen. Nach einem mehrmaligen fehlgeschlagenen Anmeldeversuch wird der entsprechende Benutzer sowie ein Systemoperator via E-Mail darüber informiert.

2.1.6 Anforderung an die Entwicklungsumgebung

- Rational Unified Prozess wird als Grundlage zur Systementwicklung verwendet.
- XML soll eingesetzt werden.
- Geeignete Hilfsmittel zur Wartung und Weiterentwicklung von Datenbanken und Programmen sollen entwickelt werden. Die SE AG soll nach Entwicklungabschluß als Application Service Provider auftreten.

2.2 Systemeinsatz und Systemumgebung

2.2.1 Systemeinsatz

Folgende Punkte muß das System ermöglichen:

Erfassung/Verwaltung der archäologischen Daten

Das System dient dazu archäologische Objekte und zugehörige Daten zu erfassen. Die archäologischen Objekte lassen sich in 3 verschiedene Typen unterteilen

1. Artefakte

   Jedes Artefakt hat immer folgende Attribute

- Inventarnummer
  - Bezeichnung
  - Dimension

folgende Attribute können verwendet werden

- Versicherungswert
  - Lagerinformation

zusätzlich können zu jedem Artefakt

- Videodaten
  - Audiodaten
  - Textdaten
  - Literaturverweise

abgelegt werden. Diese werden logisch getrennt abgelegt und können mit einem oder mehreren Objekten in Beziehung gesetzt werden (Verweis). Zusätzlich kann jedem Artefakt mit einem Grab bzw. einer Siedlung in Beziehung gesetzt werden (Fundort). Wenn keine solche Beziehung gefunden werden kann erscheint es sinnvoll einen Fundort für das Artefakt anzugeben.

Vorschläge für weitere Attribute sind:

- Entdecker
  - Erfasser
  - Fundort
  - Alter

1. Gräber

   Jedem Grab werden folgende Attribute abgelegt

- Bezeichnung

- Inventarnummer (wenn möglich)
- Dimension

folgende Attribute können verwendet werden

- Versicherungswert
  - Lagerinformation

zusätzlich können zu jedem Grab

- Videodaten
  - Audiodaten
  - Textdaten
  - Literaturverweise

abgelegt werden. Diese werden wir bei den Artefakten logisch getrennt abgelegt und können mit einem oder mehreren Objekten in Beziehung gesetzt werden (Verweise). zusätzlich kann jedes Grab mit einer Siedlung in Beziehung gesetzt werden. Wenn keine solche Beziehung gefunden werden kann erscheint es auch hier sinnvoll einen Fundort für das Grab anzugeben.

Vorschläge für weitere Attribute sind:

- Entdecker
  - Erfasser
  - Fundort
  - Alter

1. Siedlungen

   Für jeder Siedlung werden folgende Attribute abgelegt

- Bezeichnung
  - Inventarnummer (wenn möglich)
  - Dimension

folgende Attribute können verwendet werden

- Versicherungswert
  - Lagerinformation

zusätzlich können zu jeder Siedlung

- Videodaten
  - Audiodaten
  - Textdaten
  - Literaturverweise abgelegt werden.

Diese werden wie bei den anderen Objekt-Typen logisch getrennt abgelegt und können mit einem oder mehreren Objekten in Beziehung gesetzt werden (Verweise). Zusätzlich zu der hirachischen Ordnung die sich durch die Beziehungen von Artefakten, Gräbern und Siedlungen ergibt kann es sinnvoll sein Querverweise zwischen den Objekten einzufügen. Damit kann z.B. eine Thematische Abhänigkeit modelliert werden.

Vorschläge für weitere Attribute sind:

- Entdecker
  - Erfasser
  - Ort
  - Alter

Die so erfassten Daten sollen beliebig (abhängig von den Rechten einer Benutzergruppe) bearbeitet werden können. Weiters ist eine Suchmöglichkeit in diesen Daten notwendig sowie die Möglichkeit Daten zu löschen.

Ein wesentlicher Punkt für den Erfolg und Verwendbarkeit des Systemes wird die Richtigkeit und Vollständigkeit (soweit ermittelbar) der obenaufgeführten Daten sein.

Virtuellen Museen

Es muss möglich sein virtuelle Museen

1. zu Erstellen

   Dabei ist die reale Struktur eine Museums mit Räumen und Etagen sowie Vitrinen zu Berücksichtigen.

2. zu Bearbeiten

3. zu Besuchen

   Dies kann sowohl interaktiv als auch in Form einer Führung geschehen.

4. zu Publizieren

   In Form einer CD-ROM oder als Katalog.

2.2.2 Systemumgebung

Alle Daten werden zentral verwaltet. Ein Zugriff auf die Daten erfolgt durch ein Web Frontend. Damit ergeben sich benutzerseitig folgende Anforderungen an die Systemumgebung:

- Benötigt einen Web-tauglichen Rechner

  Es wird ein entsprechender Rechner mit einer genügend leistungsfähigen Hardware, sowie der erforderlichen Software (lt. Kundenangabe nur Standartsoftware) benötigt.

- Benötigt einen (Web-) Zugang zum Internet

  Der Benutzer benötigt einen ausreichend schnellen und Web-tauglichen Zugang zum Internet erfolgen. Insbesondere muss dieser Zugang auch das HTTPS Protokoll unterstützen.

- Benötigt gegebenenfalls Zusatzsoftware/Hardware um Video/Audiodaten anzeigen zu können.

  Der Benutzer benötigt entsprechende Soft und Hardware um die Video und Audiodaten anzeigen zu können. Lt. Kundenangabe sollte dies nur Standartsoftware sein.

Für uns als Service Provider ergeben sich folgende Anforderungen:

- Geeignete Datenbank/Serverkapazitäten

  sowohl hinsichtlich des erforderlichen Speicherplatzes als auch hinsichtliche der Performance

- Geeignete Internetanbindung

  Wir müssen über eine geeignete Internetanbindung verfügen um alle Benutzer welche gleichzeitig das System benutzen eine akzeptable Antwortzeit zu garantieren. Weiters muss sichergestellt werden das die Bandbreite für die zu Übertragenden Daten ausreichend ist. Insbesondere betrifft dies Video und Audiodaten.

- Gegebenenfalls Bereitstellung von Reserve/Redundanten Kapazitäten

  zur Senkung der Ausfallzeiten. Der Kunde verlangt keine sehr hohe Zuverlässigkeit des Systems und erlaubt eine Ausfallzeit von maximal 2h pro Tag.

- Geeignete Software zum Erstellen des Systems bzw. um die Leistungen zu erbringen

  z.B. Zusatzsoftware um Video oder Audiodaten zu übertragen)

2.3 Benutzerschnittstelle

Die Benutzerschnittstelle besteht im gegenständlichen Projekt nur aus einem Web Frontend. Alle benötigten funktionalen Anforderungen an die Benutzerschnittstelle sollen, mit Ausnahme der Wartung des Systems, rein durch das Web Frontend erfüllt werden. Dieses Frontend soll nun folgende Schnittstellen realisieren:

Anmelden

Der jeweilige Benutzer kann sich mit Eingabe des Benutzernamens und des Passwortes an das System anmelden.

Registrieren

Der jeweilige Benutzer kann sich mit Eingabe seiner e-mail Adresse registieren lassen (Rechte werden ausgeweitet). Nach der Registrierung erhält er einen Benutzernamen und ein Passwort. Weiters werden Newsletters und Werbeemails an den registrierten Internetbenutzer verschickt.

Erfassung und Objekte zu löschen. Unter Objekte fallen archäologische Objekte (Siedlungen, Gräber, und Artfakte), Personen, Mediendateien (Texte, Bilder, Video- und Audiodateien).

Informationen Der Benutzer kann dann von einem gesuchten Objekt mit Hilfe von links zu anderen Objekten "weitersurven".

ErstellungvonVirtuellenMuseen

Der Benutzer kann virtuelle Museen erzeugen, bearbeiten und löschen. Es können auch Teile von verhanden Museen kopiert werden. Dabei stellt das System einen Assistenten für die Erstellung der Virtuellen Museen zur Verfügung.

ErstellungStandardDabei stellt das System Assistenten für die Erstellung der CD-ROM's und Katalogen zu verfügung (läuft weitesgehend automatisch ab).

BesuchQuotes erdsurven". Je nach Benutzer werden verschiedene Informationen nicht oder zusätzlich dargestellt.

VerwaltungStandardDabei können Benutzer zu Benutzergruppen hinzugefügt, bearbeitet oder gelöscht werden. Zusätzlich übernimmt der Administrator Sicherheitsfunktionen indem er per e-mail über falsche Passworteingaben informiert wird und dementsprechend reagieren kann.

WartungStandardIn diesem Fall werden alle erforderlichen Wartungsarbeiten, Pflegearbeiten und Erweiterungen durchgeführt. Gegebenenfalls ist das System an neue Anforderungen anzupassen.

2.4 Funktionale Anforderungen

Im folgenden werden die funktionalen Anforderungen aufgeführt. Dies ist zum derzeitigen Zeitpunkt nicht vollständig da die Benutzerschnittstelle noch nicht ausgearbeitet ist und damit die funktionalen Anforderungen an diese noch nicht spezifiziert werden können. Für jede Anforderung werden Prioritäten von 1-10 vergeben wobei die Priorität desto höher ist je kleiner die Zahl ist.

2.4.1 Ein- und Ausgabe

E1 Die (noch festzulegenden) Datenformate müssen bei der Eingabe akzeptiert werden.

Art: MUSS Priorität: 1

E2 Folgende Datenformate sollen bei der Eingabe akzeptiert werden.

Art: SOLL Priorität: 5

E3 Folgende Objekte müssen erfasst werden können:

- Artefakte
- Gräber
- Siedlungen

Art: MUSS Priorität: 1

E4 Folgende Daten müssen zu jedem Objekt eingegeben werden können:

- Bezeichnung
- Inventarnummer
- Dimension
- Versicherungswert
- Lagerinformation

Art: MUSS Priorität: 1

E5 Objekte müssen untereinander in Beziehung gesetzt werden können.
Art: MUSS Priorität: 1

E6 Virtuelle Mussen müssen erstellt werden können.
Art: MUSS Priorität: 1

E7 Generierung einer CD-ROM aus einem Virtuellen Museum
Art: MUSS Priorität: 1

E8 Generierung eines Kataloges aus einem Virtuellen Museum
Art: MUSS Priorität: 1

E9 Daten müssen durchsucht werden können.
Art: MUSS Priorität: 1

E10 Virutelle Mussen müssen besucht werden können.
Art: MUSS Priorität: 1

E11 Die Daten müssen in gängigen Formaten angezeigt werden können.
Art: MUSS Priorität: 1

E12 Die Daten sollen in gängigen (noch festzulegenden) Formaten ausgegeben (Exportiert) werden können
Art: SOLL Priorität: 5

E13 Benutzer müssen erstellt und geändert werden können.
Art: MUSS Priorität: 1

E14 Benutzergruppen müssen erstellt und geändert werden können.
Art: MUSS Priorität: 1

E15 Benutzergruppen müssen einzelnen Benutzern zugeordnet werden können.
Art: MUSS Priorität: 1

E16 Alle Eingaben von Benutzern sollen durch eine Multilevel-Undo-Funktion wieder rückgängig gemacht werden können.
Art: SOLL Priorität: 5

E17 Benutzer sollen sich beim System speziell registrieren können, um einen größeren Funktionsumfang erwerben zu können.
Art: SOLL Priorität: 7

E18 Vorgaben für das Aussehen des Webfrontendes müssen erstellt werden können, wobei weitere Änderungswünsche durch den Kunden von der SE AG jederzeit über die Wartungsschnittstelle durchgeführt werden können müssen.
Art: MUSS Priorität: 3

E19 Diese Vorgaben müssen mit Benutzergruppen verlinkt werden können.
Art: MUSS Priorität: 3

E20 Meta-Begriffe müssen eingegeben und Objekten zugeordnet werden können.
Art: MUSS Priorität: 3

2.4.2 Datenbanksystem

D1 Für alle archäologischen Objekte müssen folgende Daten abgelegt und verwaltet werden können:

- Bezeichnung
- Inventarnummer (wenn möglich)
- Dimension
- Versicherungswert
- Lagerinformation

sowie weitere, erst festzulegende Attribute.

Art: MUSS Priorität: 1

D2 Weiters müssen folgende Daten abgelegt, verwaltet und mit archäologischen Objekten in Verbindung gesetzt werden können

- Videodaten
- Audiodaten
- Textdaten
- Literaturverweise

Art: MUSS Priorität: 2

D3 Suche über alle Attribute muß möglich sein (einfache Suche)

Art: MUSS Priorität: 1

D4 Suche nach bestimmten Attributen bzw. Verknüpfungen dieser muß möglich sein (erweitere Suche)

Art: MUSS Priorität: 1

D5 Berücksichtigung der Zugriffsrechte beim Zugriff auf Daten

Art: MUSS Priorität: 1

D6 Virtuelle Museen müssen abgelegt werden können

Art: MUSS Priorität: 1

D7 Touren durch Mussen müssen abgelegt werden können

Art: MUSS Priorität: 3

D8 Benutzerdaten/Rechte müssen verwaltet werden können

Art: MUSS Priorität: 1

D9 Eingabe Masken für das Web-Frontend müssen verwaltet werden können

Art: MUSS Priorität: 3

2.4.3 Systemfunktionalität

Funktionen welche das System zur Verfügung stellen muß um die geforderten Anforderungen zu erfüllen und die sich weder dem Datenbanksystem noch der Benutzerschnittstelle zuordnen lassen.

S1 Das System soll automatisch Führungen durch virtuelle Museen erstellen können.
Art: SOLL Priorität: 5

S2 Das System soll Neuzugänge von Benutzern verhindern, falls die in NP4 definierten Anforderungen überschritten werden.
Art: SOLL Priorität: 4

S3 Benutzer einer bestimmten Benutzergruppe soll bei Einbruchversuchen (z.B. falsche Eingabe von Passwörtern etc) verständigt werden.
Art: SOLL Priorität: 4

S4 Der Systemaustausch mit Schwestersystemen soll möglich sein.
Art: SOLL Priorität: 7

2.5 Nicht Funktionale Anforderungen

2.5.1 Hilfsmittel

Das System muss Mittel zur Verfügung stellen mit dessen Hilfe das System verwaltet werden kann. Dies kann z.B. umfassen:

- Backup der Daten
- Reorganisation/Optimierung der Datenbank
- Testprogramme und Testumgebungen

Eine genaue Spezifikation kann aufgrund von noch nicht feststehenden Daten hier nicht erfolgen.

NH1 Das Projektdesign erfolgt mit Hilfe des Rational Unified Process.
Art: MUSS Priorität: 2

2.5.2 Struktur

NST1 Die Datenhaltung erfolgt zentral.
Art: MUSS Priorität: 1

2.5.3 Performance

NP1 Die maximale Antwortzeit des Systemes sollte innerhalb von 2 - 5 Sekunden sein.

Art: SOLL Priorität: 5

NP2 Der Bandbreitebedarf zwischen Benutzer und System darf nur so groß werden das ein Standartbenutzer noch mit dem System arbeiten kann.

Art: MUSS Priorität: 3

NP3 Die Gestaltung des Web Frontends muß so erfolgen das ein durchschnittlicher Anwender keine Performanceprobleme bekommt.

Art: MUSS Priorität: 4

NP4 Das System muß 100 Internetbenutzer und 5 Bearbeiter gleichzeitig und ohne große Performanceeinbußen verwalten können.

Art: MUSS Priorität: 3

2.5.4 Zuverlässigkeit

NZ1 Das Datenbanksystem sollte fehlerredundant ausgelegt sein

Art: SOLL Priorität: 5

NZ2 Das Datenbanksystem sollte eine Wartung während des Betriebes erlauben

Art: SOLL Priorität: 5

NZ3 Das Datenbanksystem sollte während des Betriebes erweitert werden können (z.B. Hardware)

Art: SOLL Priorität: 10

NZ4 Das System muß auch bei einer Unterbrechnung der Verbindung zum Anwender (z.B. durch Anwenderseitige Hard oder Softwarefehler) eine konsistenten Zustand behalten.

Art: MUSS Priorität: 2

NZ5 Das System sollte über Hilfsmittel verfügen um eine Fehlbedienung zu vermeiden

Art: SOLL Priorität: 4

NZ6 Die Gesamtzeit der Ausfälle des Systems pro Tag muÿ 2 Stunden unterschreiten.

Art: MUSS Priorität: 4

2.5.5 Kosten

NK1 Die Kosten sollten relativ genau vorhersehbar sein/bleiben.

Art: SOLL Priorität: 4

2.5.6 Sicherheit

NS1 Die Daten müssen regelmässig gesichert werden

Art: MUSS Priorität: 1

NS2 Die Daten müssen verschlüsselt übertragen werden

Art: MUSS Priorität: 1

NS3 Das System muss sicherstellen das keine unberechtigen Personen auf Daten zugreifen können

Art: MUSS Priorität: 1

2.5.7 Wartbarkeit

Aus den Anforderungen an die Zuverlässigkeit ergibt sich Anforderung NZ2 und NZ3.

Weitere Anforderungen können noch nicht spezifiziert werden.

2.5.8 Flexibilität/Portabilität

NF1 Das System muß Grafiken (Bilder) im Format JPEG und GIF importieren und exportieren können.

Art: MUSS Priorität: 1

NF2 Das System muß es ermöglichen Daten mit einem Schwesternsystem (gleiche Soft und Hardware) auszutauschen.

Art: MUSS Priorität: 1

NF3 Das System sollte möglichst oft XML verwenden

Art: SOLL Priorität: 3

2.5.9 Design/Implemenationseinschränkungen

Derzeigt noch nicht festgelegt.

2.6 Fehlverhalten

Fehlverhalten wird hauptsächlich bei der Benutzerschnittstelle durch Bedienungsfehler der Benutzer auftreten. Da die funktionalen Anforderungen an diese noch nicht spezifiziert sind wird hier auch die Vorgehensweise bei einem Fehlverhalten noch nicht festgelegt. Im allgemeinen wird jedoch versucht werden solche Fehler abzufangen und den Benutzer darüber zu Informieren bzw. ihn gegebenenfalls zu einer neuen Eingabe auffordern.

Folgende Fehlerfällen können für obrige Anforderungen festgelegt werden:

E1-E2 Bei fehlerhaften Daten wird eine entsprechende Fehlermeldung ausgegeben. Der Benutzer hat die Möglichkeit den Vorgang zu wiederholen.

E3-E6 Wenn Fehler in den Angaben erkannt werden können wird der Benutzer darüber Informiert. Der Benutzer hat die Möglichkeit den Vorgang zu wiederholen.

E7-E8 Eine CD-ROM oder ein Katalog konnte nicht aus einem Virtuellen Museum erstellt werden. Es wird angenommen das über das Web-Frontend nur logisch korrekte virtuelle Museen erstellt werden können. Dieser Fehler kann deshalb nur durch einen internen Fehler oder durch einen Fehler beim Zugriff auf die CD-ROM bzw. das Medium auf dem der Katalog gespeichert/gedruckt wurde auftreten. Eine entsprechende Fehlermeldung wird ausgegeben.

E9-E12 Ein systeminterner Fehler ist aufgetreten. Der Benutzer sowie der Operator wird darüber in Kenntniss gesetzt.

E13-E15 Zwei Möglichkeiten:

- Wenn Fehler in den Angaben erkannt werden können wird der Benutzer darüber Informiert. Der Benutzer hat die Möglichkeit den Vorgang zu wiederholen.
- Ein Systeminterner Fehler ist aufgetreten. Der Benutzer sowie der Operator wird darüber in Kenntniss gesetzt.

D1-D8 Ein Datenbank interner Fehler ist aufgetreten. Es wird eine entsprechende Fehlermeldung angezeigt und der Operator benachrichtigt.

S1 Ein systeminterner Fehler ist aufgetreten. Der Benutzer sowie der Operator wird darüber in Kenntniss gesetzt.

NH1 Kein Fehlerfall spezifizierbar.

NST1 Kein Fehlerfall spezifizierbar.

NP1 Das System setzt die Funktion fort. Der Anwender und der Operator wird durch eine Nachricht davon in Kenntnis gesetzt. Der Operator hat den Fehler zu suchen bzw. bei einer Überlastung durch zu viele Anwender die Hardware/Software nach Absprache mit dem Kunden entsprechend zu erweitern.

NP2 Das System informiert den Anwender über diesen Zustand. Gegebenenfalls wird die Menge der zu übertragenden Daten dynamisch Reduziert (durch eine Reduktion der Qualität/Auflösung der Daten). Dies ist mit dem Kunden abzuklären.

NP3 Das System hat im Normalfall keine Möglichkeit festzustellen ob der Rechner des Anwenders mit der Darstellung überfordert ist. Es erfolgt keinerlei Reaktion durch das System.

NZ1-NZ3 Ein interner Fehler ist aufgetreten. Es wird eine entsprechende Fehlermeldung angezeigt und der Operator benachrichtigt.

NZ4 Wenn das System keinen konsistenten Zustand einnehmen kann ist das Web-Frontend zurückzusetzen und der Benutzer muss sich neu Anmelden.

NZ5 Wenn das System fehlerhafte Daten nicht abfängt sollte durch das Design der Datenbank sichergestellt werden das sich solche Daten nicht negativ auswirken.

NK1 Die Planung war fehlerhaft. Der Kunde ist über diesen Zustand sofort zu Informieren.

