Universität SALZBURG
Institut für Computerwissenschaften

Proseminar Software Engineering (WS '00/'01)
gehalten von Roland Schwaiger

ARIS

Archäologisches Information System

Gruppe III

ELSÄSSER Rene
relsaess@cosy.sbg.ac.at

FRONER Gerhard
gerhard.froner@sbg.ac.at

GÖTTLICH Rudolf
rudolf.goettlich@sbg.ac.at

SPANRING Harald
hspan@cosy.sbg.ac.at

Abstract

This document deals with the development of a software-system with the working title "ARchaeological Information System - ARIS" for entering, administering and presenting archaeological data via the internet.

Contents

Chapter 1

Productanalysis

Our missions is to develop a software-system with the working title "ARchaeological Information System - ARIS" for entering, administering and presenting archaeological data via the internet. The following sections will describe the software's requirements as provided by the customer so far. There are still open questions about several parts of the requirements, so further interviewing with the customer will be necessary.

1.1 General Goals

As main features, ARIS shall:

- Provide an interface via the world-wide-web (WWW) enabling our customer to enter archaeological data such as tombs, settlements or artefacts into a central database.
- Enable third parties to query the database for information via the WWW.
- Present the data via a virtual museum.
- Allow production of multimedia-CD-ROMs and printable exhibition catalogues.

We are responsible for the design and implementation of this system including the following points:

- Web-frontend to manipulate and display data.
- The design and implementation of the database itself.
- Maintain these services.
- Host the database on a server or at least administer this work.

Unknown are the stratetic objectives of the customer, i.e. if the customer pursues any profit-oriented goals like fund-raising, increasing visitor-count of a (real) museum, etc., or if the more altruistic goals like sharing knowledge with other scientists, etc. build the main motivation, or anything else. Further investigation on this issue is recommended to ensure that the resulting system will focus on the right purpose.

1.2 System Environment

ARIS has to be completely implemented as a web-based solution provided by us as an Application Service Provider (ASP). No other protocols than HTTP to access our web-server via a web-browser are allowed on the client side.

1.2.1 Data Quantity

The customer has by now estimated an amount of 400 pictures p.A., 400 drawings p.A. and 10 videos (length 10sec.) p.A, unknown yet is the amount of data-items (artefacts) p.A.

1.2.2 Data Formats

The customer insists on the use of XML as the only data-format in order to be as platform-independent as possible. Still unknown are the real purposes which XML shall be used for, and if HTML is a valid format, too.

1.2.3 Hardware

We provide two PCs as servers, specified as PIII 700Mhz, 512MB RAM, 50GB diskspace. One of these will be used as web-server, the other as database-server. A streamer can be used for backups.

1.2.4 Software

We are fully responsible for the server-software used. Currently we suggest Linux 2.2 as operating system, Apache 1.3.x as web-server with extensions for JSP or PHP3/4 as development framework (the final choice will depend on the final draft of requirements and the developer-team choosen to implement the system). As database-engine, we suggest the latest version of mysql. The choice for the server-software are based on the facts that all of them are open-source products, which guarantees both high-reliability and low license cost. On the client-side, the customer provides already installed web-browsers, the vendor/version of these is yet unknown. No other software may be installed on the client-side!

1.2.5 Internet Connectivity

The customer is equipped with an ADSL-Connection with a dedicated line to enable permanent access to our software-system. Our network facilities are not specified by now as they depend on the estimated network-load, which depends mainly on the estimated count of user-accesses from the the internet. As a rule of thumb, a server with a 100Mbit network card and a 2Mbit backbone will be able to serve at least 100 connections (users) per second, though most of the time the bandwith is limited by the user-connection (and the bandwidth of his/her ISP), not the server-connection, which decreases the expected bandwidth-usage on the server-side very much.

1.2.6 Open Questions

- Is there currently a homepage or a special corporate design our software-system has to be adopted to?
- Are there any other ASPs providing such a homepage?
- How many data-items (artefacts) will be entered into the system p.A.?
- How many attributes per data-item will be needed?
- What shall XML be used for? Will the customer use an XML-enabled web-browser, or shall XML be used as basis to generate HTML inside the web-server (e.g. to support different web-browsers), or is XML planned to be used for data-export?
- Are there any other WWW-technologies as Flash/Java/Javascript allowed?
- How many user-accesses per second shall be possible?
- Which web-browser have to be supported?

1.3 User Interfaces

This part of the product analysis will be omitted until the customer has provided more detail on how the system should be accessed.

1.4 Functional Requirements

In the following sections, the functional requirements of the software-system will be specified. For a more structured reading, the requirements are grouped in functional realms.

1.4.1 Basic Data Entry

The system shall provide a web-frontend to enter archaeological data such as tombs, settlements and artefacts. Data entry shall be restricted to 'collectors' and 'scientific collectors'.

Basic Data Entry Form: The system must contain an input form to enable entering of archaeolgical data.

Type: MUST

Priority: 1

Rationale: This enables data entry.

Basic Data Entry Search and List: There must be a way to find a data entry, either through a list of found items after a search or a full list of data-items.

Type: MUST

Priority: 2

Rationale: Without a navigation, old data-items could not be modified.

Insert of a Data Entry: There must be a way to insert data entry, ideally via the data entry form.

Type: MUST

Priority: 2

Rationale: Without a possibility to input a data entry, no new data could be added.

Deletion of a Data Entry: There must be a way to delete a data entry, ideally via the data entry form, or picked directly from a list provided via a search.

Type: MUST

Priority: 2

Rationale: Without a possibility to delete a data entry, erronous data could not be removed.

Update of a Data Entry: There must be a way to update a data entry, ideally via the data entry form.

Type: MUST

Priority: 2

Rationale: Without a possibility to update a data entry, modification could only take place via deletion and re-entering of a data-item.

Access Restrictions: Only authenticated collectors may enter or modify data-items. A logon process is needed.

Type: MUST

Priority: 2

Rationale: Data-items may not be modified by anonymous users.

1.4.2 Scientific Data Entry

The system shall provide a web-frontend to enter additional scientific data related to entered data-items. This data entry shall be restricted to 'scientific collectors' only.

Adding Documents: There must be a way to add scientific documents to an existing data-item. Ideally the document can just be added via an upload-form.

Type: MUST

Priority: 2

Rationale: Additional documents may be needed for further and specialized information.

Different Data Types: It must be possible to enter any important data-type (tombs, settlements, artefacts). This may require different input forms.

Type: MUST

Priority: 2

Rationale: Any important data-type must be supported.

Adding Document-links: There should be a way to add links to external scientific documents to an existing data-item.

Type: SHOULD

Priority: 5

Rationale: There might be external ressources which the collector wants to refer to (e.g. to avoid a long upload).

Viewing Documents of a Data-items: There should be a way to view all documents added to a data-item to ease later modification via picking and editing a document.

Type: SHOULD

Priority: 3

Rationale: Document links might be modified later.

Updating Documents of a Data-items: There should be a way to update a document added to a data-item.

Type: SHOULD

Priority: 3

Rationale: Document links might be modified later.

Deleting Documents of a Data-items: There should be a way to delete a document added to a data-item.

Type: SHOULD

Priority: 3

Rationale: Documents might be deleted later.

Pictures: It should be possible to add a picture to a data-entry. Ideally, the picture can be added in the same form where the data is entered.

Type: SHOULD

Priority: 3

Rationale: Pictures will be needed for data-display.