NS1-NS3 Designfehler beim Systemdesign. Diese dürfen nach überprüfen der Abnahmekriterien nicht auftreten.

NF1-NF2 Implementations oder Designfehler. Diese dürfen nach überprüfen der Abnahmekriterien nicht auftreten.

NF3 Kein Fehlerfall spezifizierbar.

2.7 Dokumentationsanforderungen

Es wird notwendig sein folgende Dokumente zu erstellen:

- Kurzanleitungen
  Um einen schnellen Einstieg in das System zu ermöglichen.
- Projektdokumentation
  Auf Basis des Rational Unifies Prozess
- Anwenderhandbücher
  Handbücher für die Anwender. Unterteilung nach verschiedenen Benutzertypen
  - Handbuch für Internetanwender
    Gegebenenfalls Online verfügbar.
  - Handbuch für "Erfasser" und "Wissenschaftliche" Anwender
  - Handbuch für die Erstellung von virtuellen Mussen
  - Handbücher für die Erstellung von CD-ROMs und Katalogen

Die Trennung der verschiedenen Handbücher hat unter anderem den Vorteil das Redundanzen großteils vermieden werden und jeder Anwender eine angepasste Dokumentation erhält.

- INTERNE DOKUMENTATION
  Erstellung von interner Dokumentation für die Operatoren des Systems innerhalb der SE AG. Dies umfasst auch eine umfangreiche Dokumentation des erstellten Sourcecodes.

- UNTERRICHTSMATERIAL
  Erstellung von Unterrichtsmaterial für eine eventuelle Anwenderschulung

2.8 Abnahmekriterien

Für die funktionellen Anforderungen sind bis jetzt folgende Abnahmekriterien spezifiziert:

E1-E15 Die Abnahmekriterien ergeben sich direkt aus der funktionellen Anforderung.

D1-D8 Abnahmekriterien ergeben sich direkt aus der funktionellen Anforderung.

S1 Es wird vom System eine Führung erstellen lassen. Diese wird zusammen mit dem Kunden getestet.

NH1 Nachweis durch den Vorweis einer entsprechenden Dokumentation.

NST1 Bestätigung durch die SE AG und optionale Überprüfung durch den Kunden vorort.

NP1 Viele Testanwender testen das System. Die Einhaltung der noch festzulegenden Grenzwerte wird kontrolliert. Wahrscheinlich ist ein Test mit einer genügen großen Anzahl von Testanwendern nicht möglich. In diesem Fall ist eine Hochrechnung erforderlich.

NP2-NP3 Mehrere Testanwender testen das System bezüglich der Performance. Die noch festzulegenden Grenzwerte müssen eingehalten werden.

NZ1 Ein vorgesehenes Hardwareversagen wird simuliert. Das System muss verfügbar bleiben.

NZ2-NZ3 Die Abnahmekriterien ergeben sich aud der funktionellen Anforderung

NZ4 Die Verbindung wird manuell unterbrochen. Die Konsistenz des Systemes wird dannach überprüft.

NZ5 Noch festzulegende Fehlverhalten werden simuliert. Das System muss diese abfangen.

NK1 Nicht festzulegen

NS1-NS3 Die Abnahmekriterien ergeben sich aus der funktionellen Anforderung.

NS1-NS2 Die Abnahmekriterien ergeben sich aus der funktionellen Anforderung.

NF1 Nicht festzulegen

Genaue Rahmenbedingungen für jeden Punkt sind unbedingt mit dem Kunden zu klären.

2.9 Risikoabschätzung (Aufgabe 2)

Aufgabe der Risikoanalyse ist es, Faktoren, die eine Gefahr für den Projekterfolg (die im Projektauftrag definierte Leistung in geplanter Zeit mit den geplanten Ressourcen im vorgegebenen Budget zu erbringen) darstellen, zu identifizieren, zu bewerten und entsprechende Gegenmaßnahmen vorzubereiten bzw. einzuleiten.

Folgene Risikogruppen können unterschieden werden:

2.9.1 Technische Risiken

- PROGRAMMFEHLER
  Durch dieses Projekt k"onnen in keiner Weise Personen gefährdet werden. Falls es zu Systemabstürtzen kommen sollte, gehen maximal die gerade eingeben Daten verloren, der Rest ist bereits im Datenbanksystem gespeichert und das System kann nach dem Neustart wieder verwendet werden. Das verwendete Datenbanksystem ist ein Produkt einer Fremdfirma gegen die wir im Schadesfalle Ersatzansprüche gelten machen können. Falls es zu Rechtsverletzungen oder Copyright Verletzungen kommen sollte übernimmen wir nur die Haftung, sofern sie unmittelbar der Ursacher für diese Verletztung ist. Für den Inhalt ist der Kunde bzw. der jeweilige Benutzer voll rechtlich verantworlich. Zudem ist bei der SE AG eine Abteilung extra für die Überprüfung rechtlicher Konsequenzen bei Programmfehlern spezialisiert. Für Rechtsverletzungen welche der Kunde selbst, durch die Eingabe geschützer oder sonstiger Daten, verursacht wird natürlich keinerlei Haftung übernommen.

- AUSFALLSICHERHEIT
  Hinsichtlich der Ausfallsicherheit ist zu überprüfen welche Kosten ein Systemausfall direkt oder indirekt verursacht. Dies ist mit den zu erwartenden Kosten für eine höhere Redundanz des Systemes abzuwiegen. Es ist sicherzustellen das die dem Kunden garantierte Verfügbarkeit auch gegeben ist da dieser sonst gegen uns Schadenersatzansprüche geltend machen kann.

- PERFORMANCE
  Es kann nicht genau vorhergesagt werden wie groß die Anzahl der Internetbenutzer werden wird. Daher kann es schnell zu Performanceeinbußen kommen. Dies führt u.a. zu einem Imageverlust unseres Kunden. Die SE AG wird zusammen mit dem Kunden das System für eine ausreichende Performance auslegen und gegebenenfalls rechtzeitig erweitern.

- SYSTEMDIMENSION
  Die Daten sprengen den vorhergesehenen Rahmen an Speicherplatz. Mit ensprechendem finanziellen Aufwand ist es möglich das System zur Laufzeit aufzurüsten. Es ist zu überlegen ob die dadurch anfallenden hohen Kosten

gerechtfertigt sind. Da von uns der Speicherplatz laut Angaben des Kunden dimensioniert wird liegt das Risiko hierbei beim Kunden.

- KOMPATIBILITÄT
  Unter Umständen sind aufgrund von Softwareneuentwicklungen Updates für Software von Dritthherstellern notwendig. Durch eine möglichst vorhersehende Auswahl der Software kann dieses Risiko verringert werden. Ein Restrisiko tragen wir

- SICHERHEIT
  Für Sicherheitsanforderungen verlassen wir uns auf Drittfirmen welche geeignete Software liefert. Diese sind mit entsprechenden Verträgen für diesen Punkt verantwortlich.

- DRITTFIRMA ERBRINGT ZUGESAGTE LEISTUNGEN NICHT
  Wir übernehmen gegenüber den Kunden das Risko, sichern uns aber mit entsprechenden Verträgen mit unseren Partnerunternehmen ab.

2.9.2 Rechtliche Risiken

- INHALTLICHE RECHTLICHE PROBLEME
  Für die Daten ist der jeweilige eingebende Benutzer verantwortlich. Dieser trägt das volle rechtliche Risiko für die von ihm eingegebenen Daten.

- LIZENZTECHNISCHE RISIKEN
  Für Lizenztechnische Risiken sind wir verantwortlich. Wir kaufen Lizenzen für Software von Dritthherstellern. Wenn diese nicht mehr ausreichen oder erneuert werden müssen sind wir dafür verantworlich. Durch eine entsprechende Planung kann dieses Risko nahezu ausgeschlossen werden.

2.9.3 Personelle Probleme

- ERFORDERLICHES PERSONAL KANN NICHT BESCHAFT WERDEN
  Die SE AG ist ein kompetentes Unternehemen und führt eine genaue Planung des Systems durch. Dadurch kann bereits am Anfang das erforderliche Personal relativ gut eingeschätzt werden. Kommt es jedoch aus irgendwelchen Gründen doch zu Personalproblemen gibt es Verträge mit Fremdfirmen, von welchen Personal "entliehen" werden kann.

- AUSFALL EINES WICHTIGEN KNOW-HOW TRÄGERS
  Ist sehr gering da die SE AG alle Mitarbeiter in das Projekt miteinbezieht und durch eine entsprechende Aufteilung der Arbeitsbereiche und Teamwork die Schäden durch das Ausfallen eines Know-How Trägers minimiert.

2.9.4 Finanzielle Probleme

- ZAHLUNG
  Der Kunde zahlt nicht, die Zahlung geht nur schleppend voran oder der Kunde setzt sich ab. Für diese Fälle ist die Rechtsabteilung der SE AG zuständig.

Chapter 3

Use Case Diagramm (Aufgabe 3)

3.1 Use Case Name

Die im folgenden erwähnten Actoren (*kursiv*), die mit den Use-Cases in Beziehung stehen, werden im nächsten Abschnitt näher erleutert. Der abstrakte Actor *Datenbanksystem* startet selbst keinen Use-Case, ist jedoch insofern bei jedem Use-Case beteiligt, als Informationen aus der Datenbank geholt bzw. in der Datenbank gespeichert werden.

Anmelden

Der use case kann entweder vom *Erfasser*, vom *Designer* oder vom *Internetbenutzer* gestartet werden. Der jeweilige Benutzer kann sich mit Eingabe des Benutzernamens und des Passwortes an das System anmelden.

Registrieren

Der use case wird vom *Internetbenutzer* gestartet. Der jeweilige Benutzer kann sich mit Eingabe seiner e-mail Adresse registieren lassen (Rechte werden ausgeweitet). Nach der Registrierung erhält er einen Benutzernamen und ein Passwort. Weiters werden Newsletters und Werbeemails an den registrierten Internetbenutzer verschickt.

Objekten erfassen

Der use case wird durch den *Erfasser* gestartet. Es werden Möglichkeiten geboten nach bereits vorhanden Objekten zu suchen und passende Objekte zu kopieren, ein neues Objekt vollständig neu zu erfassen, Links auf Objekte zu erstellen und Objekte zu löschen. Unter Objekte fallen archäologische Objekte (Siedlungen, Gräber, und Artfakte), Personen, Mediendateien (Texte, Bilder, Video- und Audiodateien).

Informationen abfragen

Der use case kann entweder vom *Erfasser*, vom *Designer* oder vom *Internetbenutzer* gestartet werden. Informationen können durch die erweiterte

Suche (nach mehreren Suchbegriffen gleichzeitig) oder durch die Standard-Suche abgefragte werden. Zusätzlich gibt es eine Meta-Suche. Der Benutzer kann dann von einem gesuchten Objekt mit Hilfe von links zu anderen Objekten "weitersurven".

Virtuellen Museen erstellen

Der use case wird vom *Designer* gestartet. Der Benutzer kann virtuelle Museen erzeugen, bearbeiten und löschen. Es können auch Teile von verhanden Museen kopiert werden. Dabei stellt das System einen Assistenten für die Erstellung der Virtuellen Museen zur Verfügung.

CD-ROMs und Kataloge erstellen

Der use case kann entweder vom *Designer* oder vom *Internetbenutzer* gestartet werden. Dabei stellt das System Assistenten für die Erstellung der CD-ROM's und Katalogen zu verfügung (läuft weitesgehend automatisch ab).

Virtuellen Museen besuchen

Der use case wird vom *Internetbenutzer* gestartet. Der Benutzer kann entweder mit Hilfe von Links oder mit Hilfe eines Guides durch das Museum "surven". Je nach Benutzer werden verschiedene Informationen nicht oder zusätzlich dargestellt.

Datenaustausch

Dieser use case wird vom *Schwestersystem* gestartet. Dabei besteht jedoch keine direkte Verbindung zum einem zweiten System, sonder die auszutauschenden Daten werden z.B. von einer Person exportiert und in das zweite System importiert. Es wird also eine physikalische Kopie der Datensätze angelegt (entspricht Entleihen).

Benutzer verwalten

Der use case wird vom *Administrator* gestartet. Dabei können Benutzer zu Benutzergruppen hinzugefügt, bearbeitet oder gelöscht werden. Zusätzlich übernimmt der Aministrator Sicherheitsfunktionen indem er per e-mail über falsche Passworteingaben informiert wird und dementsprechend reagieren kann.

System warten

Dieser use case wird vom *Systemadministrator* gestartet. In diesem Fall werden alle erforderlichen Wartungsarbeiten, Pflegearbeiten und Erweiterungen durchgeführt. Gegebenenfalls ist das System an neue Anforderungen anzupassen.

3.2 Actors

Erfasser (Person)

Ein Erfasser ist eine Person, die Daten (Objekte) erfasst.

Designer (Person)

Ist ein Benutzer welcher aus den vorhanden Daten virtuelle Museen, CD-ROM's und Kataloge erstellt.

Internetbesucher (Person)

Ein Internetbesucher ist eine Person, die Daten sucht und abruft.

Systemadministrator (Person)

Ein Systemadministrator ist eine Person welche alle erforderlichen Wartungsarbeiten beim System und der Datebank übernimmt.

Administrator (Person)

Ein Administrator ist eine Person welche alle Benutzer verwaltet. Er ordnet die Benutzer verschiedenen Benutzergruppen zu und gibt ihnen dadurch verschiedene Rechte. Zusätzlich übernimmt der Aministrator Sicherheitsfunktionen indem er per e-mail über falsche Passworteingaben informiert wird und dementsprechend reagieren kann.

Datenbanksystem (Abstrakt)

Das Datenbanksystem wird ebenfalls als eigener Actor spezifiziert, welcher nun mit der Benutzerschnittstellen (Web-Frontend) Daten austauschen kann.

Schwesternsystem (Abstrakt)

Ermöglicht den Austausch zu anderen Systemen, welche Hard- und Softwaremässig getrennt sind

Chapter 4

Projektdauer- und Kostenabschätzung (Aufgabe 4)

4.1 Gewichtung der Actors

Die Actors werden wie in Tab. 7.1 dargestellt gewichtet.

4.2 Gewichtung der Use Cases

Die Use Cases werden wie in Tab. 7.2 dargestellt gewichtet.

4.3 Gewichtung der technischen Faktoren

Diese Faktoren werden nun in Tab. 4.3 festgelegt.

Der Technical Factor berechnet sich wie folgt:

$$TFactor = \sum (TLevel) \cdot (Weight) = 0+3+4+1+5+0.5+2.5+8+5+5+3+0+2 = 39$$

Daraus wird nun der Technical Complexity Factor (TCF) berechnet:

Actor	Actortype	Factor
Erfasser	Komplex	3
Designer	Komplex	3
Internetbesucher	Komplex	3
Systemadministrator	Komplex	3
Administrator	Komplex	3
Datenbanksystem	Simple	1
Schwesternsystem	Avarage	2
	Summe	18

Table 4.1:

Use Case	Use Case Type	Factor
Anmelden	Simple	5
Registrieren	Simple	5
Objekte erfassen	Komplex	15
Informationen abfragen	Avarage	10
Virtuelle Museen erstellen	Komplex	15
CDROMs und Kataloge erstellen	Avarage	10
Virtuelle Museen besuchen	Avarage	10
Datenaustausch	Simple	5
Benutzer verwalten	Avarage	10
System warten	Complex	15
	Summe	100

Table 4.2:

Factor number	Factor description	Weight	TLevel
T1	Distributet system	2	0
T2	Response or throughput performance objectives	1	3
T3	End-user efficiency (online)	1	4
T4	Complex internal processing	1	1
T5	Code must be reusable	1	5
T6	Easy to install	0.5	1
T7	Easy to use	0.5	5
T8	Portable	2	4
T9	Easy to change	1	5
T10	Concurrent	1	5
T11	Includes special security features	1	3
T12	Provides direct access for thirt parties	1	0
T13	Special user training facilities are required	1	2

Table 4.3:

Factor Number	Factor description	Weight	FLevel
F1	Familiar with Rational Unified Process	1.5	3
F2	Application experience	0.5	5
F3	Object-oriented experience	1	5
F4	Lead analyst capability	0.5	4
F5	Motivation	1	5
F6	Stable requirements	2	4
F7	Part-time workers	-1	3
F8	Difficult programming lanugage	-1	2

Table 4.4:

$$TCF = 0.6 + (0.01 \cdot TFactor) = 0.6 + 0.39 = 0.99$$

Anschließend werden die Umgebungsfaktoren für das Entwicklerteam festelegt (siehe dazu Tab. 4.4)

Nun wird folgende Berechnung angestellt:

$$EFactor = \sum(ELevel) \cdot (Weight) = 4.5 + 2.5 + 5 + 2 + 5 + 8 - 3 - 2 = 22$$

Daraus wird nun der Enviromentfactor (EF) berechnet:

$$EF = 1.4 + (-0.03 \cdot EFactor) = 1.4 - 0.66 = 0.74$$

4.4 Berechnung der Use Case Points (UCP)

Aus Abschnitt 4.1 und 4.2 kann man den Unadjusted Use Case Point (UUCP) berechnen:

$$UUCP = \sum Weight_{Actors} + \sum Weight_{UseCases} = 18 + 100 = 118$$

Daraus wird nun der UCP wie folgt berechnet:

$$UCP = UUCP \cdot TCF \cdot EF = 118 \cdot 0.99 \cdot 0.74 = 86.4468$$

4.5 Abschätzung der Dauer und der Kosten des Projektes:

Als Dauer wurde vom Kunden ein invariabler Zeitpunkt festgelegt. Der Abgabetermin darf den **31. Jänner 2000** nicht überschreiten :-). Dieser Zeitpunkt wird auch von der SE AG angepeilt. Daraus ergiben sich nach heutigem Datum (16. November 2000) ca. 10 Arbeitswochen.

Es wird weiters ein Faktor von 20 Arbeitsstunden pro UCP angenommen. Daraus ergibt sich, daß die Arbeitszeit t_a zur Realisierung des gesamten Projektes $t_a = UCP * 20 = 86.4468 * 20 = 1728.936$ Stunden beträgt.

Eine Arbeitswoche t_w wird mit $t_w = 40$ Stunden veranschlagt. Daher kann ein Programmierer bis zum Abgabetermin $t_P = t_w \cdot 10 = 400$ Arbeitsstunden leisten. Daraus ergibt sich die Mindestanzahl der Angestellen $N_{min} = 5$.

Die Arbeiterkammer gibt als Richtwert einen zu verrechnenden Stundensatz von 1400S für einen Programmierer an. Die Personalkosten schlagen sich somit mit $Stundensatz * Studenanzahl = 1400 * 1728.936 = 2.420.600$ zu Buche.

Für technische Anschaffungen (Server, etc.) werden nochmals 1.000.000 Schilling verbucht, womit der Kunde mit Kosten von ungefähr 3.420.600 Schilling zu rechnen hat.

Chapter 5

Detailes Use Case Description (Aufgabe 5)

Es hat sich bei der Ausarbeitung herausgestellt, dass man die User Cases weiter unterteilen muss um deren Komplexitaet zu verringern.

5.1 Virtuelles Museum erstellen

Ausgearbeitest von Markus Spitzlinger

Use Case Name:

Virtuelle Museen Erstellen Der use case wird vom Designer gestartet. Der Benutzer kann virtuelle Museen erzeugen, bearbeiten und löschen. Es können auch Teile von verhanden Museen kopiert werden. Dabei stellt das System einen Assistenten für die Erstellung der Virtuellen Museen zur Verfügung.

Actors:

Designer

Preconditions:

Designer muß am System angemeldet sein und die Daten müssen bereits eingegeben sein.

Postconditions:

Das Museum wird gespeichert und für andere Benutzer zugänglich gemacht.

Extension Points:

ErrorHandling.

"Used" Use Cases:

InformationenAbfragen, Virtuelle-MuseenBesuchen

Flow of Events

1. Der use case beginnt mit der Selektion "Virtuelle-MuseenErstellen".
2. Der Benutzer gibt den Namen des zu erstellenden Museums ein, um dieses anzulegen.
3. Der Benutzer bestätigt diese Eingabe.
4. if (System findet Namen)
   (a) Das System meldet, das das Museum breits existiert.
   (b) while (kein neuer Name eingegeben)
      i. Der Benutzer gibt einen neuen Namen ein, unter welchem das Museum gespeichert wird.
      ii. Der Benutzer bestätigt die Eingabe.
      iii. Das System sucht nach dem eingegeben Namen.
   (c) end.
   (d) Das System kopiert das Museum und speichert es unter dem neuen Namen ab.
   (e) Das System öffnet den Assistenten zur Erstellung und Bearbeitung des Museums.
5. if (System findet keinen Namen)
   (a) Das System öffnet den Assistenten zur Erstellung und Bearbeitung des Museums.
6. Der Benutzer plegt die Attribute des Museums.
7. Der Benutzer sichert das Museum oder bricht die Verarbeitung ab.
8. if (Benutzer bricht die Verarbeitung ab)
   (a) Das System verlässt die Eingabemaske. Der use case ist beendet.
9. if (Benutzer sichert Museum)
   (a) Das System übergibt das Museum zur Speicherung an das DMBS und verläßt die Eingabemaske. Der use case ist beendet.

Pflege der Attribute des Museums:

1. if (Benutzer wählt Etage anlegen)
   (a) Benutzer gibt den Namen der Etage an
   (b) Benutzer bestätigt diese Eingabe
   (c) if (System findet Namen)

i. Das System meldet, das die Etage breits existiert.
ii. while (kein neuer Name eingegeben)
   A. Der Benutzer gibt einen neuen Namen ein, unter welchem die Etage gespeichert wird.
   B. Der Benutzer bestätigt die Eingabe.
   C. Das System sucht nach dem eingegeben Namen.
iii. end.
iv. Das System kopiert die Etage und speichert sie unter dem neuen Namen ab.
v. Das System zeigt die Eingabemaske zur Pflege der Attribute der Etage an.

(d) if (System findet keinen Namen)
   i. Das System zeigt die Eingabemaske zur Pflege der Attribute der Etage an.
(e) Der Benutzer sichert die Etage oder bricht die Verarbeitung ab.
(f) if (Benutzer bricht die Verarbeitung ab)
   i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribute eines Museums zurück.
(g) if (Benutzer sichert Etage)
   i. Das System übergibt die Etage zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Museums zurück.

2. if (Benutzer wählt Raum anlegen (ein Raum kann dabei einer Siedlung entsprechen))
   (a) Benutzer gibt den Namen des Raums an
   (b) Benutzer bestätigt diese Eingabe
   (c) if (System findet Namen)
      i. Das System meldet, das der Raum breits existiert.
      ii. while (kein neuer Name eingegeben)
         A. Der Benutzer gibt einen neuen Namen ein, unter welchem der Raum gespeichert wird.
         B. Der Benutzer bestätigt die Eingabe.
         C. Das System sucht nach dem eingegeben Namen.
      iii. end.
      iv. Das System kopiert den Raum und speichert ihn unter dem neuen Namen ab.
      v. Das System zeigt die Eingabemaske zur Pflege der Attribute des Raums an.
   (d) if (System findet keinen Namen)
      i. Das System zeigt die Eingabemaske zur Pflege der Attribute des Raums an.

(e) Der Benutzer sichert den Raum oder bricht die Verarbeitung ab.

(f) if (Benutzer bricht die Verarbeitung ab)

   i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Etage zurück.

(g) if (Benutzer sichert Raum)

   i. Das System übergibt den Raum zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Etage zurück.

3. if (Benutzer wählt Vitrine anlegen (eine Virtrine kann dabei einem Grab entsprechen))

   (a) Benutzer gibt den Namen der Vitrine an

   (b) Benutzer bestätigt diese Eingabe

   (c) if (System findet Namen)

      i. Das System meldet, das die Vitrine breits existiert.

      ii. while (kein neuer Name eingegeben)

         A. Der Benutzer gibt einen neuen Namen ein, unter welchem die Vitrine gespeichert wird.

         B. Der Benutzer bestätigt die Eingabe.

         C. Das System sucht nach dem eingegeben Namen.

      iii. end.

      iv. Das System kopiert die Vitrine und speichert ihn unter dem neuen Namen ab.

      v. Das System zeigt die Eingabemaske zur Pflege der Attribute der Vitrine an.

   (d) if (System findet keinen Namen)

      i. Das System zeigt die Eingabemaske zur Pflege der Attribute der Vitrine an.

   (e) Der Benutzer sichert die Vitrine oder bricht die Verarbeitung ab.

   (f) if (Benutzer bricht die Verarbeitung ab)

      i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Raums zurück.

   (g) if (Benutzer sichert Vitrine)

      i. Das System übergibt die Vitrine zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Raums zurück.

4. if (Benutzer wählt Artefakt einfügen)

   (a) Benuter gibt einen Suchbegriff ein.

   (b) Benutzer bestätigt die Eingabe.

   (c) Benutzer wählt gefundenes Artefakt aus.

(d) Benutzer bestätigt die Eingabe.
(e) Benutzer wählt darzustellende Attribute und aus und gibt eine Beschreibung des Artefakts ein.
(f) if (Benutzer wählt Verweis einfügen)
    i. Benuter gibt einen Suchbegriff ein.
    ii. Benutzer bestätigt die Eingabe.
    iii. Benutzer wählt gefundenes Objekt aus.
    iv. Benutzer bestätigt die Eingabe.
(g) Der Benutzer sichert das Artefakt oder bricht die Verarbeitung ab.
(h) if (Benutzer bricht die Verarbeitung ab)
    i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.
(i) if (Benutzer sichert Artefakt)
    i. Das System übergibt das Artefakt zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.

5. if (Benutzer wählt Etage aus (entspricht Bearbeitungsmodus))
    (a) Das System zeigt die Eingabemaske zur Pflege der Attribute der Etage an.
    (b) if (Benutzer wählt löschen aus)
        i. Das System liefert die Meldung "Artfakt <name> sicher löschen?".
        ii. if (Benutzer bestätigt die Löschung)
            A. Das Artefakt wird gelöscht.
            B. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.
        iii. if (Benutzer verwirft die Löschung)
            A. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske zur Pflege der Attribute eines Artefaktes zurück.
    (c) Benutzer ändert die Attribute.
    (d) Der Benutzer sichert die Etage oder bricht die Verarbeitung ab.
    (e) if (Benutzer bricht die Verarbeitung ab)
        i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribute eines Museums zurück.
    (f) if (Benutzer sichert Etage)
        i. Das System übergibt die Etage zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Museums zurück.

6. if (Benutzer wählt Raum aus (entspricht Bearbeitungsmodus))
    (a) Das System zeigt die Eingabemaske zur Pflege der Attribute des Raums an.

(b) if (Benutzer wählt löschen aus)

    i. Das System liefert die Meldung “Artfakt <name> sicher löschen?”.

    ii. if (Benutzer bestätigt die Löschung)

        A. Das Artefakt wird gelöscht.

        B. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.

    iii. if (Benutzer verwirft die Löschung)

        A. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske zur Pflege der Attribute eines Artefaktes zurück.

(c) Benutzer ändert die Attribute.

(d) Der Benutzer sichert den Raum oder bricht die Verarbeitung ab.

(e) if (Benutzer bricht die Verarbeitung ab)

    i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Etage zurück.

(f) if (Benutzer sichert Raum)

    i. Das System übergibt den Raum zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Etage zurück.

7. if (Benutzer wählt Vitrine aus (entspricht Bearbeitungsmodus))

    (a) Das System zeigt die Eingabemaske zur Pflege der Attribute der Vitrine an.

    (b) if (Benutzer wählt löschen aus)

        i. Das System liefert die Meldung “Vitrine <name> sicher löschen?”.

        ii. if (Benutzer bestätigt die Löschung)

            A. Die Vitrine wird gelöscht.

            B. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Raumes zurück.

        iii. if (Benutzer verwirft die Löschung)

            A. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske zur Pflege der Attribute einer Vitrine zurück.

    (c) Benutzer ändert die Attribute.

    (d) Der Benutzer sichert die Vitrine oder bricht die Verarbeitung ab.

    (e) if (Benutzer bricht die Verarbeitung ab)

        i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Raums zurück.

    (f) if (Benutzer sichert Vitrine)

        i. Das System übergibt die Vitrine zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte eines Raums zurück.

8. if (Benutzer wählt Artefakt aus (entspricht Bearbeitungsmodus)

(a) Das System zeigt die Eingabemaske zur Pflege der Attribute eines Artefaktes an.

(b) if (Benutzer wählt löschen aus)

  i. Das System liefert die Meldung “Artfakt <name> sicher löschen?”.

  ii. if (Benutzer bestätigt die Löschung)

    A. Das Artefakt wird gelöscht.

    B. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.

  iii. if (Benutzer verwirft die Löschung)

    A. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske zur Pflege der Attribute eines Artefaktes zurück.

(c) Benutzer ändert die Attribute.

(d) Der Benutzer sichert das Artefakt oder bricht die Verarbeitung ab.

(e) if (Benutzer bricht die Verarbeitung ab)

  i. Das System verlässt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.

(f) if (Benutzer sichert Artefakt)

  i. Das System übergibt das Artefakt zur Speicherung an das DMBS, verläßt die Eingabemaske und kehrt zur Eingabemaske Pflege der Attribte einer Virtrine zurück.

Activity Diagram

In Abbildung 5.1 ist ein wichtigste Teil des “Flow of Events” dargestellt, der Rest wurde auf Grund der Alternativität und der Komplexität nicht gezeichnet.

User Interface

Abbildung 5.2 zeigt die Haupfunktioenen dieses Use-Cases. Abbildung 5.3 zeigt einen Teil des Bearbeitungsmodusund des Löschmodus dieses Use-Case.

5.2 Benutzergruppen verwalten

Ausgearbeitest von Gerald Stieglbauer

***Hinweis*:** Im folgenden wird der Benutzer als Actor bezeichnet, um Verwechslungen des Begriffs „Benutzer” bei der eigentlichen Benutzerverwaltung mit dem Benutzer der die Benutzerverwaltung bedient zu vermeiden.

Use Case Name

Benutzer verwalten: Der use case wird vom *Administrator* gestartet. Dabei können Benutzer zu Benutzergruppen hinzugefügt, bearbeitet oder gelöscht werden. Zusätzlich übernimmt der Aministrator Sicherheitsfunktionen indem er per e-mail über falsche Passworteingaben informiert wird und dementsprechend reagieren kann.

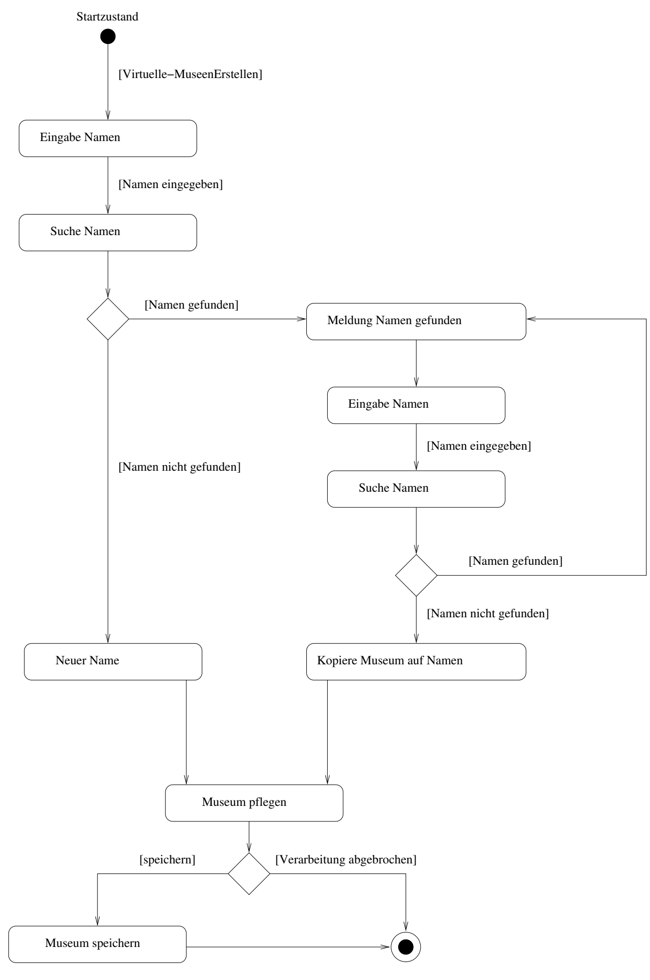

Figure 5.1: Activity Diagram (Virtuelle Museen erstellen)

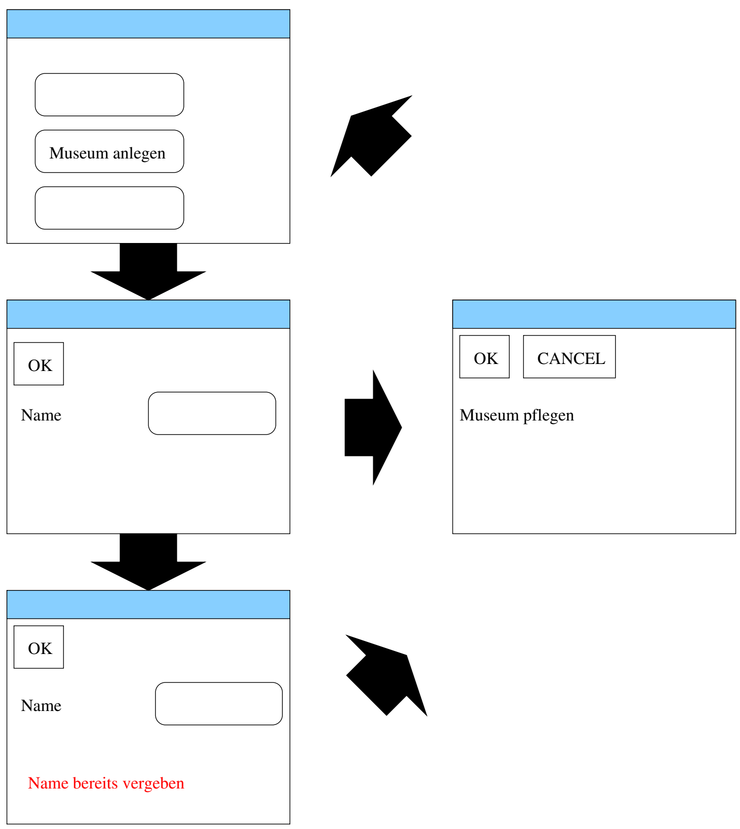

Figure 5.2: User Interface der Haupfunktionen (Virtuelle Museen erstellen)

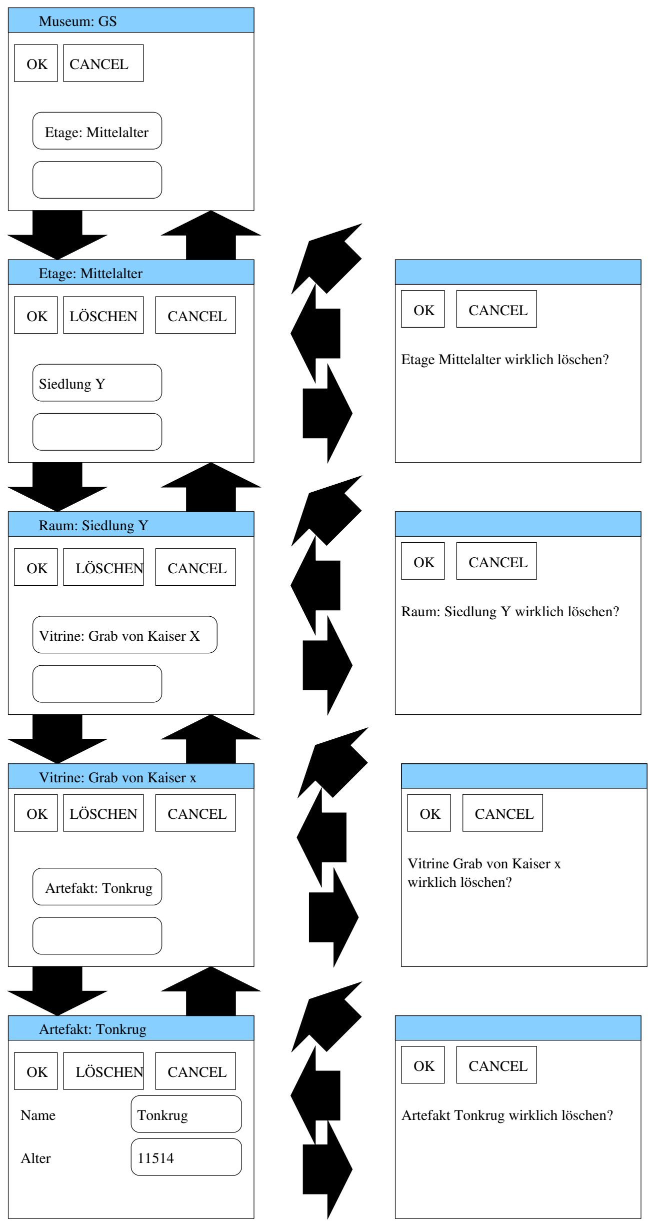

Figure 5.3: User Inerface des Bearbeitungs- und Löschmodus (Virtuelle Museen erstellen)

Actors:

- Administrator

Priority: 2

Status: Filled

Preconditions

- Der Benutzer der den Actor *Administrator* darstellt muß beim System angemeldet sein.
- Der Actor *Administrator* muß den Punkt „Benutzer verwalten" im Hauptmenü des Web-Frontends ausgewählt haben.

Postconditions

- Die Veränderungen in der Benutzerverwaltung wurden erfolgreich abgespeichert oder der Use-Case wurde vom Actor *Administrator* abgebrochen.

Extension Points

- Error-Handling

"Used" Use Cases

Es werden keine weiteren Use-Cases verwendet.

Flow of Events

1. Der Actor *Administrator* ist beim System angemeldet und wählt im Hauptmenu des Web-Frontends den Punkt „Benutzerverwaltung" aus.
2. Der Actor wählt aus zwischen
   - Benutzergruppen verwalten
   - Benutzer verwalten
   - Benutzerverwaltung verlassen
3. **if (**Benutzergruppen verwalten**)**
   (a) Es erscheint eine (möglicherweise leere) Liste der vorhandenen Benutzergruppen.
   (b) Der Actor kann aus dieser Liste einen Eintrag markieren und wählt weiters aus zwischen
      - Benutzergruppe erstellen (aktueller markierter Eintrag wird dafür nicht benötigt)
      - Benutzergruppe editieren
      - Benutzergruppe löschen
      - Menü verlassen

   (c) **if (**Benutzergruppe erstellen**)**

i. Es erscheint eine Auflistung der Benutzergruppen-Attribute (inkl. Benutzergruppen-Name) mit angrenzendem leeren Datenfeld
ii. Der Actor füllt die leeren Datenfelder aus und bestätigt seine Eingabe.
iii. Das System bestätigt dem Actor die korrekte Eingabe und speichert die neue Benutzergruppe.

(d) **if (**Benutzergruppe editieren**)**

i. Es erscheint eine Auflistung der Benutzergruppen-Attribute (inkl. Benutzergruppen-Name) mit angrenzendem Datenfeld, das die bisher gespeicherten Daten enthält.
ii. Der Actor ändert die gewünschten Datenfelder und bestätigt seine Eingabe.
iii. Das System bestätigt dem Actor die korrekte Eingabe der Veränderungen und speichert die neue Benutzergruppe.

(e) **if (**Benutzergruppe löschen**)**

i. Es erscheint ein Dialog als Sicherheitsabfrage, ob der markierte Eintrag gelöscht werden soll.
ii. **if (**Benutzer wählt *„Ok”***)**
    A. Das System bestätigt dem Actor das Löschen des Eintrags und löscht den Eintrag aus der Benutzergruppen-Liste.

(f) **if (**Menü verlassen**)**

i. Es wird zu Punkt 2. zurückgekehrt.

4. **if (**Benutzer verwalten**)**

(a) Es erscheint eine (möglicherweise leere) Liste der vorhandenen Benutzergruppen.

(b) Der Actor kann aus dieser Liste einen Eintrag markieren und wählt weiters aus zwischen

- Benutzer erstellen
- Benutzer editieren
- Benutzer löschen
- Menü verlassen

(c) **if (**Benutzergruppe erstellen**)**

i. Es erscheint eine Auflistung der Benutzer-Attribute (inkl. Benutzer-Name, Passwort) mit angrenzendem leeren Datenfeld. Weiters steht eine leere Liste von Benutzergruppen als Attribut zur Verfügung
ii. Der Actor füllt die leeren Datenfelder aus und bestätigt seine Eingabe.
iii. Beim Ausfüllen der (leeren) Liste für die Benutzergruppe kann der Actor wie in ähnlich wie in 3. beschrieben entweder zwischen
    - Neue Einträge hinzufügen oder
    - Bestehende Einträge löschen

auswählen.

iv. **if** (Neue Einträge hinzufügen)

A. Es erscheint eine Liste von vorhandenen Benutzergruppen aus der der Actor einen Eintrag auswählen kann.

B. Der Eintrag wird der Liste für den aktuellen Benutzer für Benutzergruppen hinzugefügt.

v. **if** (Bestehende Einträge löschen)

A. Es erscheint ein Dialog, ob der betreffende Eintrag tatsächlich gelöscht werden soll.

B. Im Falle einer Bestätigung von A. wird der betreffende Eintrag gelöscht.

vi. Das System bestätigt dem Actor die korrekte Eingabe und speichert den neuen Benutzer.

(d) **if** (Benutzer editieren)

i. System verhält sich wie in 4 (c). beschrieben, mit der Ausnahme, daß Benutzter-Attribute von bestehenden Benutzern nur verändert werden, jedoch kein neuer Benutzer angelegt wird.

(e) **if** (Benutzer löschen)

i. Es erscheint ein Dialog als Sicherheitsabfrage, ob der markierte Eintrag gelöscht werden soll.

ii. **if** (Actor wählt *„Ok"*)

A. Das System bestätigt dem Actor das Löschen des Eintrags und löscht den Eintrag aus der Benutzer-Liste.

(f) **if** (Menü verlassen)

i. Es wird zu Punkt 2. zurückgekehrt.

5. **if** (Benutzerverwaltung verlassen)

(a) Der Use-Case terminiert.

Activity Diagram

Dieses Diagramm ist der Abb. 5.4 zu entnehmen.

User Interface

Auch dies wird in Abb. 5.5graphisch dargestellt.

Secondary Scenarios

Der Actor kann zu jeder Zeit mit dem Abbruch-Button den Vorgang abbrechen und kehrt zum Ausgangszustand zurück.