Videos: It should be possible to add a video to a data-entry. Ideally, the video can be added in the same form where the data is entered.

Type: SHOULD

Priority: 5

Rationale: Videos might be needed for data-display.

Access Restrictions: Only authenticated scientific collectors may enter or modify data-items. A logon process is needed.

Type: MUST

Priority: 2

Rationale: Data-items may not be modified by anonymous users, nor normal collectors.

1.4.3 Data Relations

The customer need the possibility to define relations via a web-frontend between a group of data-items, or between a data-item and a scientific document. We assume, that a group of data-item is a data-item itself, which has to be entered before defining relations on it.

Adding Relations between Data-items: There should be a way to add relations between two data-items. Possible relations are grouping and similarity.

Type: MUST

Priority: 3

Rationale: Relations between data-items should be displayed, and thus have to be defined by collectors.

Adding Relations between a Data-item and a Document: There should be a way to add relations between a data-items and a scientific document. The only possible relations is 'additional information'. Documents may reside locally or externaly.

Type: MUST

Priority: 3

Rationale: Relations between a data-items and documents should be displayed (e.g. as link), and thus have to be defined by collectors.

Viewing Relations on a Data-items: There should be a way to view all relations defined on a data-item to ease later modification via picking and editing a relation.

Type: SHOULD

Priority: 3

Rationale: Relations might be modified later.

Updating Relations on a Data-items: There should be a way to update a relation defined on a data-item to ease later modification.

Type: SHOULD

Priority: 3

Rationale: Relations might be modified later.

Deleting Relations on a Data-items: There should be a way to delete a relation defined on a data-item to ease later modification.

Type: SHOULD

Priority: 3

Rationale: Relations might be deleted later.

Access Restrictions: Only authenticated collectors may enter or modify data-relations. A logon process is needed.

Type: MUST

Priority: 2

Rationale: Data-relations may not be modified by anonymous users.

1.4.4 Data Search and Display

Users of the software-system shall have the possibility to search for data-items via a web-frontend, pick and view any of them. The type of data presented is restricted.

Search for Data-items: There must be a way to search for data-items. Results should be presented as a list where the user can pick a single item.

Type: MUST

Priority: 3

Rationale: Users should find data-items.

Display of a single Data-items: A single data-item must be displayed to the user with any data the user is allowed to see. The view may include pictures.

Type: MUST

Priority: 1

Rationale: Users should view data-items.

Relations of Data-items.: A single data-item may be displayed to the user with one or more links to related data-items.

Type: MUST

Priority: 3

Rationale: Users should be able to see relations between data-items.

Download of Documents, Videos, etc.: A single data-item may be displayed to the user with one or more links to related documents, videos, etc.

Type: MUST

Priority: 4

Rationale: Users should be able to download additional information.

Access Restrictions: Only certain types of data may be displayed (maybe even found).

Type: MUST

Priority: 2

Rationale: There is either confidential or too much data which the user should not get in contact with.

1.4.5 Virtual Museum

The customer wants an addiontal presentation of the data-items grouped in a virtual museum, accessible via a web-frontend. The rooms, show-cases and data-items have to be administrable. The view of items should be able to be modified to some extent.

Room Administration: It must be possible to create virtual rooms. Rooms have an additional text describing their theme/contents.

Type: MUST

Priority: 5

Rationale: The museum is not static.

Show-case Administration: It must be possible to create virtual show-cases, and to add them to virtual rooms. Show-cases have additional text describing there contents. Collectors should be able to add data-items to show-cases.

Type: MUST

Priority: 5

Rationale: The museum is not static.

Display of Rooms, Show-cases: The user will need some view of the room/show-case it is currently watching.

Type: MUST

Priority: 4

Rationale: The user walks through the virtual museum to view data-items.

Navigation through Rooms, Show-cases: The user will need some navigation from room/show-case to room/show-case.

Type: MUST

Priority: 4

Rationale: The user walks through the virtual museum to view data-items.

Customizing: The collectors should have the possibility to influence the appearance of rooms/show-cases.

Type: SHOULD

Priority: 6

Rationale: Some rooms/show-cases might need special visualization.

Access Restrictions: Administration of the virtual museum may only be accessed by collectors.

Type: MUST

Priority: 2

Rationale: Normal users may only view but not alter a museum.

1.4.6 Administration

Besides the data-item and virtual-museum administration, which will be executed by collectors, there are further things to administer, currently only user-logins and -privileges. This work will be done by the 'mighty' administrators.

Viewing User-logins: Administrators need to know about any existing user-logins.

Type: MUST

Priority: 7

Rationale: The system needs access-restrictions, thus authentication is needed, which must be configured.

Adding User-logins: Administrators may add user-logins.

Type: MUST

Priority: 7

Rationale: The system needs access-restrictions, thus authentication is needed, which must be configured.

Modifying User-logins: Administrators may modify user-logins.

Type: MUST

Priority: 8

Rationale: The system needs access-restrictions, thus authentication is needed, which must be configured.

Deleting User-logins: Administrators may delete user-logins.

Type: MUST

Priority: 8

Rationale: The system needs access-restrictions, thus authentication is needed, which must be configured.

User Groups: There must be user-groups defining the privileges of administrators, scientific collectors, collectors and normal users. Administrators add a user to exactly one user-group.

Type: MUST

Priority: 2

Rationale: The system needs access-restrictions, thus authorisation is needed, which must be configured.

Access Restrictions: Administration of user-logins and -privileges may only be executed by administrators.

Type: MUST

Priority: 2

Rationale: Any other user could promote himself to an administrator else.

1.4.7 Exhibition CD

The customer wants to produce exhibitions CDs containing data of parts of the virtual museum. We assume that producing such a CD is possible the easiest way by exporting subsets of the database as XML and converting the data automatically to linked static HTML-files which can be browsed without a web-server.

XML Export: There should be a way to export parts of the virtual museum as XML-files.

Type: SHOULD

Priority: 7

Rationale: XML export enables conversion to static HTML-files.

XML to HTML Conversion: There should be a way to convert exported XML-files to linked static HTML-files.

Type: SHOULD

Priority: 7

Rationale: Static HTML-files are the basis of the exhibition CD.

CD-archive Download: There should be a way to download a compressed archive-file (e.g. a ZIP-archive) of a set of HTML-files building an exhibition CD.

Type: SHOULD

Priority: 7

Rationale: This frees us from the duty to burn and send such CDs via 'snail-mail'.

Access Restrictions: Exporting a exhibition CD might need special privileges.

Type: SHOULD

Priority: 8

Rationale: Data export is time- and load-consuming.

1.4.8 Exhibition Catalogue

The customer wants to produce printed exhibitions catalogues containing data of parts of the virtual museum. We assume that producing such a catalogue is possible the easiest way by exporting subsets of the database as XML and converting the data automatically to linked PDF-files which can be browsed with the Acrobat Reader, or printed with any standard printer.

XML Export: There should be a way to export parts of the virtual museum as XML-files.

Type: SHOULD

Priority: 7

Rationale: XML export enables conversion to static PDF-files.

XML to PDF Conversion: There should be a way to convert exported XML-files to linked PDF-files.

Type: SHOULD

Priority: 7

Rationale: PDF-files are the pre-stage of the printed exhibition catalogue.

PDF-archive Download: There should be a way to download a compressed archive-file (e.g. a ZIP-archive) of a set of PDF-files building an exhibition catalogue.