5.3 Virtuelle Museen besuchen

Augearbeitet von Wiesbauer Gerald

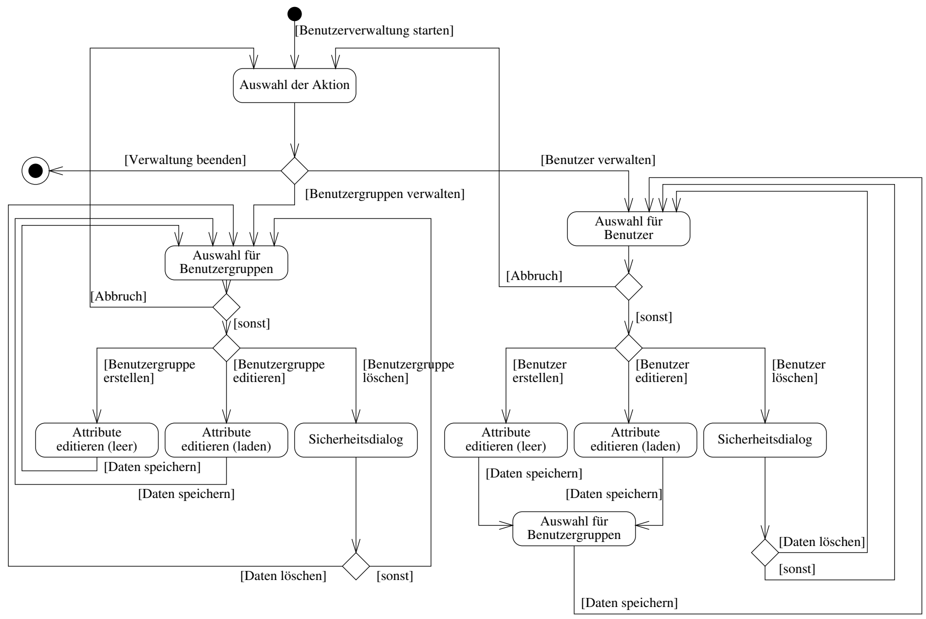

Figure 5.4: \langamericanActivity Diagram (Benutzer verwalten)

Use Case Name

Hinweis: mit Besucher ist ein Benutzer der Virtuellen Museums gemeint.

Virtuelles Museum besuchen: Der use case wird vom *Internetbenutzer* gestartet. Der Benutzer kann entweder mit Hilfe von Links oder mit Hilfe eines Guides durch das Museum "surven". Je nach Benutzer werden verschiedene Informationen nicht oder zusätzlich dargestellt.

Actors:

- Besucher
- Datenbank

Priority:

1

Status:

Filled

Preconditions:

Es muessen bereits Virtuelle Museen erstellt worden sein. Dies setzt voraus dass zumindest einmal der Use Case Virtuelle Museen Erstellen durchlaufen wurde.

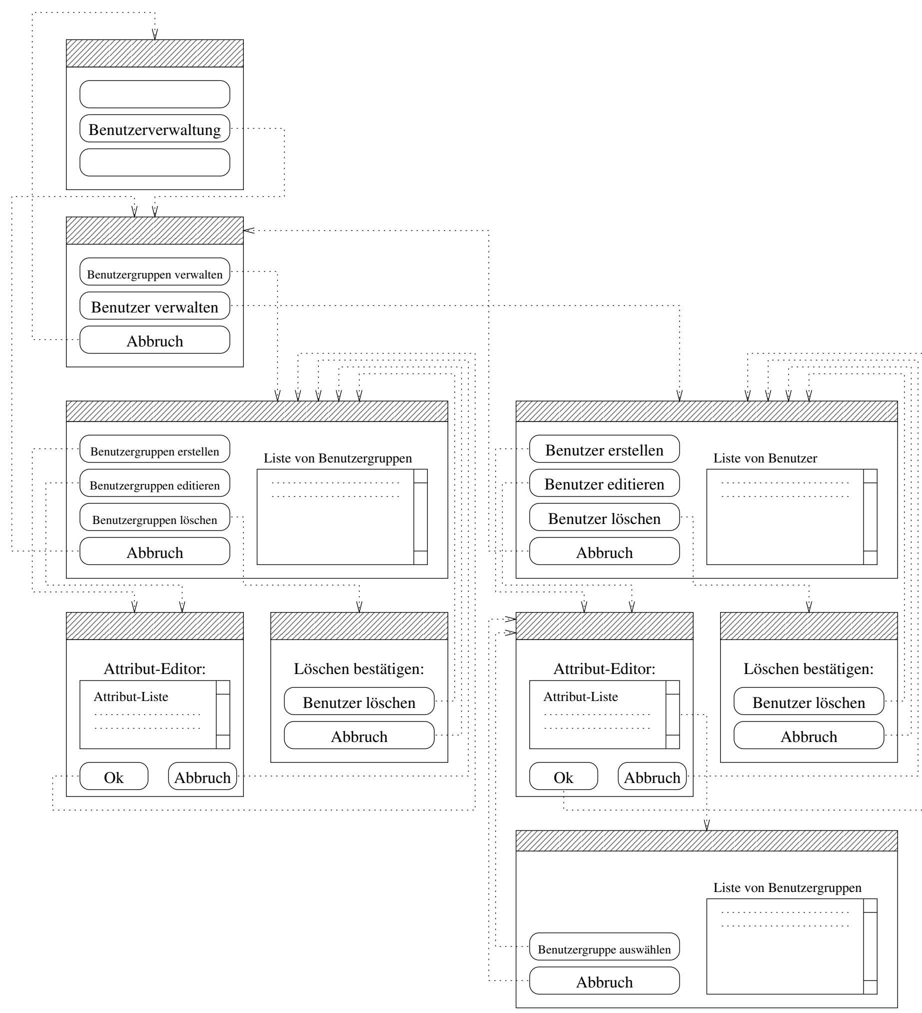

Figure 5.5: User Interface (Benutzer verwalten)

Postconditions:

Die Datenbank wurde mehrmals abgefragt. Damit ist gemeint dass mehrmals Anfragen an die zentrale DB gestellt wurden um ueberhaupt die Daten fuer das Virtuelle Museeum zu bekommen

Extenion Points:

Error Handling. Falls irgendwo ein Fehler auftritt sollte dem Besucher ein Standardtext praesentiert werden.

Used Use Cases:

Info Abfragen. Da der Besucher sicher mehrmals nach Stichworten/Begriffen suchen kann/wird.

Flow of Event:

Hinweis: Beim Flow of Event wurden einige Punkte zusammengefasst die im Activity-Diagramm wesentlich genauer beschrieben werden. Da dies aus Uebersichtsgruenden erforderlich ist.

1. Der Use Case beginnt mit dem Eingeben der URL des Virtuellen Museums
   (a) Der Besucherr gibt die URL im Browser ein
   (b) Besucher bestaetigt die Eingabe
2. Die Startseite des Virtuellen Museums wird gelagen und der Besucher kann nun eine der vorgegebenen Optionen waehlen
   - Daten suchen (auf 2 Arten)
   - Virtuelles Museum durchwandern
   - An einer Virtuellen Fuehrung teilnehmen
3. Der Besucher kann auf folgende Weise verfahren, es handelt sich dabei um eine grobe Einteilung die der Designer implementieren muss. Es koennen aber jederzeit weitere Optionen durch den Designer hinzugefuegt werden.
   (a) IF (Stichwortsuche)
      i. Das System preasentiert dem Besucher eine Eingabemaske in der er ein Stichwort nach dem spaeter gesucht wird eingeben kann.
      ii. IF (Der Beucher gibt nichts ein aber bestaetigt)
         A. Das System kehrt zu der Eingabemaske zurueck und es wird keine Fehlermeldung ausgegeben.
      iii. Der Besucher gibt den Suchberiff (Stichwort) ein
      iv. Der Besucher betaetigt die Eingabe
      v. IF (Daten zu einem Suchbegriff gefunden wurden)
         A. Die Daten in geeigneter Form (Link + Kurzbeschreinung) darstellen, es koennen dabei auch Bilder zusaetzlich zum Text praesentiert werden

B. Der Besucher kann nun mit Hilfe dieser Links genauere Info ueber ein Artefact erhalten das mit der Stichwortsuche gefunden wurde

vi. Wenn der Besucher die Information zu einem Artifact gelesen hat kann er mit einem "Back"-Button wieder zu den Suchergebnissen zurueckkehren und dort weitere Links checken.

(b) IF (META-Suche) es handelt sich dabei um vorgegeben META Begriffe (Links) wie zB Epochen.

i. Dem Besucher werden auf einer Maske vorgegebene Metabegriffe praesentiert zu denen manuell "Suchergebnisse" erstellt wurden.

ii. IF (Der Besucher waehlt einen dieser Links aus)

A. Es werden zu dem Metabegriff Informationen in aehnlicher Form wie in der Stichwortsuche dargestellt. Bei den Suchergebnissen handelt es sich um eine "grobe Einteilung" von Informationen ueber Artifacten die mit dem Metabegriff zu tun haben. Mit "grober Einteilung" ist gemeint dass Artifacte mit aehnlichen Eigenschaften zu einer Gruppe zusammengefasst werden.

iii. IF (Der Besucher waehlt einen dieser "groben" Links)

A. Nach dem Waehlen eines dieser "groben" Suchergebnissen gibt es eine weitere, feinere Unterteilung unter der der Besucher waehlen kann. Es werden nun die einzelnen Artifacte die zuerst zu einer Gruppe zusammengefasst wahren einzeln dargestellt mit Link und Kurzbeschreibung.

iv. IF (Der Besucher waehlt nun einen Link zu einem Artifact aus)

A. Jetzt werden genaue Informationen zu diesem Artifact angeboten

v. Der Besucher kann Jederzeit wieder mit einem "Back" Button zu den Suchergebnissen zuruekkehren

(c) IF (Durch das Virtuelle Museem auf den angebotenen Links durchwandern)

i. Der Besucher kommt automatisch in einem Virtuellen Eingangsbereich in diesem Eingangsbereich sind Links auf alle angrenzenden Raeume und Etagen.

ii. IF (Der Besucher waehlt einen Link auf eine andere Etage)

A. Jetz befindet sich der Besucher im Eingangsbereich der jeweiligen Etage

iii. IF (Der Besucher waehlt eine Link in einen angrenzenden Raum)

A. Jetzt kommt der Besucher in einen der Raeume des Museums und kann sich dort umsehen

B. Es gibt dort eine Reihe von Gegenstaenden die sich der Besucher ansehen kann wie etwar Virtuelle Vitrinen und Schautafen

iv. IF (Der Besucher sieht sich eine virtuelle Vitrine an)

A. Es werden nun die Artifacte die sich ein einer dieser Virtuellen Vitrinen befinden dargestellt

B. Man kann zu jedem diese Artifacte genauere Informationen abfragen wenn das gewuenscht ist
C. Mit einem Back-Button kann der Besucher wieder in den Raum zurueckkehren

v. IF (Der Besucher sieht sich eine Schautafen an)
A. Die Schautafel wird als Text am Bildschirm ausgegeben
B. Auch hier sind Hyperlinks moeglich
C. Mit dem Back-Button kann der Besucher wieder in den Raum zurueck

vi. Damit sich der Besucher im Virtuellen Museum nicht “verirrt” gibt es Links mit denen man wieder in den Eingangsbereich zuruekkehren kann.

(d) IF (Am einer Virtuellen Fuehrung teilnehmen)
i. Es erscheint nun eine Seite in der man die Kategorie der Fuehrung auswaehlen kann (zB.: Mittelalter, Steinzeit, ...)
ii. Diese Kategorien werden in Form von Link und Kurzbeschreibung dargestellt
iii. IF (Besucher waehlt ein Kategorie aus)
A. Nun kann sich der Besucher auf einem vogegeben Pfad durch das Museum klicken
B. Die Fuehrung ist so gestalltet dass sich der Besucher nur nach ”vorne” und ”zurueck” bewegen kann.
C. Dabei kommt man in Raeume die fuer die Virtuell Fuehrung zusammengestellt wurden.
D. Diese Raeume beinhalten wieder Vitrinen und Schautafen

Activity Diagramm

In Abb.:5.6 ist das Activity Diagramm zu sehen

Das User Interface

In Abb.:5.7 ist das User interface zu sehen

Scondaery Scenarios:

Der Besucher kann jederzeit mit einem Back-Button zum ”Ausgansort” zurueckkehren

5.4 Informationen abfragen

Ausgearbeitet von Christopher Rettenbacher

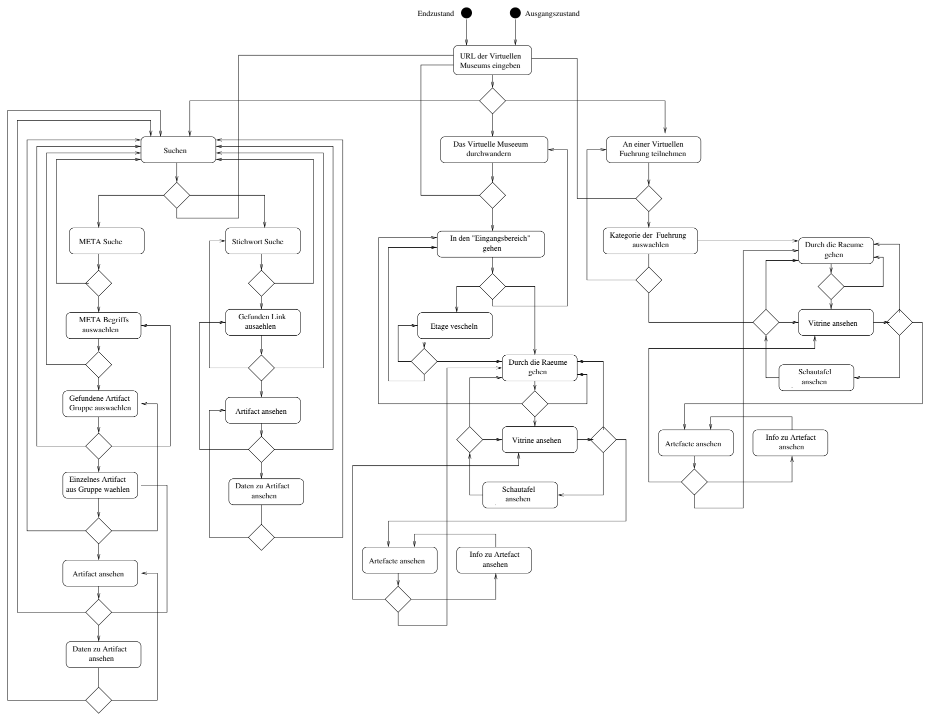

Figure 5.6: Activity Diagram (Virtuelle Museen besuchen)

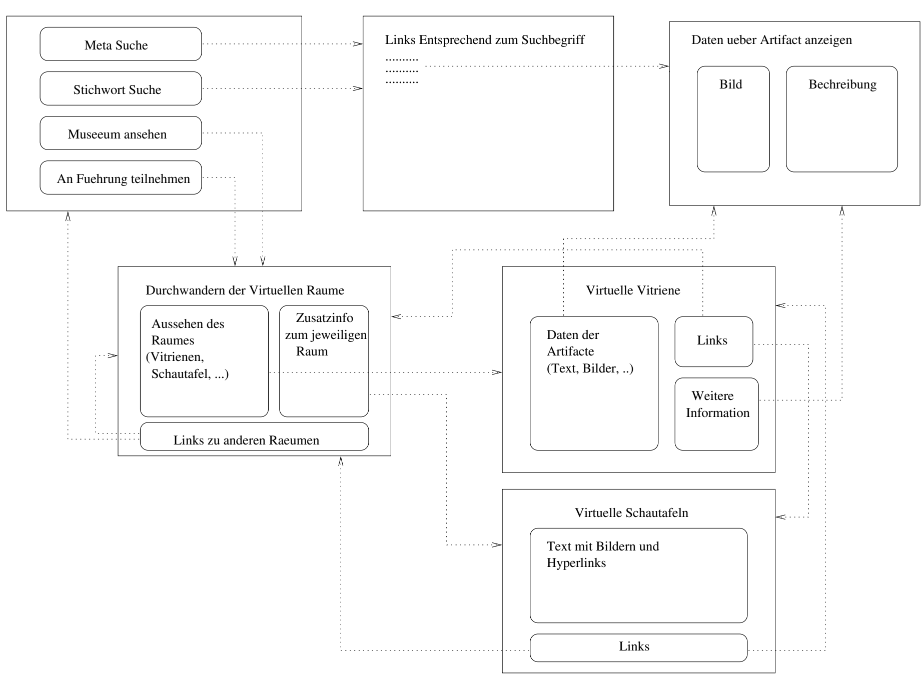

Figure 5.7: User Interface (Virtuelle Museen besuchen)

Use Case Name

Information Abfragen: Informationen könne sowohl direkt auf dem Datenbestand (mittels geeigneter Suchmodi) als auch indirekt über das virtuelle Museum abgefragt werden. Die Definition der Zielmenge erfolgt entweder über die Einschränkung durch geeignete Suchbegriffe, oder durch manuelle Orientierung entlang der zwischen den Objekten definierten Beziehungen. Auch die Auflistung aller Objekte eines Typs ist möglich.

Actors

- Erfasser
- Designer
- Internetbesucher
- Systemadministrator

Priority

Die Priorität des Use Cases wird mit der höchsten Priorität der durch das Use Case umfaßten Systemanforderung gleichgesetzt.
Art: MUSS
Priorität: 1

Status

Filled.

Preconditions

- Benutzer muß am System angemeldet sein.

Postconditions

- Benutzer hat Zugriff auf Information erhalten.

Extension Points

- Error Handling

"Used" Use Cases

keine bekannt

Flow of Events

1. Der Use Case beginnt mit der Selektion "InformationAbfragen"
2. Der Benutzer wählt zwischen "einfacherSuche" und "erweiterter Suche"
3. if (Auswahl = "einfacheSuche")
   (a) Der Benutzer gibt selbst einen Suchbegriff ein.

4. if (Auswahl = "erweiterteSuche")
   (a) Das System zeigt eine Maske mit vordefinierte Auswahlmöglichkeiten zur Eingrenzung des Suchbegriffes.
   (b) Der Benutzer grenzt das Ergebnis seiner Suche ein.
   (c) Der Benutzer bestätigt die Eingabe.
5. Der Benutzer gibt die Anzahl der maximal anzuzeigenden Treffer an. [bei betimmten Benutzern bereits vorgegeben]
6. Das System ermittelt anhand der Zugriffsrechte des Benutzers den freigegebenen Bereiche der Datenbank. [Bei der Entscheidung ob bestimmte Attribute durchsucht werden, orientiert sich das System an den getroffenen Voreinstellungen für bestimmte Benutzergruppen.]
7. while ([Trefferzahl $<$ maximal anzuzeigende Trefferanzahl] und nicht [gesammter freigegebener Datenbestand durchsucht])
   (a) Das System durchsucht die freigebenen Bereiche der Datenbank nach dem Suchbegriff.
8. if (Suchbegriff gefunden)
   (a) for (Trefferanzahl)
      i. if (Zugriff auf Suchbegriff erlaubt) [Bei der Entscheidung ob bestimmte Treffer angezeigt werden, orientiert sich das System an den auf Attributebene festgelegten Zugriffsrechte.] Objekt (Treffer) auf Trefferliste angefügt.
   (b) System zeigt Trefferliste an
   (c) Bei Auswahl eines Listenelements durch den Benutzer, zeigt das System den entsprechenden Datensatz an. [einzellne Attribute werden jedoch nur entsprechend der Zugriffsrechte des Benutzers angezeigt]
9. else
   (a) Der Benutzer wird darauf hingewiesen, daß kein seinen Kriterien entsprechender Suchbegriff gefunden wurde.
10. Use Case Ende

Activity Diagram

Das Activity Diagramm ist in Abb.:5.8 zu sehen

User Interface

Das User interface ist in Abb.: 5.9 zu sehen

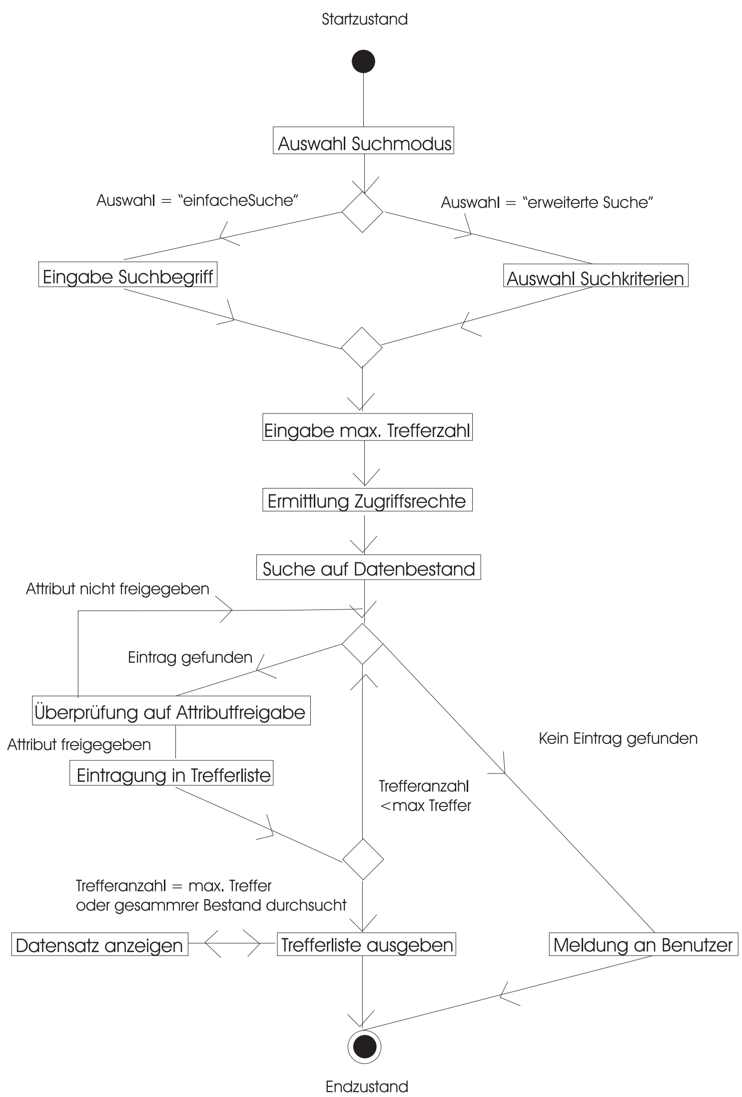

Figure 5.8: Activity Diagram (Information abfragen)

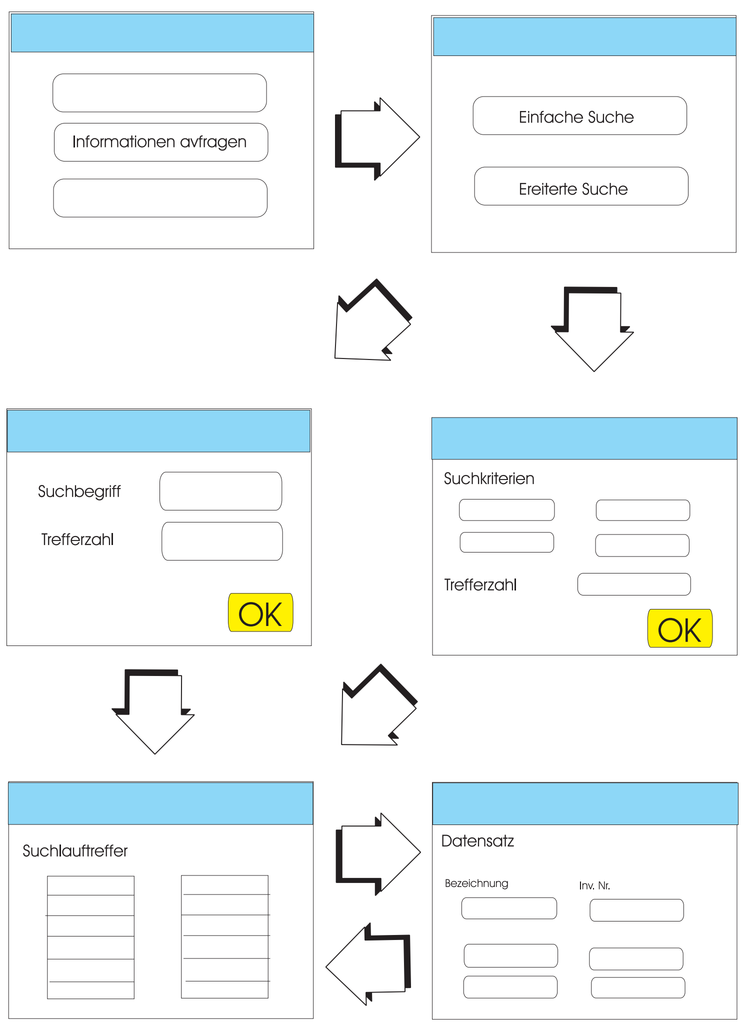

Figure 5.9: User Interface (Information abfragen)

Secondary Scenarios

- Der Benutzer kann zu jeder Zeit den Use Case abbrechen, und wieder zum Einstiegspunkt zurückkehren.
- Die Suche des Systems auf der Datenbank kann vom Benutzer jederzeit unterbrochen werden. Der Flow of Events bleibt jedoch gleich. [Abbruch bei zu langen Suchzeiten.]

Sequence Diagrams

nicht relevant für diese Aufgabe

Subordinate Use Cases

nicht relevant für diese Aufgabe

View of Participating Classes

nicht relevant für diese Aufgabe

Other Artifacts

nicht relevant für diese Aufgabe

Other Requirements

nicht relevant für diese Aufgabe

5.5 Anmelden

Augearbeitet von Josef Schmidbauer

Use Case Name

Anmelden: Der use case kann entweder vom Erfasser, vom Designer oder vom Internetbenutzer gestartet werden. Der jeweilige Benutzer kann sich durch Eingabe des Benutzernamens und des Passwortes am System anmelden.

Actors: Erfasser, Designer, Internetbenutzer

Priority:

1

Status:

Filled

Preconditions:

Keine

Postconditions:

Der Benutzer ist am System angemeldet. Ihm werden die entsprechenden Benutzergruppen zugeordnet. Es ist möglich das einem Benutzer keine Rechte zugeordnet werden.

Extension Points:

Fehlerbehandlung

Used use cases:

Keine

Flow of events:

1. Das System präsentiert dem Benutzer eine Begrüßungsseite.
2. Der Benutzer kann wählen ob er sich anmelden möchte oder ohne Anmeldung arbeiten möchte (Internetbenutzer).
3. if (Anmelden)
   (a) Der Benutzer gibt einen Benuternamen ein
   (b) Der Benutzer gibt es Passwort ein
   (c) Das System überprüft ob die Benutzername/Passwort kombination gültig ist.
   (d) if (Benutzername/Passwort) gültig
      i. Die entsprechenden Benutzergruppen und deren Rechte werden zugeordnet.
      ii. Weiter mit Punkt 5
   (e) if (Benutzername/Passwort) ungültig
      i. Das System überprüft ob der Benutzer mit dem aktuellen Benutzernamen bereits 3 mal erfolglos beim System versucht hat sich anzumelden.
      ii. if (Ja)
         A. Der Benutzer mit dem verwendeten Benutzernamen wird via E-Mail darüber informiert das 3 mal erfolglos versucht wurde das System mit seinem Benutzernamen zu verwenden.
         B. Weiter mit 3.a.iii
      iii. Das System informiert den Benutzer über den ungültigen Anmeldeversuch
      iv. Weiter mit Punkt 1
4. if (Nicht anmelden)
   (a) Die entsprechenden Benutzergruppen und deren Rechte werden zugeordnet. (In diesem Fall alle Benutzergruppen für die keine Anmeldung erforderlich ist.)

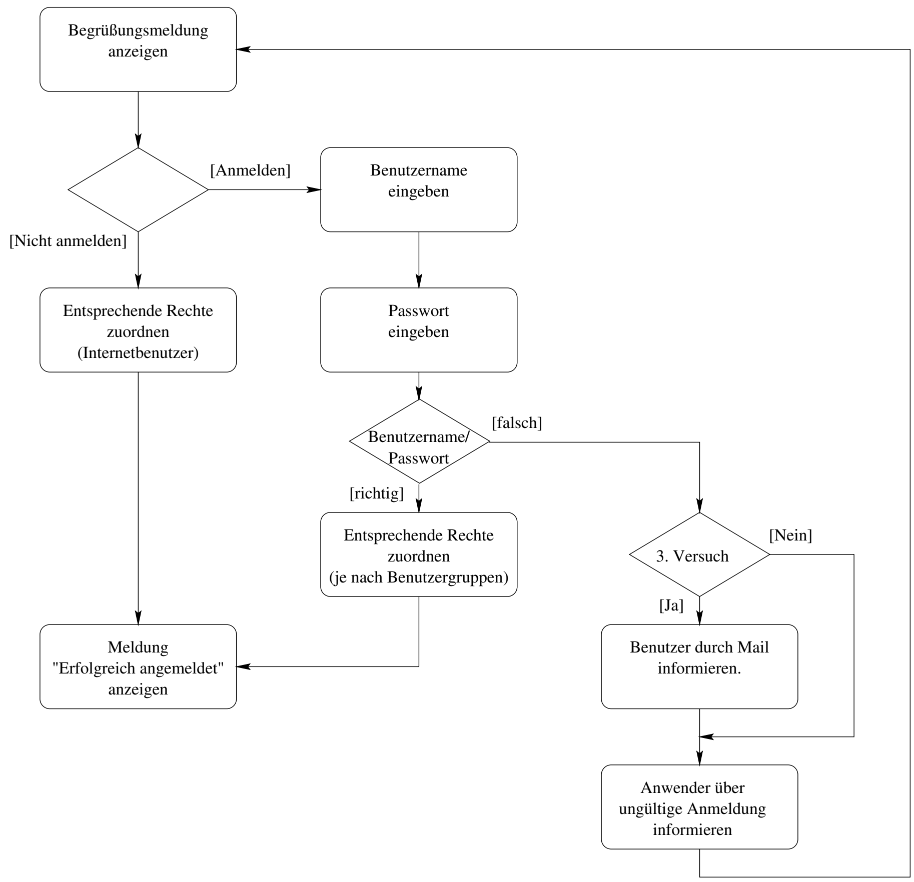

Figure 5.10: \langamericanActivity Diagram (Anmelden)

(b) Weiter mit Punkt 5

5. Der Benutzer wird auf einer Seite über die erfolgreiche Anmeldung informiert.

Activity Diagram:

Das Activity Diagramm ist in Abb.:5.10 zu sehen

User Interface:

Das User interface ist in Abb.: 5.11 zu sehen

Secondary Scenario:

Der Benutzer kann an jeder Stelle mit der “Abbruch” Taste abbrechen und gelangt wieder an den Anfang.

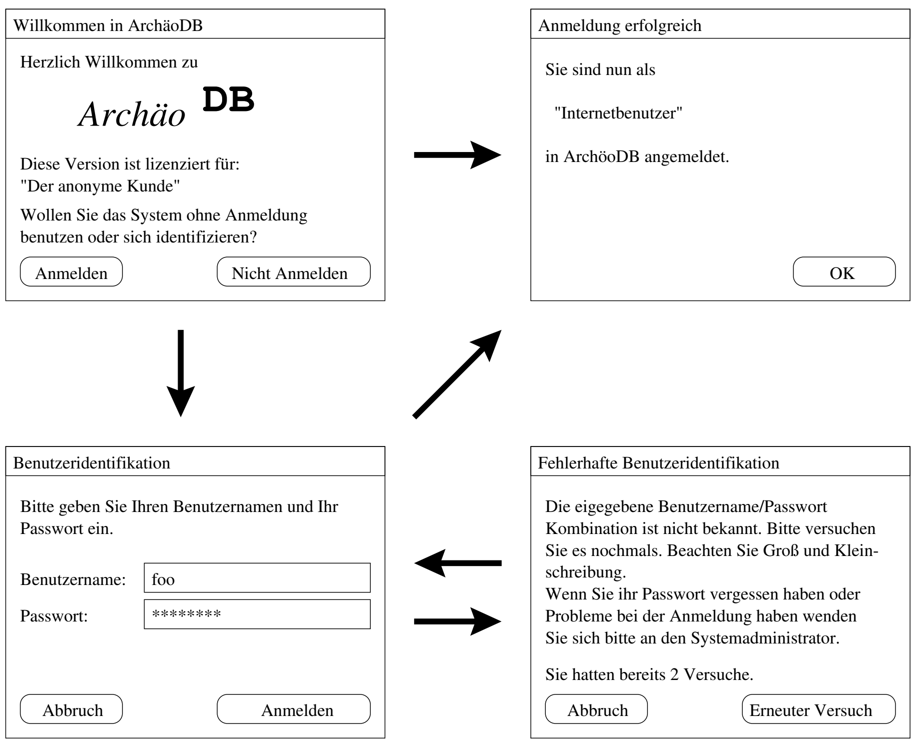

Figure 5.11: User Interface (Anmelden)

Sequence Diagrams:

Nicht relevant für diese Aufgabe

Subordinate Use Cases:

Nicht relevant für diese Aufgabe

View of Participating Classes:

Nicht relevant für diese Aufgabe

Other Artifacts:

Nicht relevant für diese Aufgabe

Other Requirements:

Nicht relevant für diese Aufgabe

Chapter 6

Systemarchitektur und Projektplan (Aufgabe 6)

6.1 Systemarchitektur

6.1.1 Definition der Module

Aufgrund der erkannten Use Cases wird eine Kategorisierung der Funktionen durchgeführt, und diese zu Modulen zusammengefaßt.

1. Objekte erfassen
   (a) Objekt_erzeugen()
   (b) Objekt_aktualisieren()
   (c) Objekt_löschen()
2. Informationen abfragen
   (a) Suchen()
   (b) Rechte_überprüfen()
   (c) Daten_abfragen()
3. Benutzerverwaltung
   (a) Anmelden()
   (b) Registrieren()
   (c) Benutzer_verwalten()
4. Virtuelles Museum erstellen
   (a) Virtuelles_Museum_erzeugen()
   (b) Virtuelles_Museum_aktualisieren()
   (c) Virtuelles_Museum_löschen()
5. Virtuelles Museum besuchen

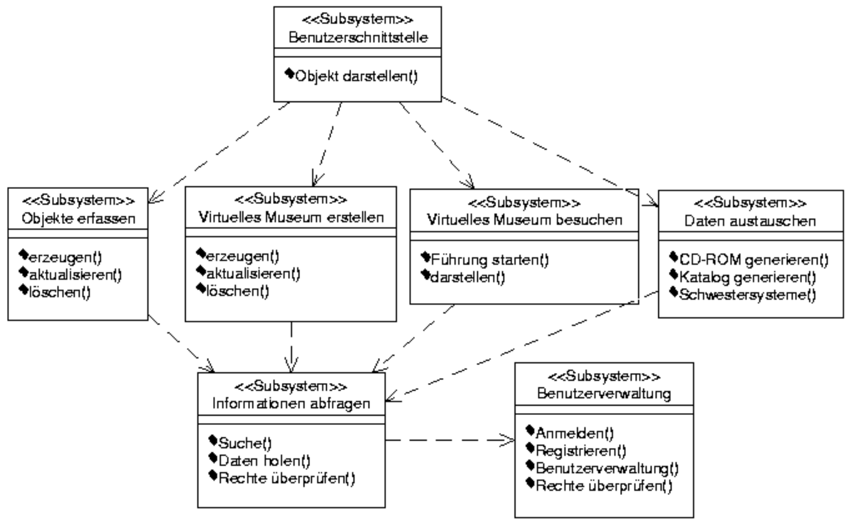

Figure 6.1: Beziehungen zwischen Modulen

   (a) Führung starten()
   (b) Virtuelles_Museum_darstellen()

6. Daten austauschen

   (a) CD-ROM_generieren()
   (b) Katalog_generieren()
   (c) Schwestersysteme()

6.1.2 Definition der Beziehung zwischen Modulen

Im nächsten Schritt werden die Beziehungen (Interfaces) zwischen den Modulen definiert.

6.1.3 Systemarchitektur

Berücksichtigen wir in unserer Betrachtung des Systems weiters die einzellnen Teile unseres Datenbestandes erhalten wir folgende Ansicht.

6.2 Projekt Plan

6.2.1 Kategorisierung der Use Cases

Reihung entsprechend der Problemstellung

Die Use Cases sind nun entsprechend ihrer Wichtigkeit bzgl. der Problemstellung zu ordnen.

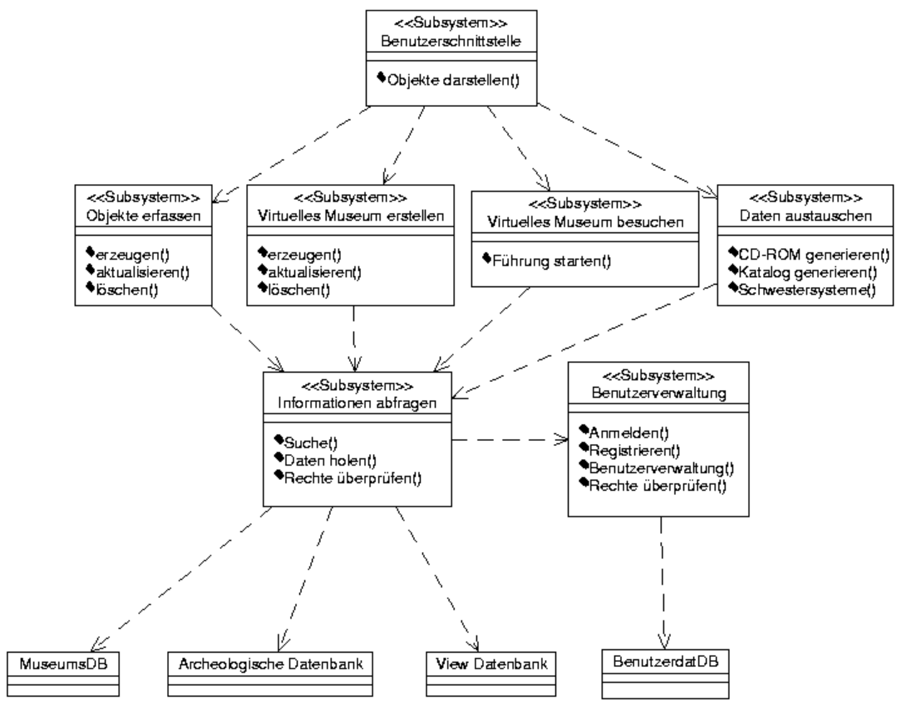

Figure 6.2: Systemarchitektur (.mdl Datei ist einzufügen (gesammt.eps))

1. Objekte erfassen
2. Informationen abfragen
3. Benutzer verwalten
4. Anmelden
5. Virtuelles Museum erstellen
6. Virtuelles Museum besuchen
7. CD-ROM generieren
8. Katalog generieren
9. Datenaustausch mit Schwestersystemen
10. System warten

Zur Entwicklung der für die Wartung des Systems erforderlichen Hilfsmittel ist zu vermerken, daß speziell der diese enthaltende Use Case, soweit als möglich, bereits paralell zum gesammten Entwicklungsverlauf realisiert werden sollte.

Reihung entsprechend dem Entwicklungsrisiko

Die Use Cases sind nun entsprechend dem Entwicklungsrisiko zu ordnen.

1. Virtuelles Museum erstellen
2. Virtuelles Museum besuchen
3. System warten
4. Objekte erfassen
5. Informationen abfragen
6. Benutzer verwalten
7. Katalog generieren
8. CD-ROM generieren
9. Datenaustausch
10. Anmelden
11. Registrieren

6.2.2 Entwicklungszeit für Use Cases

Bei der Schätzung der Entwicklungszeit der Use Cases orientieren wir uns an der bei der Schätzung der Projektdauer verwendeten Berechnungsmethode.

$UCP = UUCP * TCF * EF = 118 * 0.99 * 0.74$

$UUCP = Weight(Actors) + Weight(UseCases)$

Vernachlässigen wir die Gewichtung gemäß den Benutzern erhalten wir eine Schätzung der Entwicklungsdauer der Use Cases entsprechend ihrer Komplxität.

Gemäß den Ergebnissen der Projektdauerabschätzung gehen wir von einem Team von 5 Programmierern aus.

Use Case	Mannwochen	Entwicklungszeit
Anmelden	2,15	0,43
Registrieren	2,15	0,43
Objekte erfassen	6,45	1,29
Informationen abfragen	4,3	0,86
Virtuelles Museum erstellen	6,45	1,29
CD-ROM/Kataloge erstellen	4,3	0,86
Virtuelles Museum besuchen	4,3	0,86
Datenaustausch	2,15	0,43
Benutzer verwalten	4,3	0,86
System warten	6,45	1,29
Summe	43	8,6

Table 6.1: Entwicklungsdauer pro Mann / pro Team in Wochen

6.2.3 Definition der Iterationslänge

Aufgrund der uns vorliegenden Daten wurde vom Entwicklungsteam eine Iterationslänge von 2,5 Wochen als sinnvoll erachtet. Diese Zeitdauer erscheint geeignet einerseits einen Teilabschnitt des Projektes zu realisieren, andererseits eine straffe Strukturierung des Projektablaufes zu ermöglichen.

6.2.4 Projekt Plan

Aus der Zuordnung der Use Cases zu den einzellnen Iterationszyklen ergibt sich der Projektplan. Bei der Zuordnung wurde sowohl die Reihung der Use Cases gemäß der Wichtigkeit bzgl. der Problemstellung, als auch des Entwicklungsrisikos berücksichtigt.

Angefügt findet sich die Summe der ermittelten Entwicklungszeit der angeführten Use Cases.

1. Iterationszyklus (2,15)
   (a) Objekte erfassen
   (b) Informationen abfragen
2. Iterationszyklus(2,15)
   (a) Benutzer verwalten
   (b) Anmelden
   (c) Registrieren
   (d) Datenaustausch mit Schwestersystemen
3. Iterationszyklus(2,15)
   (a) Virtuelles Museum erstellen
   (b) Virtuelles Museum besuchen
4. Iterationszyklus(2,15)

(a) Katalog/CD-ROM generieren

(b) System warten

Chapter 7

Prototyp (Aufgabe 7)

Hier sollen nun notwendige Prototypen für Problembereiche ermittelt und beschrieben werden.