Type: SHOULD

Priority: 7

Rationale: This frees us from the duty to print and send such catalogues via 'snail-mail'.

Access Restrictions: Exporting a exhibition catalogue might need special privileges.

Type: SHOULD

Priority: 8

Rationale: Data export is time- and load-consuming.

1.5 Non-functional Requirements

The customer still has to define non-funtional requirements for the following benchmarks:

- maximum response-time
- maximum page-hits/minute
- maximum data quantity
- maximum network bandwidth
- maximum size of documents, images, videos, ...
- maximum allowed down-time
- actuality of data backups

1.6 Error Management

We will monitor the system for any failures from the outside with automated tools. The dynamic pages on the server will be able to detect failure of the database-server and report this to the user in a decent way. If the web-server fails, or the machine crashes, the user will not be able to access the system, but our monitor system will immediatly inform our operators.

1.7 Acceptance Criteria

The customer still has to define its acceptance criterias to the already defined requirements.

Chapter 2

Risc Analysis

This chapter handles the different risks which threat the project.

2.1 Technical Risks

2.1.1 Server Performance

The current server hardware should suffice even if the system grows more than expected, in the worst case, we could always move the system to a bigger machine (no big porting efforts will be needed).

2.1.2 Disk Space

The current diskspace (50GB) should suffice for a long time, in the worst case, if the space needed by large documents exceeds the available diskspace, we could easily extend the system by mounting another harddisk to the system.

2.1.3 Response Time

The network-bandwidth and performance of the servers should suffice as stated in the system-environment chapter. In the worst case, if our system suffers of bandwidth-bottlenecks on our side or at our provider's side, we could always change our provider and/or add a faster network-card to the system (e.g. 1GBit ethernet card). Performance issues were discussed above already.

2.1.4 Reliability

Through backup and monitoring, we will notice any failure early and will always be able to restore the servers from the data-backups with data which is at least not older than one day, so data loss would be minimum the data of a days work. If data recovery has to be more complete, when backup is restored, a RAID system would be needed to always have redundant storage of data. In this case, no data loss can occur, though backup is still recommended to manage user-faults like unintentional deletion of data. For hardware errors, a stock of spare parts should be held.

2.1.5 Incompatibilities

The probability of incompatibilities is massively reduced by the use of HTML/XML, though different browser-versions could make problems, for this reason, browser-specific implementations should be avoided, or being addressed by using XML as the base document representation and different style-sheets for each browser to generate specialized HTML-documents for each browser.

2.2 Legal Risks

2.2.1 Outsourced Services

We do not outsource any services except the internet connection, which is provided by our ISP. In case of failure of our ISP, it is easily possible to find alternative ISPs.

2.2.2 Data Protection

It has to be cleared if we (or our customer) needs a DVR-number.

2.2.3 Copyrights

The customer has to be interviewed about any copyright issus.

2.3 Staff Risks

We are an international accepted company with highly motivated employees, which are allrounders in many realms. The experiance gain from many other Web/ASP projects has established a rich knowledge of the technologies needed with this project. A high rate of fluctuation or brain drain is therefor not likely. The current amount of free resources should suffice to grant a decent delivery date.

2.4 Defect Risks

Our development is supported by sophisticated testing-techniques and quality-assurance, so defects caused by our development are reduced to a minimum. Any defect which is not to be classified as harmful will require a support-contract to be treated by our support-department. Past experiences have shown that our support-department can cope with bug-reports and user-complaints very well. Any defect after delivery, which is not a functional defect, but caused by the absence of a pre-existing functional requirement, will not be fixed until the customer starts a new project-iteration.

2.4.1 Data Loss

We do not accept any warranty for data loss caused by software-defects. Our backup should suffice to restore most data in case of data loss, though.

2.5 Image Risks

Our customer seems to be a reputable person/company, so this project would be another good reference. Risk of failure of the whole project is low, as we did enough similar projects by now.

2.6 Financial Risks

Our solvency is not threatened as funding of the project is insured by a bank guarantee, which insurances project-costs if the customer gets out of money. Our current revenues from other projects allow us to pre-finance the project, in the worst case, we could demand interim payment from the customer, when bigger steps in the project have been completed.

Chapter 3

Use Cases

This chapter deals with the planned functional units (use cases), their environment/users (actors) and the relations between them (use case diagram).

3.1 Actors

User: The system will be visited from anonymous users via the internet. A user will search for artefacts to get data about them. He/she will also visit the virtual museum.

Registered User: Registered users have full read access to the system, but must not enter/modify any data.

Collector: The collector enters the basic data of artefacts.

Scientific collector: The scientific collector adds additional documents to the artefacts. The scientific collector may enter basic data just like a normal collector.

Administrator: The administrator creates new user-logins and grants privileges to them.

3.2 Primary Use Cases

3.2.1 EnteringBasicData

Description

The collectors and scientfic collectors enter new settlements, tombs and artefacts into the system. Each data-item has several attributes. To modify existing data-items, there must be a way to search and select them for updates or deletion.

Actors

- Collectors
- Scientific collectors

Priority

1

Status

Filled

Preconditions

The user must be logged on to be able to enter some artefacts.

Postconditions

The modified artefacts have been saved to the database.

Extension Points

AccessRestrictions, ErrorHandling

"Used" Use Cases

Insert, DisplayArtefactForEdit, Search, SelectArtefact

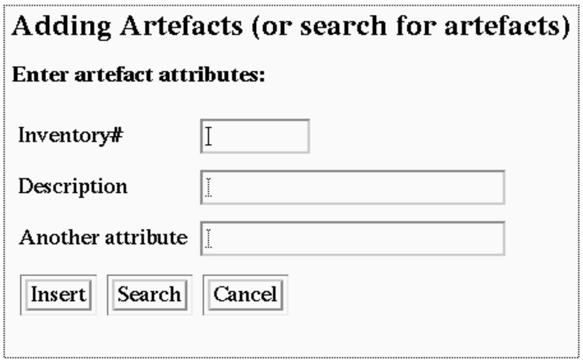
Flow of Events

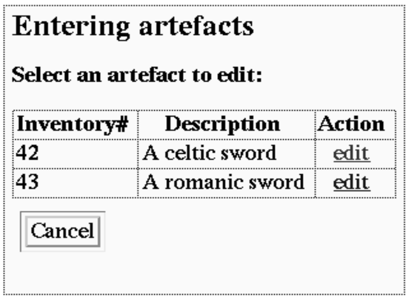
1. The use case starts with the selection of 'EnterBasicData'.
2. The browser shows an empty fill out form for artefacts.
3. The user enters an inventory number or other data.
4. if (User clicks button 'Insert')
    1. (System enters new use case 'Insert')
    2. while (inventory number already exists)
        1. Show Warning: 'Duplicate Inventory Number!'
        2. Show filled form with writeable inventory number.
        3. User should enter another inventory number.
        4. User clicks 'Insert' or 'Cancel'.
        5. if (User clicks 'Cancel')
            1. Use case 'EnterBasicData' starts again.
        6. end if
    3. end while
    4. System saves the new artefact.
    5. System enters new use case 'DisplayArtefactForEdit'...