7.1 Beschreibung der Mittel zum Eingrenzen des Problembereichs

Für die Beurteilung wie sich das Problemfeld abgrenzen soll, werden vor allem die Abschnitte 2.9, 4, 5herangezogen. Aus diesen Kapiteln werden nun wiederum konkret folgende Bereiche verwendet und im folgenden näher beschrieben:

- Komplexität der Teilaufgabe
- Priorität der Teilaufgabe im Bezug auf das Gesamtprojekt
- Funktionale Abhängigkeiten zwischen Teilaufgaben.
- Performance-kritische und verwandte Bereiche
- Vorhandene Erfahrungen der Mitarbeiter der SE AG.

Komplexität der Teilaufgabe Darunter verstehen wir den Funktions- und Arbeitsumfang funktionell abgeschlossener Teilaufgaben des Gesamtprojekts. Die Teilaufgaben soll somit weitgehend auch funktionell atomar sein, wobei jedoch eine gewisse Anfangskomplexität vorausgesetzt wird, um zu vermeiden, daß z. B. auch schon ein Eingabefeld als funktionell atomar angesehen wird. Wir setzten voraus, daß diese Komplexitätsgrenze durch die Formulierung der Use-Cases vorgeschlagenen Unterteilung gegeben ist.

Damit diese Teilaufgaben als Indikatoren für einen Problembereich müssen sie jedoch noch einer weiteren notwendigen, jedoch nicht hinreichenden Bedingung genügen: Aussagen über Machbarkeit und tatsächlich zu erwartenden Umfang können erst mithilfe eines Prototypen ermittelt bzw. verbessert werden. Mit obigen Bedingungen ergibt dies nun eine hinreichende Bedingung zur Abgrenzung.

Es sollen damit zwei kritische Ereignisse vermieden werden:

- Zum einen soll vermieden werden, daß sich ein Teilbereich (aus welchen Gründen auch immer) nicht realisieren läßt, jedoch bereits viel Zeit und Geld dafür investiert worden ist.
- Zum anderen sollen Fehlkalkulationen bezüglich Zeit- und Geldinvestitionen vermieden werden, die sich daraus ergeben, daß sich bei der Programmtechnischen Realisierung das in den Use-Cases formulierten Aufgabengebiet als zu klein erweist.

In der Komplexität der Teilaufgaben kann also unter oben beschriebenen Bedingungen ein Indikator für ein Problemfeld gesehen werden.

Priorität der Teilaufgabe im Bezug auf das Gesamtprojekt Unter der Priorität einer Teilaufgabe verstehen wird konsequenterweise die Reihenfolge, in der die einzelnen Teilaufgaben realisiert werden können. Diese Priorität ergibt sich unter Berücksichtigung folgender Punkte:

- Funktionale Abhängigkeiten einzelner Teilaufgaben erfordern (falls diese Abhängigkeit nicht bidirektional ist) eine Reihung einzelner Arbeitsschritte in Bezug auf das Erstellen dieser Teilaufgaben. Mehr dazu siehe jedoch im nächsten Paragraphen.
- Teile der Software können in manchen Fällen schon vor der Fertigstellung des gesamten Projektes eingesetzt werden, bzw. dem Kunden als Demo gezeigt und somit als Verifizierung bzw. Falsifizierung der aktuellen Produktionsrichtung verwendet werden. Daher sind solche Bereiche anderen vorzuziehen.

Wie kann das ganze nun mit Problemfeldern in Beziehung gesetzt werden? Unerwartet auftretende Probleme sind vor allem in Teilaufgaben mit hoher Priorität besonders schmerzlich, da z. B. im letzteren der zuvor beschriebenen Punkte zur Begründung der Prioritätsvergabe der Fall auftreten könnte, daß aufgrund unlößbarer Probleme in Gebieten mit hoher Priorität das gesamte Projekt verworfen werden muß, während bei den selben Problemen in weniger wichtigen Bereichen zumindestens noch Teile des Projekts kommerziell verwertet werden könnten.

Dies ist jedoch kein Grund Problembereiche auch in Teilaufgaben niedriger Priorität zu suchen, jedoch erst nachdem die Teilaufgaben höherer Priorität nach Problemfeldern abgesucht worden sind.

Funktionelle Abhängigkeiten zwischen Teilaufgaben Darunter verstehen wird das Bestehen von Schnittstellen zwischen Teilaufgaben, deren Existenz und Verwendung für die Gesamtfunktion des Projektes unerläßlich sind. Es ist zwischen bi- und unidirektionalen Abhängigkeiten zwischen zwei Teilaufgaben zu unterscheiden. Diese Abhängigkeiten können eventuell durch "Dummy"-Funktionen vorläufig vermieden werden.

Weisen jedoch ein oder mehrere Teilbereiche a_1 bis a_n mehrere unidirektionale funktionelle Abhängigkeiten zu verschiedenen anderen Teilbereichen b_1 bis b_m auf, deren Relationsrichtung besagt, daß b_1 bis b_m direkt funktional abhängig ist von einem Element a der Menge $M_a = \{a_1, \ldots, a_n\}$ und ist m zu diesem Element sehr groß im Bezug auf andere Elemente a der Menge M, so kann dies ein Argument für die Prioritätsvergabe des betroffenen Teilbereiches sein (bei dieser Darstellung wurde außer acht gelassen, daß zwischen den Elementen der Menge M_a ebenfalls eine Relation bestehen kann; in diesem Fall wird einfach angenommen, daß jedes Element dieser Menge ein Abbild in der Menge $M_b = \{b_1, \ldots, b_n\}$ besitzt).

Damit können wir auf selbige Punkte stoßen, wie wir sie im vorigen Paragraphen erleutert haben. Somit ist auch die funktionelle Abhangigkeit ein wichtiger Punkt zur Ermittlung von Problembereichen.

Performance-kritische und verwandte Bereiche Dabei handelt es sich nun schon um konkretere Argumente zur Ermittlung von Problemfeldern, als jene die in vorigen Abschnitten vorgebracht wurden. Zur der Gruppe dieser konktreteren Argumente zählen z. B. Algorithmische Probleme, Gewährung von Datensicherheit etc. Es wurde jedoch für unser Projekt perförmance-kritische Bereiche ausgewählt, da es, wie sich erst später zeigen wird, einen eigenen Problempunkt darstellen kann.

Wir definieren mit Performance vor allem zeitkritische Probleme. Diese können folgendermaßen bedingt sein:

- Hardwarebedingt:
  Die Hardwareaustattung entscheidet über die Performance; vor allem Kostenfragen

- Systembedingt:
  Grundlegende Implementierungsstrategien bedingen die Performance

Es gilt hardware- und systembedingte performance-kritische Bereiche zu ermitteln und deren Machbarkeit durch Prototypen abzusichern.

Vorhandene Erfahrung der Mitarbeiter der SE AG Natürlich spielt auch die Erfahrung der einzelnen Mitarbeiter eine wesentliche Rolle bei der Abgrenzung von Problembereichen. Bereiche in denen eine große Anzahl von Mitarbeitern eine eher geringere Erfahrung aufweisen sind als wesentlich kritischer einzustufen, als Bereiche mit denen die Mitarbeiter vertraut sind.

Die Entwicklung von Prototypen soll zeigen, wo noch Wissenprobleme bestehen und eventuelle Nachschulungen ;-) angeordnet werden müssen.

7.2 Untersuchung der Problemfelder anhand der Projektspezifikation

Hier werden nun konkret jene in Abschnitt 7.1 erwähnten Kapitel dazu verwendet, um mit den ebenfalls in diesem Abschnitt definierten Mittel zur Abgrenzung der Problembereiche, selbige zu ermitteln.

Komplexität der Teilaufgabe Zur Beurteilung der Komplexität werden noch einmal folgende Tabelle zur Abschätzung des zu erwartenden Produktionsaufwand aus Abschnitt 4 betrachtet.

Actor	Actortype	Factor
Erfasser	Komplex	3
Designer	Komplex	3
Internetbesucher	Komplex	3
Systemadministrator	Komplex	3
Administrator	Komplex	3
Datenbanksystem	Simple	1
Schwesternsystem	Avarage	2
	Summe	18

Table 7.1:

Use Case	Use Case Type	Factor
Anmelden	Simple	5
Registrieren	Simple	5
Objekte erfassen	Komplex	15
Informationen abfragen	Avarage	10
Virtuelle Museen erstellen	Komplex	15
CDROMs und Kataloge erstellen	Avarage	10
Virtuelle Museen besuchen	Avarage	10
Datenaustausch	Simple	5
Benutzer verwalten	Avarage	10
System warten	Complex	15
	Summe	100

Table 7.2:

Use Case	Actors
Anmelden	Erfasser, Designer, Internetbenützer
Registrieren	Internetbenützer
Objekte erfassen	Erfasser
Informationen abfragen	Erfasser, Designer, Internetbenützer, Administrator,Administrator, Datenbanksystem
Virtuelle Museen erstellen	Designer
CDROMs und Kataloge erstellen	Systemadministrator
Virtuelle Museen besuchen	Internetbesucher, Datenbanksystem
Datenaustausch	Systemadministrator, Schwesternsystem
Benutzer verwalten	Administrator
System warten	Systemadministrator

Table 7.3:

Aus Tabelle 7.2 entnehmen wir nun zumindestens jene Use-Cases, deren Use-Case-Type ungleich Simple ist. Weiters setzen wir anhand Kapitel 5 die Actors aus Tabelle 7.1 mit den Use Cases in Bezug (siehe Tabelle 7.3).

Nun nehmen wir auch noch jene Use-Cases hinzu, die selbst den Use-Case-Type Avarage aufweisen und eine Beziehung zu Actrors haben, deren Actortype wiederum Komplex ist.

Damit erhalten wir folgende Use-Cases:

- Durch Schritt 1:
  - Objekte erfassen
  - Virtuelle Museen erstellen
  - System warten
- Durch Schritt 2:
  - Informationen abfragen
  - Virtuelle Museen besuchen
  - CDROMs und Kataloge erstellen
  - Benutzer verwalten

Die zu diesen Use-Cases gehörenden Problemfelder werden im folgenden weiter diskutiert.

Priorität der Teilaufgabe im Bezug auf das Gesamtprojekt Laut Abschnitt 4 weisen folgende Use-Cases die höchste Priorität auf:

- Objekte erfassen
- Virtuelle Museen erstellen
- Information abfragen
- Virtuelle Museen besuchen

Anmerkung: In diesem Kapitel wurden jedoch nicht alle Use-Cases diskutiert. Dennoch dient die hohe Priorität als Indikator für ein Problemfeld.

Funktionale Abhängigkeiten zwischen Teilaufgaben. Die funktionalen Abhängigkeiten stehen natürlich wieder mit vorigem Paragraphen in Bezug. Besonders deutlich kommen jene Abhängigkeiten aber in Kapitel 6.1 zum Vorschein.

Hier erweist sich vor allem der Punkt *Information abfragen* als ein äußerst zentraler, der viele unidirektionalen Relationen zu anderen Use-Cases aufweist (siehe dazu Abschnitt 7.1).

Performance-kritische und verwandte Bereiche Zuerst wollen wir noch kurz erleutern, warum verwandte Bereiche wie sie in Abschnitt 7.1 erwähnt worden sind von geringer Bedeutung sind:

- Algorithmische Probleme:
  Kein zu implementierendes Themengebiet benötigt bei der Realisierung komplexere oder gar zu entwickelnde Algorithmen. Daher sind hier keine Problemfelder zu orten.

- Datensicherheit:
  Es wir bei der Datenübertragung auf bestehende Protokolle zurückgegriffen, die in der Praxis vielfach bewährt sind. Weiters sind die vom Kunden gewünschten Anforderungen an die Betriebsicherheit nicht sehr groß, da auch mehrere kurze Ausfälle pro Tag erlaubt sind. Also ist auch in diesem Bereich nicht mit Problemen zu rechnen.

Nun gehen wir zu den performanc-kritischen Bereichen über. Dazu argumentieren wir in folgenden Unterbereichen:

- Hardwarebedingte Problemfelder:
  Da die Kalkulation der benötigten Hardware aufgrund Erfahrungswerte (puncto Internetanbindung) und aufgrund der geringen algorithmischen Ansprüchen (puncto Hardwareperformance) dem Kunden leicht vorgelegt werden kann, sind bei dessen Zustimmung über die Kosten keine weiteren Probleme zu erwarten.

- Systembedingte Problemfelder:
  Diese können nicht durch den Einsatz höherer Mittel kompensiert werden. Wiederum ist aufgrund der algorithmischen Einfachheit nicht mit Problemen zu rechnen.
  Ein Ausnahme hiervon kann jedoch in der Internetanbindung benutzerseitig bestehen. Aufgrund noch geringer Erfahrung des Kunden im Umgang mit dem Internet könnte ihm die Bedienung des Web-Frontends subjektiv im Vergleich zu anderen herkömmlichen Bedienmethoden als zu langsam erscheinen. Ein Prototyp, der dem Kunden den Sachverhalt illustriert, könnte daher von Vorteil sein.

Vorhandene Erfahrungen der Mitarbeiter der SE AG Prinzipiell ist die Kompetenz der Mitarbeiter des SE AG natürlich sehr groß :-). Dennoch wurde bei manchen (wenn auch nicht bei allen) ein kleineres Manko im Umgang mit Internetanbindungen festgestellt. Weitere Themengebiete sind davon nicht betroffen, da alle Mitarbeiter Erfahrung im Programmieren und auch mit Datenbankanbindungen (PS Datenbanken I :-)) aufweisen.

Aus diesen Überlegungen sind Use-Cases, die thematisch mit dem Internet in Verbindung stehen für die Prototypentwicklung näher zu betrachten.

Diskussion zur Abgrenzung des Problembereiches Es werden zuerst alle Use-Cases in Betracht gezogen, die eine hohe Komplexität aufweisen, da sich keine anderen weniger aufwendigen Use-Cases als funktionell zentral erwiesen haben. Zusammenfassend handelt es sich dabei nun um folgenden Use-Cases:

- Objekte erfassen
- Virtuelle Museen erstellen
- System warten
- Informationen abfragen
- Virtuelle Museen besuchen
- CD-ROMs und Kataloge erstellen
- Benutzer verwalten

Nach Betrachtung deren Priorität bleiben folgende über:

- Objekte erfassen
- Virtuelle Museen erstellen
- Informationen abfragen
- Virtuelle Museen besuchen

Da zu einen der Use-Case *Information abfragen* sich als thematisch als sehr zentral erweist, aber durch den auch, wie ebenfalls die Use-Cases *Objekte erfassen, Virtuelle Museen erstellen* und *Virtuelle Museen,* dem Kunden die Internetbedienung gut illustriert werden kann, wird das Problemfeld auf diesen Use-Cases eingeschränkt.

Nach dem Betrachten des Erfahrungsschatzes einzelner Mitarbeiter erweist sich diese Wahl als besonders passend, da dadurch auch der Umgang mit Internetanbindungen erprobt werden kann.

7.3 Beschreibung des gewählten Prototypen

Es wurde zur Demonstration der Machbarkeit sowie für eventuelle Vorführungen für den Kunden ein einfacher Prototyp des Use-Cases *Informationen abfragen* implementiert.

Der Prototyp basiert auf cgi Scripts welche in C implementiert wurden und unter dem Apache Web Server laufen. Dies stellt die sog. Middle-Ware (in der Literatur oft auch als Business-Logic bezeichnet) dar. Der User greift über einen Web Browser darauf zu. Die Middle-Ware selbst greift auf einen Datenbankserver zu. Dadurch ergibt sich die, in der Praxis oft eingesetzte, 3 schichtige Architektur von Anwender, Middle-Ware und Datenbank im Hintergrund. Da zum Testen keine Datenbank zur Verfügung stand liefert die, von der Middle-Ware verwendete Datenbankschnittstelle fixe Demonstrationswerte zurück.

Der Prototyp ist unter *http://otto.ath.cx/˜jschmid/Prototyp* (nicht 24h online) sowie unter *http://www.cosy.sbg.ac.at/˜jschmid/Prototyp* (eventuell Probleme mit dem cgi Scripten auf dem Cosy Solaris Web Server) erreichbar. Ein Screenshot ist in Abbildung 7.1 dargestellt.

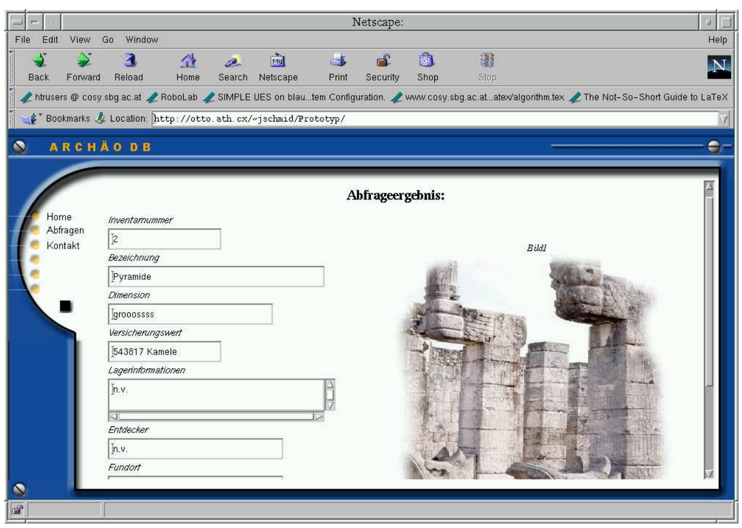

Figure 7.1: Screenshot des Prototypen

Chapter 8

Konstruktionsphase (Aufgabe 8)

8.1 Konzeptionelles Klassendiagramm

In Abbildung 8.1 ist das konzeptionelle Klassendiagramm der Iteration 1 (*Objekte erfassen*) ersichtlich. Diese Iteration beinhaltet *Objekte erzeugen*, *Objekte aktualisieren* und *Objekte löschen*.

8.2 Sequenz-Diagramm

In Abbildung 8.2 ist das Sequenz-Diagramm für die Iteration 1 ersichtlich.

8.3 Kollaborations-Diagramm

In Abbildung 8.3 ist das Kollaborations-Diagramm für die Iteration 1 ersichtlich.

8.4 Zustandsdiagramm

In Abbildung 8.4 ist das Zustands-Diagramm für das *Anmelden* ersichtlich.

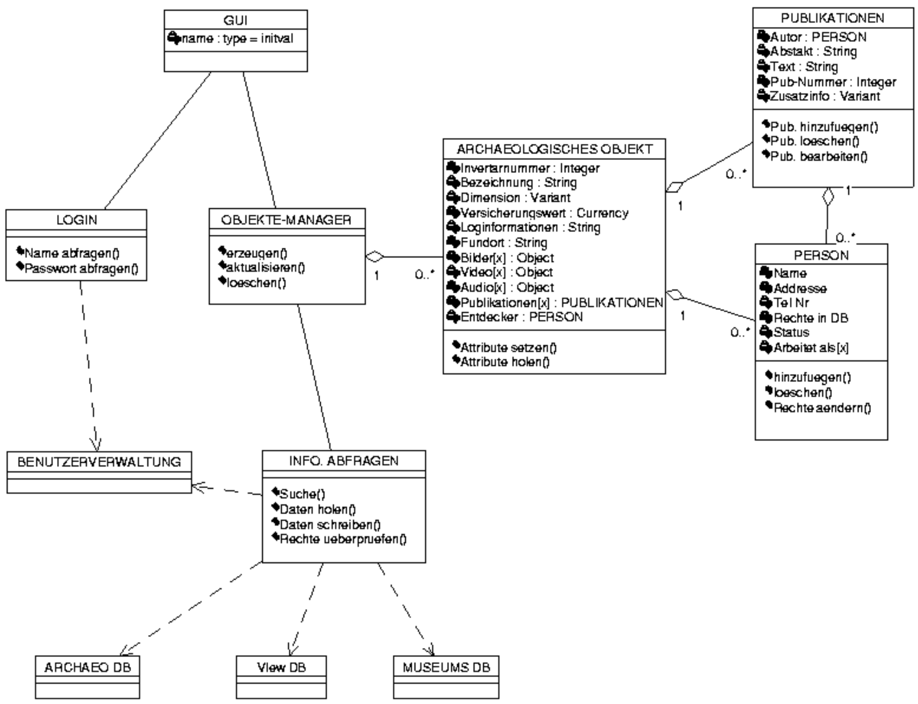

Figure 8.1: Klassendiagramm

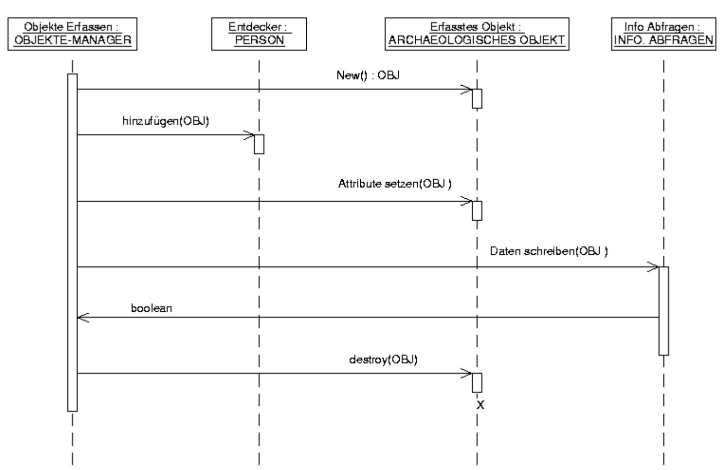

Figure 8.2: Sequenz-Diagramm

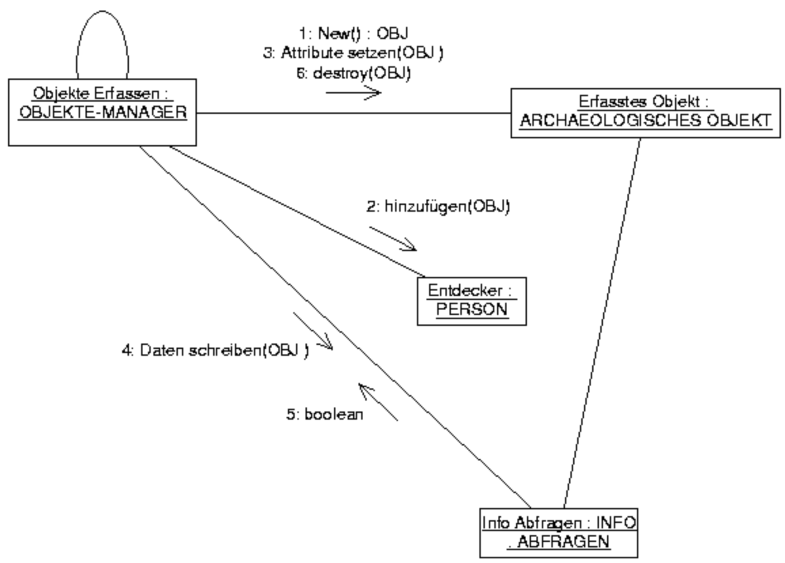

Figure 8.3: Kollaborations-Diagramm

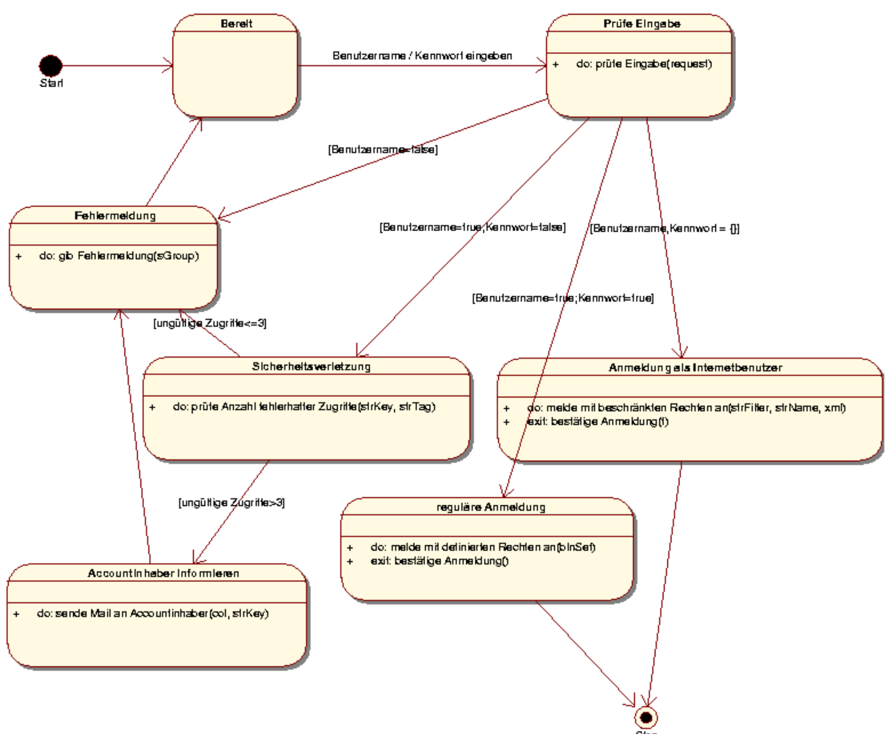

Figure 8.4: Zustandsdiagramm

Appendix A

Glossar

Abnahmekriterien: Wenn diese Kriterien erfüllt sind, muß das in Auftrag gegebene Programm vom Auftraggeber abgenommen werden.

Administrator: Der „Verwalter" der Datenbank / des Programms (Systemverantwortlicher).

Alpha-Test: Erster Testdurchlauf in der Zielumgebung, um die Funktionsfähigkeit des gesamten Software-Systems zu überprüfen. Dieser Test findet in der Regel direkt nach Abschluß der Implementierung statt.

Anmeldung: Durch die Anmeldung wird einem Anwender ein Benutzer zugeordnet. Wenn keine Anmeldung erfolgt so erhält der Anwender alle Benutzergruppen zugewiesen für die keine Anmeldung erforderlich ist. Bei einer Anmeldung muß sich ein Anwender mit seinem Benutzernamen und dem zugeordneten Passwort authentifizieren.

Anwender: Eine physikalische Person welche das System verwendet. Wenn sich ein Anwender *angemeldet* hat wird er einem *Benutzer* zugeordnet. Die Benutzerdaten werden zentral vom System verwaltet.

archäologisches-ObjektItemizeSiedlungen

zu jedem dieser Objekte wird eine bestimmte Anzahl an Attributen abgelegt.

Attribut: Ein Attribut ist eine Eigenschaft eines Objektes welche zusammen mit diesem in der Datenbank abgelegt wird.

Benutzer: Ein Anwender welches das System verwendet. Er kann sich über das Web Frontend beim System anmelden und bekommt damit die, ihm zugeordneten *Benutzergruppen* zugewiesen. Je nach Zugehörigkeit zu einzelnen Benutzergruppen erhält der Benutzer verschiedene Rechte im System. Weiters ist ein Benutzer für alle von ihm eingegebenen Daten verantwortlich.

Benutzergruppe: Einer Benutzergruppe sind bestimmte Rechte zugeordnet. Dies Umfasst unter anderen den Zugriff auf Objekte und Attribute sowie auch die Benutzung des Web Frontends, einen Virtuellen Musseums und aller anderen Systemteile. Jedem *Benutzer* werden eine oder mehrere Benutzergruppen zugeordnet.

Datenbank: Ein System zur Speicherung und Wiedergewinnung großer Datenmengen.

Datensatz: Eine Ansammlung von Daten, die in einer elementaren Beziehung zueinander stehen oder gemeinsame Merkmale haben (hier: z.B. ein Artefakt.)

funktionale-Anforderungen: Features die das Programm bieten muß, die direkt die Funktionalität des Programms betreffen.

HTTPS: (Secure Hyper-Text Transfer Protocol) Das Secure HyperText Transfer Protocol stellt eine Erweiterung zu den von Hallam-Baker vorgeschlagenen Möglichkeiten der Sicherung von Informationen im World Wide Web dar.
Zur Unterscheidung von gewöhnlichen, nicht gesicherten Übertragungen tragen URLs, welche nur für die Zusammenarbeit mit S-HTTP geeignet sind, einen anderen Typ, nämlich "shttp" anstatt von "http". Ein Zugriff auf das hinter dieser URL stehende Dokument soll nur verschlüsselt erfolgen. Ein Client, welcher dieses Protokoll nicht "spricht", kann nicht auf die Daten zugreifen. Es werden keine sicherheitsrelevanten Daten unverschlüsselt übertragen.

nichtfunktionale-Anforderungen: Features die das Programm bieten soll, die nicht direkt die Funktionalität des Programms betreffen (z.B. Sicherheitsbelange).

Online-Hilfe: Ein Teil des Programms, das Funktionen des Programms während des Programmlaufs erklärt.

UML: (Unified Modeling Language) Die UML ist eine graphische Sprache, die sich im gesamten Software-Entwicklungsprozeß, von der konzeptionellen Analyse bis zur Beschreibung der Implementierung einsetzen läßt. Die UML enthält allerdings kein Vorgehensmodell, das die methodische Entwicklung der Software angibt.

XML: (Extensible Markup Language) ist eine textbasierte Metasprache, die es ermöglicht, Daten bzw. Dokumente so zu beschreiben, dass sie - insbesondere über das Internet - zwischen einer Vielzahl von Anwendungen ausgetauscht und weiterverarbeitet werden können. Im Gegensatz zu HTML ist es mit XML möglich, auch die Struktur von Daten zu beschreiben. Die Bezeichner für die Tags (Abschnittsmarkierungen, wie z. B. "ARTIKEL") können frei gewählt werden.

Appendix A

Programme

A.1 Listing eines erstellten Prototypen

Im folgenden findet sich das Listing des query.cgi Scripts welches vom ausgewählten Prototyp verwendet wird. Es dient nur zur Demonstration der Machbarkeit des ausgewählten Prototypen und muß natürlich für das richtige System neu implementiert werden.

```
   #include <stdio.h>
   #define NOT_FOUND -1

   define SYSTEM_VERSIONSSTRING "ArchäoDB Middle-Ware - query.cgi Script
0.1"
   define SYSTEM_DATABASE_IMAGE_PATH "images/"
   define SYSTEM_IMAGE_VOID "spacer_black.gif"

   define FELD_INVENTARNUMMER "Inventarnummer"
   define FELD_BEZEICHNUNG "Bezeichnung"
   define FELD_DIMENSION "Dimension"
   define FELD_VERSICHERUNGSWERT "Versicherungswert"
   define FELD_LAGERINFORMATIONEN "Lagerinformationen"
   define FELD_ENTDECKER "Entdecker"
   define FELD_FUNDORT "Fundort"
   define FELD_ALTER "Alter"
   define FELD_IMAGE "image"

   define FELD_SYSTEM_VERSION "version"

   define TEMPLATE_FILE_ARTEFACT "abfrage_template1.html"

   int display_artefact_by_inv_nr(long inv_nr);
   char database_get_field_by_key(char *feldname, long key, char *result);


   int main(int argc, char *argv[])
   {
```

```
  int c;
  int i;
  char buf[256];
  char *q;
  char *p;

  FILE *f;

  long inv_nr;

  // Teste auf Parameter

q=getenv("QUERY_STRING");

  if(*q)
  {
        // Aufruf mit Parametern -> Teste
        if(p=strstr(q,FELD_IMAGE))
        {
          // Bild ist zurückzuliefern
          int bildnr;
          while(*p!='=' && *p) p++;
          bildnr=atoi(p+1);
          p=strstr(q,FELD_INVENTARNUMMER);
          while(*p!='=' && *p) p++;
          inv_nr=atoi(p+1);
 sprintf(buf,"%sartefakt-%d-%d.jpg",SYSTEM_DATABASE_IMAGE_PATH,inv_nr,bildnr);
         f=fopen(buf,"rb");
          if(f)
          {
           printf("Content-type:  image/jpeg\n\n");
            while((c=fgetc(f))!=EOF) putc((char)c,stdout);
            fclose(f);
          }
          else
          {
           sprintf(buf,"%s%s",SYSTEM_DATABASE_IMAGE_PATH,SYSTEM_IMAGE_VOID);
           f=fopen(buf,"rb");
           printf("Content-type:  image/gif\n\n");
            while((c=fgetc(f))!=EOF) putc((char)c,stdout);
            fclose(f);
          }
        }
        else
          // Andere Arten derzeit nicht implementiert
        printf("Content-type:  text/html\n\nERROR - Invalid parameters");

  }
  else
  {
```

```
    // Datenbankabfrage starten

   printf("Content-type:  text/html\n\n");

    inv_nr=NOT_FOUND;
    while(fgets(buf,255,stdin))
    {
      if(q=strstr(buf,FELD_INVENTARNUMMER))
      {
          while(*q!='=' && *q) q++;
          if(*q) inv_nr=atol(q+1);
      }

    }
    if(inv_nr!=NOT_FOUND)
      display_artefact_by_inv_nr(inv_nr);
    else
   printf("ERROR - Invalid usage of script");
  }
  // Ende des Scripts
}


int display_artefact_by_inv_nr(long inv_nr)
{
  FILE *f;
  char *p;
  char buf[513];
  char buf2[513];
  char term[513];

  // Parsen des Template Files fuer die Antwort und Ausgabe
f=fopen(TEMPLATE_FILE_ARTEFACT,"ra");

  while(fgets(buf,255,f))
  {
    strcpy(buf2,buf);
    // Inventarnummer in Template einfügen
   if((p=strstr(buf, "$$" FELD_INVENTARNUMMER)))
    {
     sprintf(term,"%d",inv_nr);
      strcpy(p,term);
     strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_INVENTARNUMMER));
    }

    strcpy(buf2,buf);
    // Bezeichnung in Template einfügen
   if((p=strstr(buf, "$$" FELD_BEZEICHNUNG)))
    {
      database_get_field_by_key(FELD_BEZEICHNUNG, inv_nr, term);
```

```
          strcpy(p,term);
         strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_BEZEICHNUNG));
        }
        strcpy(buf2,buf);
        // Dimension in Template einfügen
       if((p=strstr(buf, "$$" FELD_DIMENSION)))
        {
          database_get_field_by_key(FELD_DIMENSION, inv_nr, term);
          strcpy(p,term);
         strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_DIMENSION));
        }
        strcpy(buf2,buf);
        // Versicherungswert in Template einfügen
       if((p=strstr(buf, "$$" FELD_VERSICHERUNGSWERT)))
        {
          database_get_field_by_key(FELD_VERSICHERUNGSWERT, inv_nr,
term);
          strcpy(p,term);
         strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_VERSICHERUNGSWERT));
        }
        strcpy(buf2,buf);
        // Lagerinformationen in Template einfügen
       if((p=strstr(buf, "$$" FELD_LAGERINFORMATIONEN)))
        {
          database_get_field_by_key(FELD_LAGERINFORMATIONEN, inv_nr,
term);
          strcpy(p,term);
         strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_LAGERINFORMATIONEN));
        }
        strcpy(buf2,buf);
        // Entdecker in Template einfügen
       if((p=strstr(buf, "$$" FELD_ENTDECKER)))
        {
          database_get_field_by_key(FELD_ENTDECKER, inv_nr, term);
          strcpy(p,term);
         strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_ENTDECKER));
        }
        strcpy(buf2,buf);
        // Fundort in Template einfügen
       if((p=strstr(buf, "$$" FELD_FUNDORT)))
        {
          database_get_field_by_key(FELD_FUNDORT, inv_nr, term);
          strcpy(p,term);
         strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_FUNDORT));
        }
        strcpy(buf2,buf);
        // Alter in Template einfügen
       if((p=strstr(buf, "$$" FELD_ALTER)))
        {
          database_get_field_by_key(FELD_ALTER, inv_nr, term);
```

```
        strcpy(p,term);
       strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_ALTER));
      }
      strcpy(buf2,buf);
      // Image in Template einfügen
     if((p=strstr(buf, "$$" FELD_IMAGE)))
      {
     sprintf(term,"query.cgi?" FELD_INVENTARNUMMER "=%ld&" FELD_IMAGE
"=1",inv_nr);
        strcpy(p,term);
       strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_IMAGE));
      }

      // SYSTEM Daten einfügen

      strcpy(buf2,buf);
      // Systemversion
     if((p=strstr(buf, "$$" FELD_SYSTEM_VERSION)))
      {
       sprintf(term,"%s",SYSTEM_VERSIONSSTRING);
        strcpy(p,term);
       strcpy(p+strlen(term),buf2+(p-buf)+strlen( "$$" FELD_SYSTEM_VERSION));
      }

      fputs(buf,stdout);
    }
    fclose(f);
    return -1;
   }



   // 1ster Versuch!  eine einfachen, abstrakten Datenbankschnittstelle
   // muss natuerlich noch sehr stark veraengert werden und soll mal
was in der form von
   // select feldname from DATENBANKNAMEN where key=unserkey
   // oder so an eine SQL Datenbank senden....

   // Da wir z.Z. keine reale Datenbank haben (aus Kostengruenden)
verwenden wir hier eine Dummy Funktion
   char database_get_field_by_key(char *feldname, long key, char *result)
   {
          if(strcmp(feldname, FELD_BEZEICHNUNG)==0) strcpy(result,"n.v.");
          if(strcmp(feldname, FELD_BEZEICHNUNG)==0 && key==1) strcpy(result,"Großer
schwarzer Sarg");
          if(strcmp(feldname, FELD_BEZEICHNUNG)==0 && key==2) strcpy(result,"Pyramide");
          if(strcmp(feldname, FELD_BEZEICHNUNG)==0 && key==3) strcpy(result,"Fluch
mittlerer Größe");

          if(strcmp(feldname, FELD_DIMENSION)==0) strcpy(result,"n.v.");
```

```
          if(strcmp(feldname, FELD_DIMENSION)==0 && key==1) strcpy(result,"1x2x1
m");
          if(strcmp(feldname, FELD_DIMENSION)==0 && key==2) strcpy(result,"grooossss");


          if(strcmp(feldname, FELD_VERSICHERUNGSWERT)==0) strcpy(result,"n.v.");
          if(strcmp(feldname, FELD_VERSICHERUNGSWERT)==0 && key==2)
strcpy(result,"543817 Kamele");
          if(strcmp(feldname, FELD_VERSICHERUNGSWERT)==0 && key==3)
strcpy(result,"Nicht versicherbar");

          if(strcmp(feldname, FELD_LAGERINFORMATIONEN)==0) strcpy(result,"n.v.");
          if(strcmp(feldname, FELD_LAGERINFORMATIONEN)==0 && key==1)
strcpy(result,"Kühl und trocken lagern - mindestens haltbar bis hier
Deckel");

          if(strcmp(feldname, FELD_ENTDECKER)==0) strcpy(result,"n.v.");
          if(strcmp(feldname, FELD_ENTDECKER)==0 && key==1) strcpy(result,"Huber
Sepp");
          if(strcmp(feldname, FELD_ENTDECKER)==0 && key==3) strcpy(result,"hat
den Fluch nicht überlebt");

          if(strcmp(feldname, FELD_FUNDORT)==0) strcpy(result,"n.v.");
          if(strcmp(feldname, FELD_FUNDORT)==0 && key==1) strcpy(result,"Ägypten");
          if(strcmp(feldname, FELD_FUNDORT)==0 && key==2) strcpy(result,"Ägypten");
          if(strcmp(feldname, FELD_FUNDORT)==0 && key==3) strcpy(result,"Cosy");

          if(strcmp(feldname, FELD_ALTER)==0) strcpy(result,"n.v.");
          if(strcmp(feldname, FELD_ALTER)==0 && key==1) strcpy(result,"4000
Jahre");
          if(strcmp(feldname, FELD_ALTER)==0 && key==3) strcpy(result,"wenige
Stunden");
           return -1;
   }
```

Appendix B

Fragen

B.1 Aufgabe 1

B.1.1 Allgemein

- Soll das System nur für ein Museum entwickelt werden, oder ist in weiterer Folge an die Einbindung anderer kultureller Institutionen oder wissenschaftlicher Forchungseinrichtungen gedacht?

  Das System ist nicht auf die Verwaltung der Daten eines Museums beschränkt. Die Datenhaltung erfolgt jedoch für alle Datenbestände zentral, wobei selbige jedoch gtrennt verwaltet werden müssen.

- Dient das System primär der Verwaltung der Daten (Lagerbestand an Artefakten != wissenschaftl. Netzwerk), oder steht die Darstellung für potentielle Museumsbesucher im Vordergrund?

  Sowohl die Verwaltung der Daten wie auch die Darstellung für Internetbesucher sind zu realisieren. Auch die Lagerverwaltung (z.B. Objekt real vorhanden) soll berücksichtigt werden.

B.1.2 Datenbestand

- Repräsentiert die bisher getroffene Auswahl (Artefakte, Gräber, Siedlungen) bereits die Objekte die das System verwalten können soll? (Aus dieser Überlegung leitet sich unmittelbar die Frage nach den möglichen Beziehungen ab.)

  Das System soll sich auf die Verwaltung dieser archäologischen Fakten beschränken. Zu anderen Objekten (wie etwa Entdecker) sollen jedoch sehr wohl weitere Daten erfaßt werden können (Lebenslauf,etc.)

- Welche groben Randbedingungen gibt es (Anzahl der Datensätze, Größe eines Datensatzes) ?

  Jährlich vergrößert sich der Datenbestand um:

1. 400 Bilder (diese Zahl entspricht auch den neu hinzugefügten Artefakten)

   (a) 400 Handzeichnungen

(b) 10 Filme a 10 sec

(c) Texte (genaue Anzahl schwer zu erfassen)

- Welche Daten sollen über Fundgegenstände gespeichert werden können? Sollen bestimmte Kerninformationen (Fundort, Alter,Lager) auf jeden Fall vorhanden sein?

  Die Attribute die für Artefakte vergeben werden können wurde nur ungefähr bestimmt. Inventarnummer, Bezeichnung und Dimension sollten in jedem Fall angegeben werden (Pflichtfelder). Neben historischen Daten sollen auch der Versicherungswert und Lagerinformation gespeichert werden können.

- Soll für einzellne Datensätze eine individuelle Rechtevergabe möglich sein?

  Individuelle Rechtevergabe soll bis auf Attributebene möglich sein. Allerdings wäre eine default Angabe, die gegebenenfalls noch vom Erfasser geändert werden kann, wünschenswert.

- Gibt es Standards die den Datenaustausch mit bereits entwickelten Systemen erleichtern könnten?

  Es sind dem Kunden keine Standarts bekannt. Informationen werden eingeholt.

- Müssen bestimmte Dateien (Graphik,Musik,...) in einem bestimmten Format gespeichert werden?

  Gewisse allgemeine Formate sollten verwendet werden, um beliebigen Internetbenutzern den Zugriff zu ermöglichen. Prinzipiell müssen die zur Dastellung nötigen Tools aber benutzerseitig geladen werden.