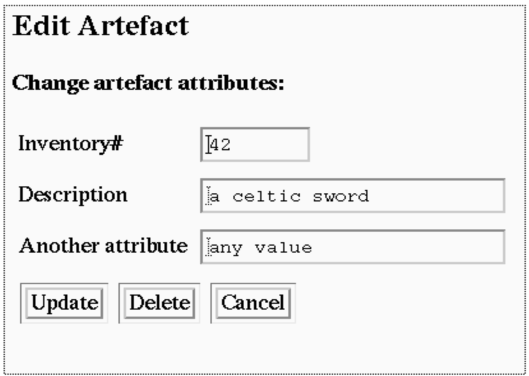
5. else if (User clicks 'Search')
    1. if ($resultCount < 1$)
        1. Show Warning "No matching artefacts found!"
        2. Use case 'EnterBasicData' starts again.
    2. else if ($resulCount == 1$)
        1. System enters new use case 'DisplayArtefactForEdit'...
    3. else if ($resultCount > 1$)
        1. (System enters new use case 'SelectArtefact')
        2. Show list of matching artefacts with links for selection.
        3. User clicks one edit link (User selects one artefact).
        4. System enters new use case 'DisplayArtefactForEdit'...
    4. end if
6. end if

Activity Diagram

User Interface







Secondary Scenarios

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.2.2 AddDocument

Description

The scientific collector adds hyperlinks from existing data-items like artefacts pointing to scientific documents or other ressources (videos, etc.). This documents/ressources may already exist on external locations in the internet or the collector may upload them. External hyperlinks have to be deletable and modifyable, internal just have to be deleteable. Uploaded documents will be removed when the associated hyperlink is deleted.

Actors

- Scientific collector

Priority

1

Status

Filled

Preconditions

The user must be logged on to be able to add documents.

Postconditions

Local Documents are uploaded. Modified Hyperlinks have been saved.

Extension Points

AccessRestrictions, ErrorHandling

"Used" Use Cases

Insert, Search, SelectDocumentsFromArtefactList, UploadDocument, AddExternalDocumentLink

Flow of Events

1. The use case starts with the selection of 'AddDocument'.
2. The browser shows an empty fill out form for artefact searching.
3. The user enters an inventory number or other data.
4. User clicks 'Search'
5. (User enters new use case 'Search')
6. if ($resultCount < 1$)
    1. Show Warning "No matching artefacts found!"
    2. Use case 'AddDocument' starts again.
7. else if ($resulCount == 1$)
    1. continue
8. else if ($resultCount > 1$)
    1. (System enters new use case 'SelectDocumentsFromArtefactList')
    2. Show list of matching artefacts with links for selection.
    3. User clicks one edit link (User selects one artefact).
9. end if
10. (System enters new use case 'DisplayDocumentListForArtefact')
11. Show list of documents previously added to the selected artefact. Each row contains an edit-link.
12. User clicks 'Add' or selects one existing document for edit.
13. if (User selects one existing document)
    1. System enters new use case 'DisplayDocumentForEdit'...
14. else if (User clicks 'Add')

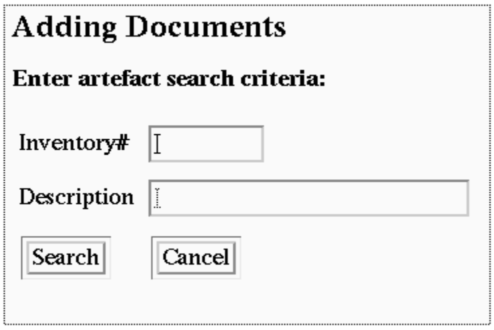
   1. (System enters new use case ’Insert’)
   2. Show empty fill out form for adding documents.
   3. User fills in a document description.
   4. User enters a file-name (with path) to upload from its local computer or enters an external URL..
   5. User selects a radio-button if he/she wants to upload a document or to reference an external document.
   6. User clicks ’Save’.
   7. if (User selected radio-button ’upload’)
      1. (System enters new use case ’UploadDocument’)
      2. if (document-path+filename is undefined)
         1. Show Warning ”no file to upload defined!”
         2. System enters use case ’Insert’ again, but with filled data.
      3. end if
      4. if (description is undefined)
         1. Show Warning ”no description of document defined!”
         2. System enters use case ’Insert’ again, but with filled data.
      5. end if
      6. System uploads the defined file and stores a link to the document with the given description.
   8. else if (User selected radio-button ’external’)
      1. (System enters new use case ’AddExternalDocumentLink’)
      2. if (URL is undefined)
         1. Show Warning ”no URL defined!”
         2. System enters use case ’Insert’ again, but with filled data.
      3. end if
      4. if (description is undefined)
         1. Show Warning ”no description of document defined!”
         2. System enters use case ’Insert’ again, but with filled data.
      5. end if
      6. System stores an external link to the document with the given description.
   9. end if

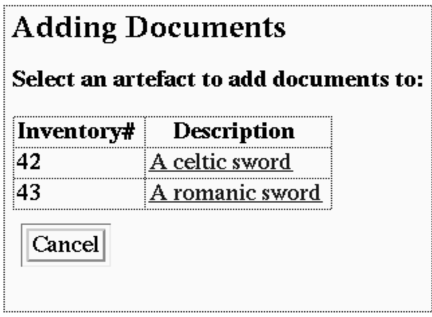
15. (System enters new use case ’DisplayDocumentListForArtefact’)...

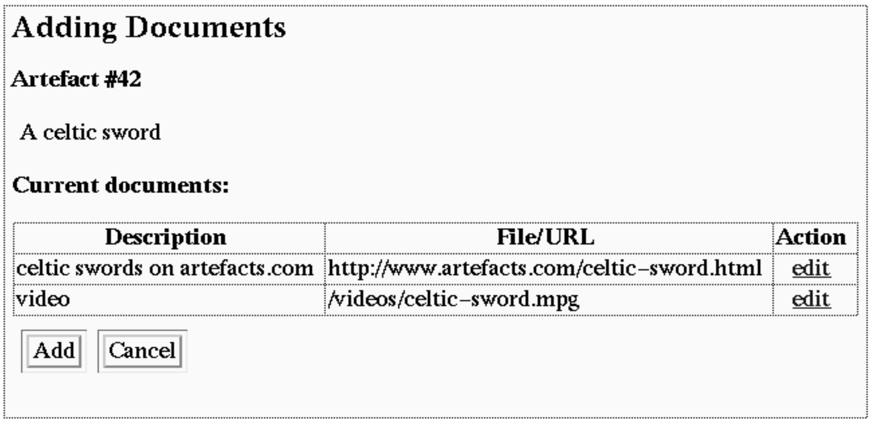
16. end if

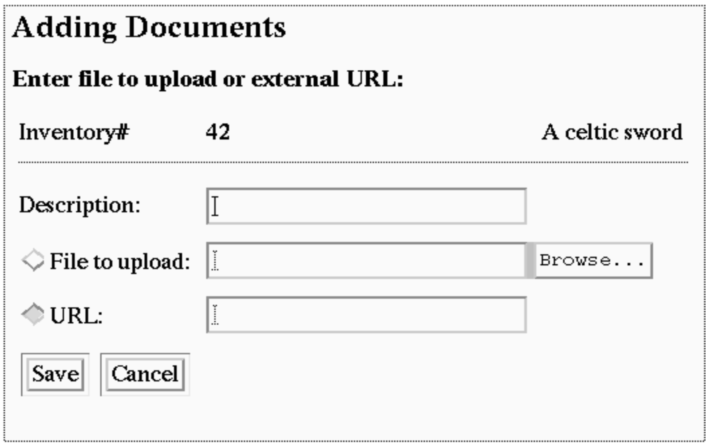
Activity Diagram

User Interface









Secondary Scenarios

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.2.3 Defining Relations

Description

Data-items may be related to each other: e.g. a settlement contains tombs, a tomb contains artefacts, but an artefact may also be a direct part of a settlement, or part of another artefact. This relation may be entered between existing data-items at any time. A generic relation may also be defined between any data-items (e.g. a sword relates to a similar sword found on another settlement). Collectors define and modify these relations.

Actors

- Collector
- Scientific collector

Priority

How important is this use case to the project?

Status

Facade

Preconditions

A list of conditions that must be true before the use case starts.

Postconditions

A list of conditions that must be true when the use case ends, no matter which scenario is executed.

Extension Points

If the use case has extension points, list them here.

"Used" Use Cases

If the use case uses other use cases, list them here.

Flow of Events

This could be a basic path and alternative paths, or the primary scenario.

1. *Preconditions*
2. *Main Flow*
3. *Subflows (if applicable)*
4. *Alternative Flows*

Activity Diagram

An activity diagram of the flow of events, or some significant or complex part of the flow of events.

User Interface

For systems that interface with people, include a description of the user interface, possibly using storyboards.

Secondary Scenarios

If alternatives and exceptions are not shown in the flow of events, scenarios should be listed here, and a brief description may be included.

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.2.4 Searching Artefacts

Description

The user will get an input form which allows him to search for various attributes.

Actors

- User
- Registerd user
- Collector
- Scientific collector

Priority

Must 1

Status

Filled

Preconditions

Filled

Postconditions

the user must be logged on

Extension Points

AccessRestriction, errorHandling

"Used" Use Cases

Select

Flow of Events

1. use case begins with selection of**Searching Artefact Menue**
2. admin select either Searching Inventarnummer, Searching all Attributes, Full text Search or exits
3. if(Searching Inventarnummer)
   3.1 user inputs**Inventarnummer**
   3.2 if (no Inventarnummer)
      3.2.1 display InvNr not found
      3.2.2 user inputs **Inventarnummer**
   3.3 else display data by access
4. admin selct either Searching Inventarnummer, Searching all Attributes, Full text Serch or exists
5. if(Searching all Attributes)
   5.1 user inputs various Attributes
   5.2 if(no results)
      5.2.1 display no result
      5.2.2 user inputs various Attributes -¿ 5.1
   5.3 else display list of results
   5.4 user chooses one of the results
   5.5 display data by access
   5.6 new selection of data list -¿ 5.3
6. admin selct either Searching Inventarnummer, Searching all Attributes, Full text Serch or exists
7. if(Full text Search)
   7.1 input Searchtext
   7.2 if(no results)
      7.2.1 display no result
      7.2.2 user inputs various Attributes -¿ 5.1
   7.3 else display list of results
   7.4 user select one of the results

   7.5 display data by access

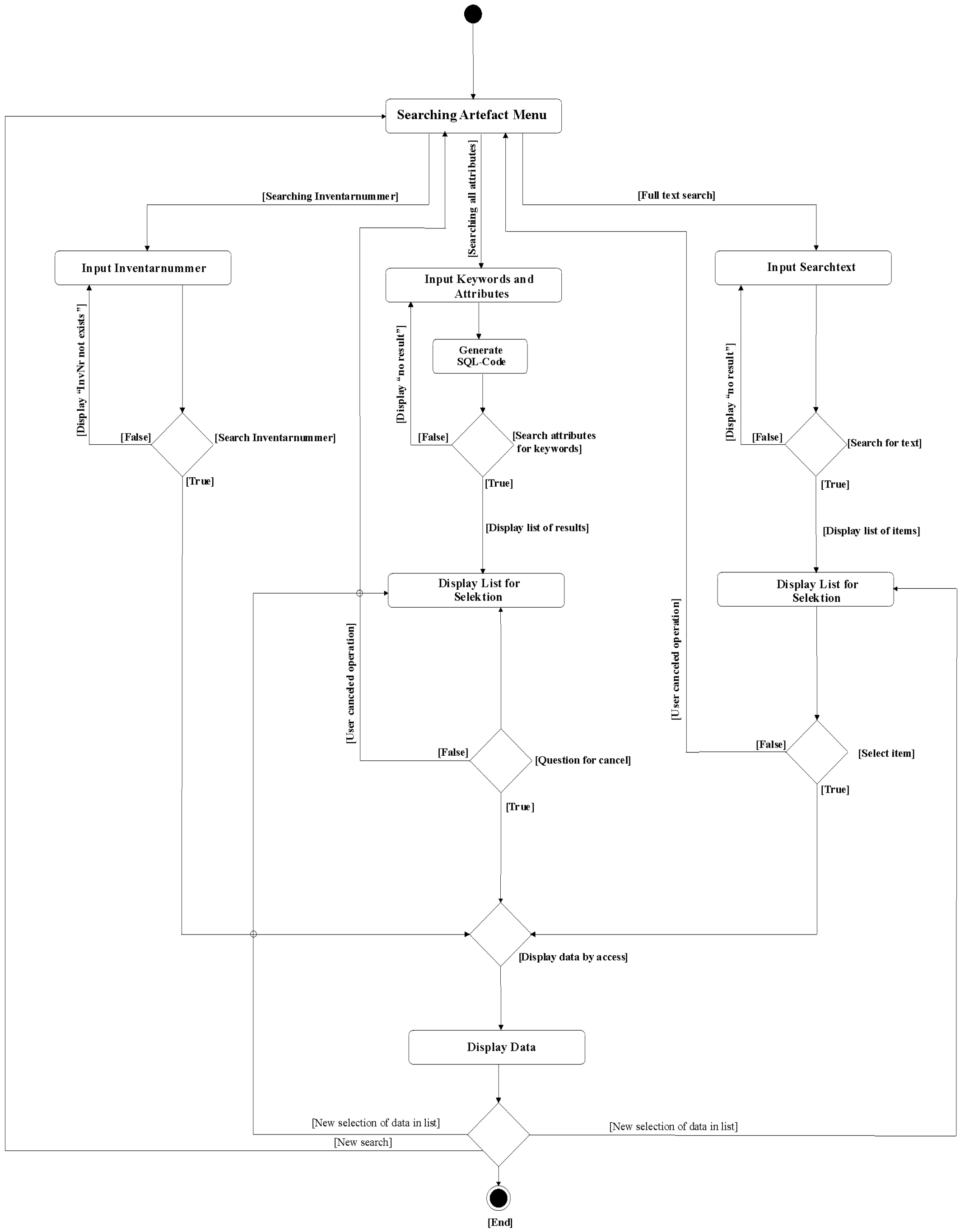
   7.6 new selection of data list -¿ 7.3

8. admin selct either Searching Inventarnummer, Searching all Attributes, Full text Serch or exists