B.1.3 Benutzerklassen

- Welche Benutzergruppen soll es geben? (Designer virtuelles Museum)

  Es sollen Benutzergruppen erstellt werden können, an die bestimmte Rechte vergeben werden können. Die bisherige Unterteilung wird -da zu einschränkend- verworfen.

- Sollen sämtliche Erfasser uneingeschränken Zugriff auf die vorhandenen Daten haben? Wer erhält Zugang (Schreib und Lesezugriff) zu welchen Daten?

  siehe oben

- Wird ein Benutzer informiert, wenn von ihm eingegebene Daten, oder der von ihm zusammengestellten virtuellen Vitrinen verändert werden?

  Dieser Aspekt wurde vom Kunden begrüßt. Die Benachrichtigung könnte über Mail erfolgen.

B.1.4 Web Frontend

- Sollen Erfasser und Internetbesucher eine gemeinsame Schnittstelle benutzen?

  Das Web Frontend sollte, je nach Rechten des einloggenden Benutzers, unterschiedliche Funktionalität aufweisen. Ein gemeinsamer Nenner ist nicht vorgeschrieben.

- Gibt es spezielle Vorstellungen, das Web Frontend betreffend (Graphik, Formulare,..) ?

  Nein.

- Nach welchen Kriterien soll gesucht werden können (Metaklassen)?

  Von unserem Kunden werden 2 Suchmodi verlangt:

einfach Der Benutzer gibt ein Stichwort ein nach dem die Datenbank durchsucht wird.

erweitert Die Suche kann durch verschiedene Einschränkungen, die aus einer vorgegebenen Auswahl getroffen werden müssen, eingeengt werden.

Auch die Suche nach abstrakten Begriffe (Epochen, Ländern,etc) soll möglich sein.

- Welche Sicherheitsstandards (Übertragung, Login) sollen realisiert werden?

  Die Internetverbindung erfolgt über HTTPS. Sämtliche weitere Sicherheitsvorkehrung werden auf HW-Ebene realisiert, und dürfen als gegeben angenommen werden.

B.1.5 Virtuelles Museum

- Gibt es mehrere virtuelle Museen?

  Die Erstellung mehrerer Museum wäre wünschenswert, ist jedoch nicht gefordert.

- Darf jeder Erfasser ein eigenes virtuelles Museum erstellen?

  siehe Rechtevergabe; Es wäre außerdem sinnvoll, den Zugriff des Großteils der Erfasser auf die von ihnen erstellten Vitrinen, bez. Räume zu beschränken.

- In welcher graphischen Form soll das virtuelle Museum dargestellt werden ?

  Die Darstellung soll zweidimensional erfolgen, wobei den realen Gegebenheiten eines Museums (Stockwerke,Räume,etc.) , so weit als möglich Rechnung getragen werden sollte. Insbesonders soll die räumliche Anordnung thematisch und zeitlich geordnet erfolgen.

  In einem Stockwerk werden, zum Beispiel, nur Fibeln ausgestellt. Die Räume enthalten enthalten die Fibeln bestimmter Epochen, während der genaue zeitliche Ablauf durch die Anordnung der Vitrinen wiedergespiegelt wird.

- Welcher Art sollen die Hilfsmittel sein, die zur Generierung zur Verfügung stehen (Schemata, freies Design) ?

  Da das virtuelle Museum an ein reales Museum erinnern soll, sollten entsprechende Hilfsmittel zur Verfügung stehen. Da wir benutzerseitig jedoch keine Programme installieren können, ist aus Gründen der Performance von besonders aufwendigen Gestaltungsmöglichkeiten abzusehen.

- Sollen Führungen durch das virtuelle Museum möglich sein?

  Diese Idee wurde als geeignetes Angebot, speziell für Internetbenutzer die sich nur kurz aufhalten wollen, erachtet. Die Erstellung soll weitgehend automatisiert werden.

- Haben wissenschaftliche und reguläre Erfasser unterschiedliche Rechte, oder andere Hilfsmittel zur Verfügung?

  siehe Rechtevergabe;

- Sollen links auch von unserem virtuellen Museum wegführen können?

  Nein.

- Wer ist für die Information zu den einzellnen virtuellen Räumen zuständig?

  siehe Rechtevergabe;

B.1.6 Sonstiges

- In welcher Form sollen die Kataloge generiert werden (automatisch, manuell) ?

  Ebenso wie die CD-Rom soll der Katalog automatisch aus dem virtuellen Museum generiert werden können.

- Ist an Anschlußprojekte gedacht, die bereits berücksichtigt werden könnten (Kartenbestellung) ?

  Das einzige Anschlußprojekt, an das momentan gedacht wird, wäre den bereits auf einer Homepage präsenten Museumsshop auf E-Commerce umzustellen.

B.2 Aufgabe 2 (KW 45)

B.2.1 Allgemein

- Sollen Bilder/Videos/Audiodaten in Hinblick auf Qualität und Bandbreite im Webfrontend/im virtuellen Museum/auf CD-ROM,.. in verschiendenen Qualitäten/Formaten präsentiert werden können?

  Verschiedene Formate mit verschiedenen Qualitäten und Bandbreitesprüchen sollen möglich sein.

- Wie groß soll die Zuverlässigkeit des Systemes sein (99,xx%)? Wie lange "darf" ein Ausfall des Systemes dauern?

Muß nicht sehr hoch sein. Der Kunde erlaubt Ausfälle bis zu maximal 2h/Tag!

- Wielviele Benutzer werden vorraussichtlich im Durchschnitt bzw. maximal das System gleichzeitig benutzen?

  Der Kunde erwartet das maximal 100 Internetbenutzer und 5 Bearbeiter das System gleichzeitig verwenden.

- Soll das System die Anzahl der Benutzer beschränken wenn sonst die Antwortzeit zu hoch würde?

  Der Kunde begrüßt das das System die Anzahl der Benutzer beschränkt z.B. wenn es sonst zu langsam würde.

B.2.2 Datenbestand

- Sind sonstige Größen beim Datenbestand (bzw. der Erweiterung dieses) bekannt (Audiodaten,... etc.)?

  Der Kunde gibt an das gleich viele Audiodaten eingegeben werden wie Videodaten.

- Inwieweit ist der Zuwachs der Daten absehbar, vorhersagbar?

  Das Wachstum entspricht den angegebenen Werten.

- Wie viele Daten sind bereits vorhanden und müssen in das System aufgenommen werden können (Startgröße)?

  Die Startgröße der Datenbank ist null. D.h. es sind am Anfang keine Daten vorhanden.

- Sollen die Rechte (bis auf Attributebene) global für alle Datensätze vergeben werden können, für jeden Datensatz einzeln oder werden Gruppen von Datensätzen eingefügt werden für die die Rechte festgelegt werden können?

  Es soll möglich sein die Rechte für jedes einzelne Attribut von JEDEM einzelnen Objekt festzulegen. D.h. Wenn z.B. ein ganz besonderes Artefakt gefunden wird sollen speziell für dieses die Rechte angegeben werden können. Dies gilt für alle verwalteten Daten.

  Der Kunde wünscht weiter das in Zusammenarbeit mit uns Benutzergruppen ausgearbeitet werden und als Vorgabe ins System eingefügt werden (Customizing). Der Kunde hat die Möglichkeit diese Benutzergruppen je nach den entsprechenden Rechten beliebig zu ändern und anzupassen bzw. neue hinzuzufügen oder existierende zu löschen.

- Recht es die Audio/Video Daten intern in einem "Web-tauglichen" Format zu speichern oder sollen sie intern in best möglicher Qualität (hinsichtlich des ursprünglichen Quellmaterials) gespeichert werden und gegebenenfalls für die Web Darstellung in ein Format mit weniger Bandbreitenbedarf umgewandelt werden (Darstellung in verschiedener Qualität für verschiedene Benutzer würde möglich, Vielleicht auch mit Hinblick auf Copyright Fragen, Kaufvideos).

Graphiken, Photos sollen dynamisch bei der Verwendung in ein geeignetes Format umgerechnet werden. Videos und Audiodaten werden in der höchst möglichen Qualität, sowie in zwei verschiedenen Qualitätsstufen (Hohe Qualität, Niedrige Qualität) abgespeichert. Die beiden Qualitätsstufen dienen vor allen für die Internetanwender und sollen in einem gängigen Format abgelegt werden. Der Kunde hat hierbei RealAudio bzw. RealVideo als Format festgelegt.

- Auf welchen Medien und Formaten ist bereis existierendes Material vorhanden? Welche Formate müssen vom System beim Import von Daten unterstützt werden?

  Der Kunde erwartet daß das System die Formate JPEG und GIF bei Bildern, sowie RealVideo und RealAudio (bzw. MP3) bei den Multimedia Daten verarbeiten kann.

- Sollen für Bilder/Video/Audiodokumente Zusatztexte/Stichwörter abgelegt werden können um ein Suchen in diesen Datenbeständen zu ermöglichen. Soll gegebenenfalls eine erweiterte Suchmöglichkeit bestehen (suche nach bestimmten Formen in Bildern oder Videos, entsprechende Software von Drittfirmen exisitert und wäre mit entsprechenden Aufwand anbildbar) ?

  Zusatzdaten wie Stichwörter zu den Videos sollen nicht abgespeichert werden. Allerdings wurde vom Kunden die Möglichkeit gewünscht z.B. Videodaten mit Personen oder Standortsangaben zu verknüpfen um nach ihnen suchen zu können.

- Welche Daten sollen noch zusaätzlich erfasst werden können (z.B. Lebenslauf, Literaturhinweise,...)?

  Übliche Daten wie z.B. Name, Geb. Dat, Verheiratet, Wohnhaft, E-Mail etc. Der Kunde überlässt es der SE AG geeignete Attribute vorzuschlagen.

- Löschen: Wirklich oder wird die komplette "Geschichte" eines Eintrages mitgeführt. d.h. auch der gelöschte Eintrag explizit als "gelöschter Eintrag"? Würde den Datenbestand vergößern aber auch eine UNDO Funktion ermöglichen bzw. Änderungen an den Daten dokumentierbar machen.

  Von Seiten des Kunden wird ein "Multilevel-Undo" gewünscht. Alle Änderungen (Deltas) sollen gespeichert werden. Damit ist ein beliebiges zurücknehmen von Aktionen möglich.

- Sind Typen für Attribute bekannt? z.B. String der Länge 30 oder String variabler Länge, Binärdaten variabler Länge,...

Der Kunde verläßt sich auf die Kompetenz der SE AG bei der Festlegung von geeigneten Bereichen, Größen,...

- Welche Verknüpfungen zwischen welchen Daten sind erforderlich oder erwünscht?

  Alle Daten sollen miteinander verknüpft werden können.

  Achtung. Der Kunde erwähnte das später eventuell neue archäologische Objekte zu den bereits festgelegten 3 Typen hinzukommen könnten! Das System sollte in der Lage sein damit umzugehen.

B.2.3 Benutzerklassen

- Wer bestimmt die Default-Rechte? Können diese vom Erfasser geändert werden oder kann dieser nur die Default Rechte für ein erfasstes Objekt abändern?

  Der Administrator soll die Default Rechte ändern können. Der Kunde soll dabei selber Administrator sein können.

- Sollen die Rechte beim Suchen für eine Benutzergruppe weiter einschränkbar sein als dies durch die Rechte für den Zugriff auf einzelne Attribute/Objekte sowieso schon gegeben ist? (z.B. Verhinderung der Verknüpfung von bestimmten Attributen beim Suchen, Einführung eines Rechts "Durchsuchbar" für ein Attribut?)

  Für einzelne Attribute soll als Recht auch festlegbar sein ob sie durchsucht werden dürfen.

- Sind wir, die SE AG, dafür zuständig die Benutzergruppen Administration durchzuführen oder übernimmt der Kunde dies?

  Der Kunde will selber Benutzergruppen Administrieren können.

- Was passiert, wenn ein Benutzer beim Einloggen ein falsches Passwort zu seiner Benutzerkennung eingibt? Denkbar wäre z.B. eine Benachrichtigung des Benutzers durch eine E-Mail mit der Aufforderung sein Passwort zu ändern.

  Der Kunde begrüßt dies. Es soll sowohl an den Administrator als auch den Benutzer selber eine Mail gesendet werden. Der Benutzer soll jedoch nicht gesperrt werden.

B.2.4 Web Frontend

- Gibt es bereits Vorgaben in welchen Format Audio und Videodaten zum Anwender übertragen werden (z.B. RealAudio, RealVideo, MPEG)?

  Der Kunde wünscht RealAudio und RealVideo.

- Soll es möglich sein das sich "Internetbenutzer" registieren und dadurch einen neuen Status erhalten (mit Username/Passwort)? Es könnten z.B. mehr Informationen (oder in einer besseren Qualität) für registierte Internetbenutzer angezeigt werden. Alternativ würde dadurch Werbung möglich, sowie eine Übername der Benutzerdaten/präferenzen in den Webshop?

  Der Kunde wünscht sich diese Möglichkeit. Er denkt an die Einführung eines Online-Fanklubs, Werbemails, etc.

- Soll das Aussehen (die Graphiken, die Anordung der Felder,...) des Web-Frontendes nachträglich geändert werden können (gegebenenfalls je Benutzergruppe)? (Verwendung von adaptierbaren Templates im System)

  Der Kunde will das Aussehen des Web-Frontendes selber ändern und verwalten können.

- Wie groß ist eine akzeptable Antwortzeit beim Web Frontend?

  Der Kunde stellt fest das dies schwer festzulegen ist. Es dürfte sich laut Kunden im 2-5 Sekunden bereich bewegen.

- Soll eine dynamische Anpassung der übertragenen Datenmenge an die Bandbreite des Anwenders möglich sein?

  Eine dynamische Anpassung ist nicht gewünscht. Der Benutzer hat jedoch die Wahl zwischen verschiedenen Qualitäten.

- Soll es möglich sein mithilfe der zwischen den Objekten definierten Beziehungen durch die Datenbank zu "surfen"?

  Dieser Punkt wird vom Kunden gefordert.

- Nach welchen "Meta-Daten" soll bei der erweiterten Suche gesucht werden können? Diese müssen intern vorgesehen werden. Denkbar wären z.B. Epoche, Stil,...

  Es soll nach Schlagwörtern gesucht werden können. Jedem Objekt können mehrere Schlagwörter zugeordnet werden (z.B. "Mittelalter"). Eine Meta Suche soll sich darauf beschränken diese Schlagwörter zu durchsuchen.

B.2.5 Virtuelles Museum

- Soll es im virtuellen Museum eine Suchfunktion / Orientierungshilfe (Guide) geben um es einfacher zu machen sich über bestimmte Interessensgebiete zu informieren? Wenn ja, über welche Funktionalität sollte dieser verfügen?

  Der Kunde begrüßt die Möglichkeit eines "Guides" bei einem virtuellen Mussum. Dieser könnte z.B. Informationen zu einembestimmten Artefakt geben oder auf den Museums-Webshop verweisen. Eine Suchmöglichkeit im virtuellen Museum soll eingeführt werden.

- Stellt eine virtuelle Tour durch das Museum nur einen vorgegebenen Weg dar oder verfügt sie über weitere Funktionalität (z.B. FührerIn welcher bei den entsprechenden Stationen Informationen zu den Objekten anzeigt oder in Form eines Audio/Videodokumentes wiedergibt)?

  Ein Führer ist bei Touren durch das virtuelle Museum erwünscht.

- Soll es möglich sein (je nach Benutzergruppe) sich die Form des Virtuellen Museums beim Besuch auszusuchen? Z.B. Ansehen im Web, Download als Katalog (z.B. PDF),...

  Ja, der Benutzer soll die Möglichkeit haben sich das auszusuchen.

B.2.6 Sonstiges

- Sind bereits Informationen über einen eventuellen Datenaustausch mit anderen Systemen bekannt? Wenn ja, mit welchen Systemen bzw. in welchen Formaten müssen die Daten ausgetauscht werden können?

  Es muß der Austausch mit Schwestersystemen möglich sein. Diese verwenden exakt die selbe Software und werden auch von der SE AG betreut - allerdings für einen anderen Kunden. Beide Systeme sind Soft und Hardwareseitig voneinander unabhängig. Es muß möglich sein das Objekte von einem System auf das andere System transferiert werden können (Ausleihe von Ausstellungsstücken). Diese haben aber nur eine bestimmte Gültigkeitsdauer. Diese muß vom Ausleiher definiert werden können. Dannach erfolgt auch die Abrechnung für ein Leihstück. Das vermietete Objekt wird als physikalische Kopie auf dem Schwestersystem gespeichert. D.h. insbesondere auch das sich das Leihstück auch nicht ändern darf, selbst wenn die ursprünglichen Daten geändert wurden.

- Gibt es eine Abnahmefrist für das System?

  Letzte Abnahmefrist ist Ende Jänner 2001 ;-)

- Welche Abnahmebedingungen gelten für die aufgeführten Punkte?

  Der Kunde erwartet von der SE AG das wir geeignete Abnahmekriterien und -bedingungen festlegen. Diese werden dann vom Kunden begutachtet und gegebenenfalls akzeptiert.

B.3 Aufgabe 3 (KW 46)

- Genügt es intern Bilder als JPEG oder GIF (mit 3 verschiedenen Auflösungen), Videos als MPEG und Audiodateien als MP3 zu speichern?

  Bilder werden intern als JPEG oder GIF gespeichert. Videos als Real-Video und Audiodateien als Real-Video. Zusätzlich werden (falls ein Freeware-Encoder verfügbar ist) Audiodateien als MP3 abgespeichert.

- Wird vom Kunden ein Assistent für die Änderung und die Verwaltung des Aussehen des Web-Frontends gewünscht, oder nicht (Fachkenntnisse!)?

Der Kunde gibt der SE-AG die Änderungswünsche bekannt und die SE-AG führt die Änderungen dann durch.

- Ist das Datenbankmanagement-System eine eigener Actor, d.h. ist die Datenbank im System integriert oder ist sie unabhängig vom System?

  Kann man diskutieren. Aber wir werden sie in Zukunft als eigenen Actor betrachten.

- Hat der Kunde bereits irgendwelche Vorstellungen welche zusätzlichen Rechte oder Möglichkeiten ein Internetbenutzer durch die Registrierung hat?

  Newsletter und Werbemail reichen.

- Werden irgendwelche zusätzlichen Tools (Datenbanken) für die registrierten Internetbenutzer benötigt? Werden Informationen über die Internetbenutzer in der Datenbank gespeichert (welche?)? Sollen automatisch (z.B. monatlich) Werbemails verschickt werden (aktuelle Informationen, Veranstaltungen)?

  Newsletter und Werbemail reichen. Es werden keine Tools benötigt.

- Sollem in real existierenden Museen Internet-Terminals aufgestellt werden, über die das Museum virtuell besucht werden kann (ev. auch andere Museen)?

  Der Kunde begrüßt diesen Vorschlag und hatte ihn sich bereits notiert.

- Hat der Internetbenutzer die Möglichkeit eine Tour anzuhalten und später fortzusetzen?

  Ist nicht notwendig.

- Wie detailiert soll die Datenbank geplant werden (relationales Schema)?

  Reicht so wie sie zur Zeit ist. Es ist nur wichtig, daß es eine Datenbank gibt?

- Einige Anmerkungen zum Use-Case-Diagramm:

  - Stereotypen werden durch Doppelpfeile gekennzeichnet (z.B. <<extends>>).
  - bei uses und extends werden offene Pfeile verwendet (siehe überarbeitetes use-case-Diagramm).
  - Beschreibungen der Use-Cases im Diagramm werden in folgender Form gemacht: Hauptwort Zeitwort.

    z.B. Objekte erfassen.
  - Bei der Beschreibung der use-cases war ürsprünglich gedacht, daß für jeden use-case ein Template verwendet werden sollte (unglückliche Formulierung führte zu einem Missverständnis). Ist bis zum nächsten Mal zu ändern!
  - Beschreibung der use-cases und der actors war genau wie gewünscht: kurz und pregnant!

B.4 Aufgabe 3 (KW 47)

- Welche Punkte im angewendeten Verfahren wurden von uns falsch abgeschätzt?
  - T5: Code must be reusable:

    Hoch (4 oder 5), da dazu auch die interne Strukturierung des Codes (effiziente Klassenstrukturierung, etc.) fällt.
  - T8: Portability:

    Hoch (4 oder 5), da das System leicht auf andere Schnittstellen (z.B. anderes Datenbanksystem) leicht portierbar sein soll.

- Wie hoch sind die Kosten eines Programmierers einzuschätzen?

  Die Arbeiterkammer gibt als Richtwert für die Kosten eines Programmierers mit 1400S pro Stunde inkl. Steuerabgaben und Lohnnebenkosten an.

Diese oben angegebenen Punkte müssen im Dokument korrigiert und die Berechnung des UCPs wiederholt werden!

- Müssen Gewinnspannen extra eingerechnet werden?

  Werden bereits in dem Richtwert der Arbeiterkammer berücksichtigt.

- Müssen sonstige Ausgabe ebenfalls dem Kunden verrechnet werden?

  Kann man. Sind allerdings schwer abzuschätzen. Es gibt weiters keine allgemeinen Richtlinien. Uns interessiert momentan dieser Aspekt nicht primär (soll nur eine interne Berechnung sein).

- Sollen die Kosten für die Erstellung dieses Dokument miteinbezogen werden?

  Sind prinzipiell in dem Berechnungsverfahren bereits inkludiert. Weiters liegt die das finanzielle Risiko für die Erstellung dieses Dokuments bei der SE AG.