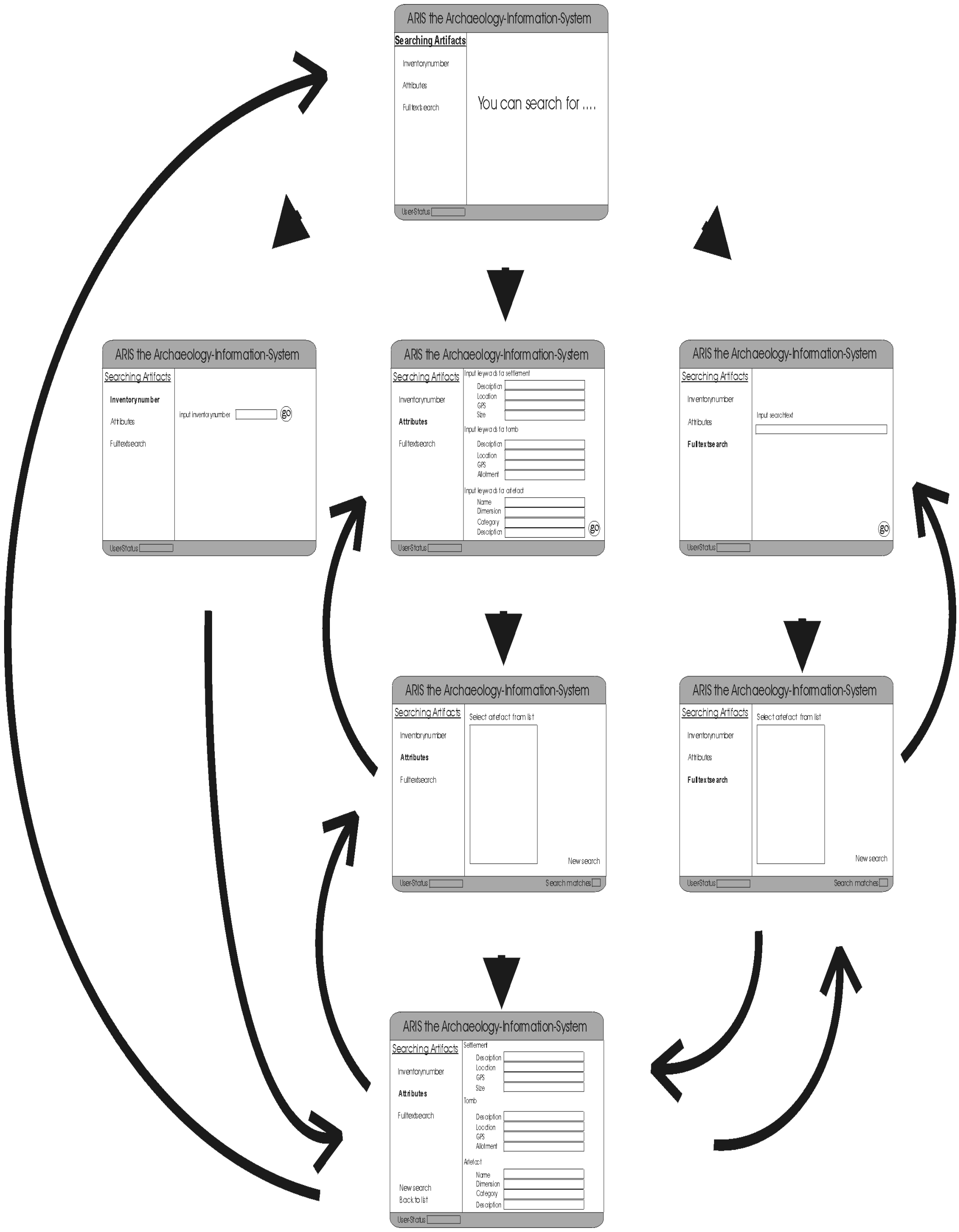
9. if(exit)

   9.1 leave Searching Artefact Menue

Activity Diagram



User Interface



Secondary Scenarios

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.2.5 Virtual Museum

Description

The user can walk trough a virtual museum, wandering from room to room, selecting show-cases and selecting artefacts contained in the show-cases for a detailed view. Rooms and show-cases have additional insignias describing their content. Rooms and show-cases are defined by collectors through a frontend provided by the system.

Actors

- User

Priority

How important is this use case to the project?

Status

Facade

Preconditions

A list of conditions that must be true before the use case starts.

Postconditions

A list of conditions that must be true when the use case ends, no matter which scenario is executed.

Extension Points

If the use case has extension points, list them here.

"Used" Use Cases

If the use case uses other use cases, list them here.

Flow of Events

This could be a basic path and alternative paths, or the primary scenario.

1. *Preconditions*
2. *Main Flow*
3. *Subflows (if applicable)*
4. *Alternative Flows*

Activity Diagram

An activity diagram of the flow of events, or some significant or complex part of the flow of events.

User Interface

For systems that interface with people, include a description of the user interface, possibly using storyboards.

Secondary Scenarios

If alternatives and exceptions are not shown in the flow of events, scenarios should be listed here, and a brief description may be included.

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.2.6 User Administration

Description

The administrator creates user-logins and grants privileges (registered user, collector, scientific collector or administrator) to them. Of course user-logins can be modified or deleted.

Actors

- Administrator

Priority

Must1

Status

Filled

Preconditions

Filled

Postconditions

loged on as admin; group-privileges must be defined

Extension Points

new user data are added

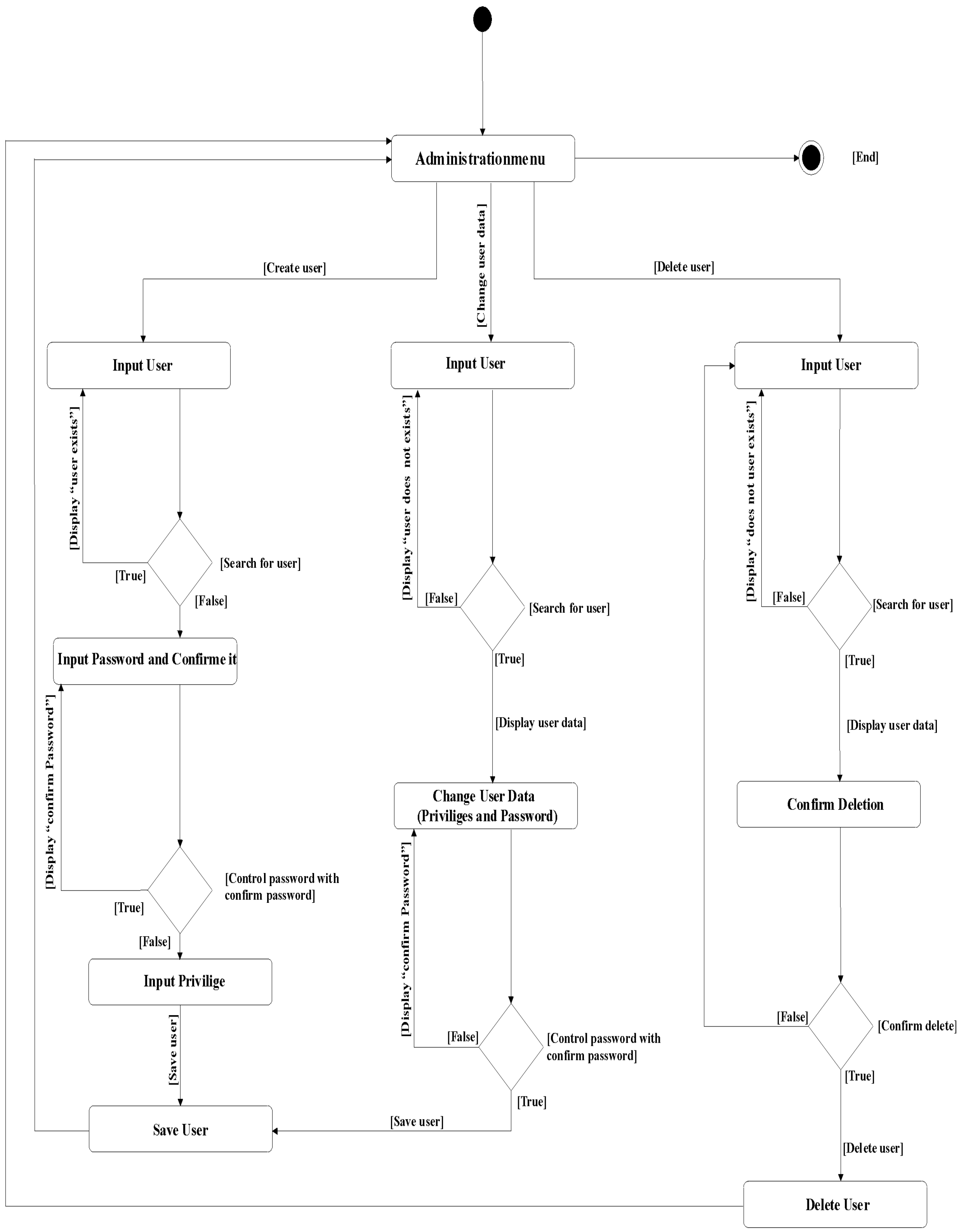
"Used" Use Cases

AccessRestrictions, ErrorHandling Insert, Search

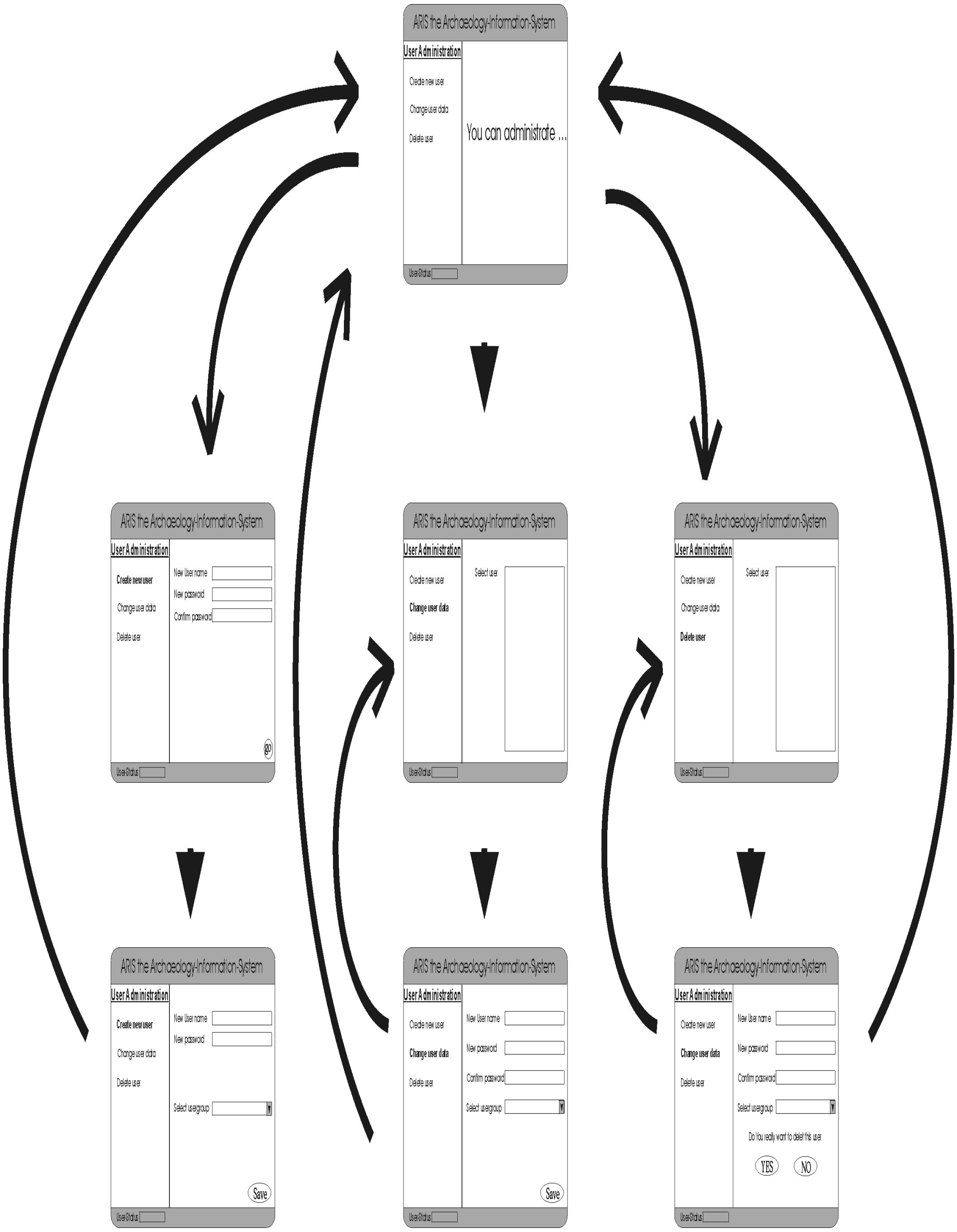
Flow of Events

1. use case begins with selection of **Administrationmenue**
2. admin select either create user, change user data, delete user or exits
3. if(create user)
   (a) admin inputs **user, password** and confirm
   (b) while (user exits or confirm password is wrong)
      i. admin inputs **user, password** and confirm
   (c) end
   (d) admin **inputs privileges**
   (e) admin **saves user**
4. admin select either create user, change user data, delete user or exits
5. if(change user data)
   (a) admin **inputs User**
   (b) while (user not exits)
      i. admin **inputs User**
   (c) admin **changes user data**(Privileges or password)
   (d) while (confirm password is wrong)
      i. admin **changes user data**
   (e) admin saves user
6. admin select either create user, change user data, delete user or exits
7. if (delete user)
   (a) admin **inputs User**
   (b) while (user not exits)
      i. admin **inputs User**
   (c) admin delete user
8. admin select either create user, change user data, delete user or exits

Activity Diagram



User Interface



Secondary Scenarios

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

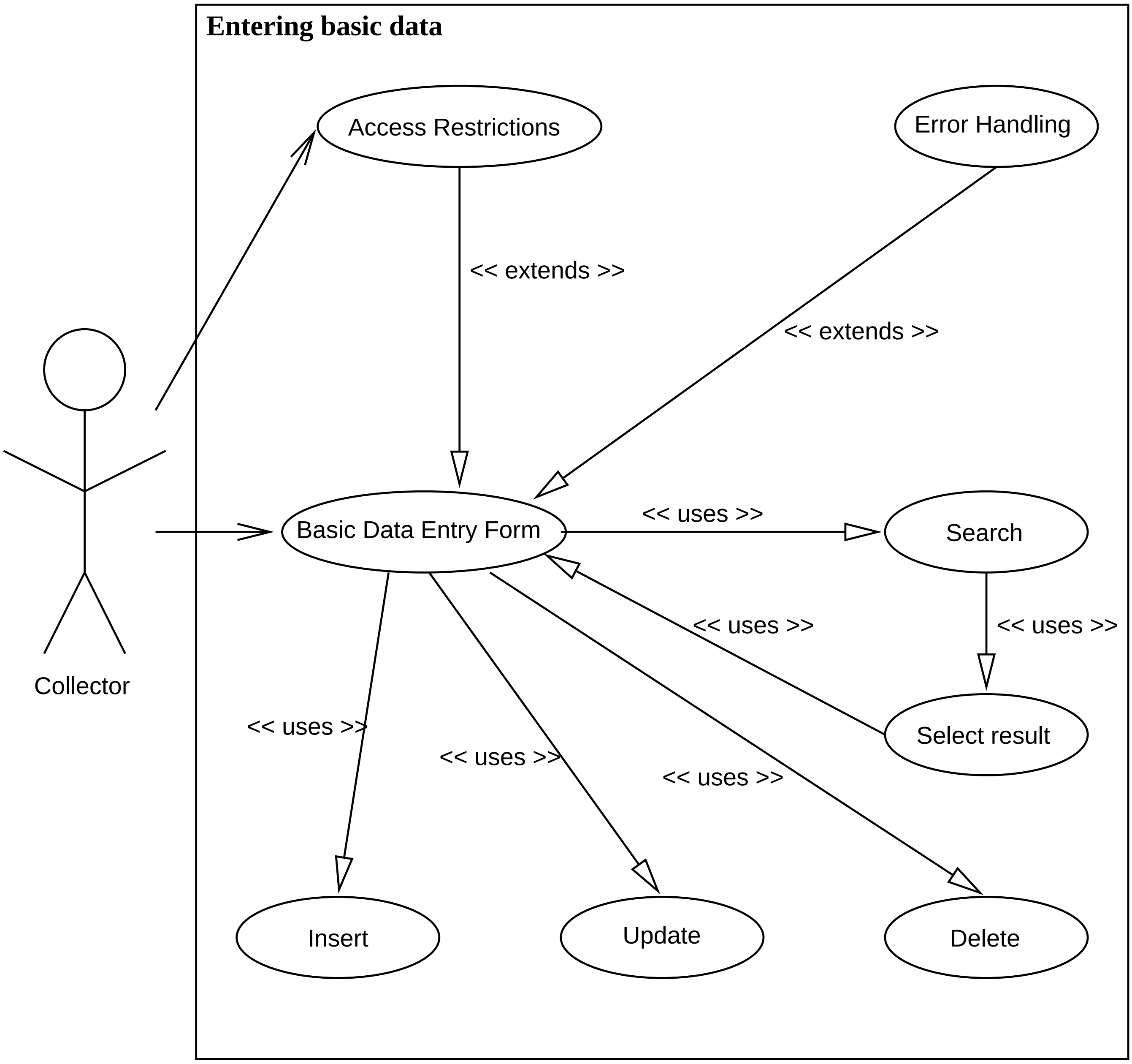
This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.3 Primary Use Case Diagrams

3.3.1 Entering Basic Data



3.4 Secondary Use Cases

3.4.1 Displaying Artefacts

Description

All data-items matching searched attributes will be displayed as links in a selectable list. After selection, the user gets a detailed view of the data-item. If the user has authenticated himself as a registered user/collector (login), more details are displayed.

Actors

- User
- Registerd user
- Collector
- Scientific collector

Priority

How important is this use case to the project?

Status

Facade

Preconditions

A list of conditions that must be true before the use case starts.

Postconditions

A list of conditions that must be true when the use case ends, no matter which scenario is executed.

Extension Points

If the use case has extension points, list them here.

"Used" Use Cases

If the use case uses other use cases, list them here.

Flow of Events

This could be a basic path and alternative paths, or the primary scenario.

1. *Preconditions*
2. *Main Flow*
3. *Subflows (if applicable)*
4. *Alternative Flows*

Activity Diagram

An activity diagram of the flow of events, or some significant or complex part of the flow of events.

User Interface

For systems that interface with people, include a description of the user interface, possibly using storyboards.

Secondary Scenarios

If alternatives and exceptions are not shown in the flow of events, scenarios should be listed here, and a brief description may be included.

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.4.2 Exhibition CD Creation

Description

If a collector has defined a virtual museum, he/she can burn a CD including all data needed for users to browse through the virtual museum via a web-browser offline from CD.

Actors

- User
- Collector

Priority

How important is this use case to the project?

Status

Facade

Preconditions

A list of conditions that must be true before the use case starts.

Postconditions

A list of conditions that must be true when the use case ends, no matter which scenario is executed.

Extension Points

If the use case has extension points, list them here.

"Used" Use Cases

If the use case uses other use cases, list them here.

Flow of Events

This could be a basic path and alternative paths, or the primary scenario.

1. *Preconditions*
2. *Main Flow*

3. *Subflows (if applicable)*
4. *Alternative Flows*

Activity Diagram

An activity diagram of the flow of events, or some significant or complex part of the flow of events.

User Interface

For systems that interface with people, include a description of the user interface, possibly using storyboards.

Secondary Scenarios

If alternatives and exceptions are not shown in the flow of events, scenarios should be listed here, and a brief description may be included.

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

3.4.3 Exhibition Catalogue Printing

Description

If a collector has defined a virtual museum, he/she can print a catalogue including all data needed for users to browse through the exhibition in printed form.

Actors

- User
- Collector

Priority

How important is this use case to the project?

Status

Facade

Preconditions

A list of conditions that must be true before the use case starts.

Postconditions

A list of conditions that must be true when the use case ends, no matter which scenario is executed.

Extension Points

If the use case has extension points, list them here.

"Used" Use Cases

If the use case uses other use cases, list them here.

Flow of Events

This could be a basic path and alternative paths, or the primary scenario.

1. *Preconditions*
2. *Main Flow*
3. *Subflows (if applicable)*
4. *Alternative Flows*

Activity Diagram

An activity diagram of the flow of events, or some significant or complex part of the flow of events.

User Interface

For systems that interface with people, include a description of the user interface, possibly using storyboards.

Secondary Scenarios

If alternatives and exceptions are not shown in the flow of events, scenarios should be listed here, and a brief description may be included.

Sequence Diagrams

If you do not have separate documents for scenarios, you might include sequence diagrams for them here.

Subordinate Use Cases

If the use case has subordinate use cases, show them here. Or you could include a use case diagram for the subordinate use case. Or both. Also tell what subsystem is responsible for this subordinate use case.

View of Participating Classes

A collaboration showing all the classes whose objects interact to implement this use case. You also can show interfaces to the use case here and which of the classes implement the interfaces.

Other Artifacts

This can include references to the subsystem the use case belongs to, an analysis model, adesign model, code, or test plans.

Other Requirements

This section is where you can put nonfunctional requirements affecting the use case.

Chapter 4

Project Proposal

This chapter estimates the costs and duration of the production and other work needed to provide the system.

4.1 UUCP (unadjusted use case points)

Actor		**12**
Data Colector (DC)	complex	3
Scientific Data Collector (SDC)	complex	3
Visitor (V)	complex	3
Database (DB)	simple	1
Administration (A)	average	2
DC		**30**
Data Entry	average	10
Data Entry Search and List	simple	5
Deletion of a Data Entry	simple	5
Update of a Data Entry	simple	5
Different Data Types	simple	5
SDC		**5**
Adding Documents	simple	5
V		**20**
Search for Data-Items	simple	5
Display of single Data-Items	average	10
Relations of Data-Items	simple	5
A		**20**
Costumer		<u>15</u>
Adding, Modifying, Deleting User	simple	5
Viewing User	simple	5
User Groups	simple	5
Access Restriction		<u>5</u>
Login	simple	5
DB		**10**
Structure	simple	5
Adding Relations between Data-Items	simple	5
VM		**25**
Room Administration	simple	5
Show-Case Administration	simple	5
Display and Navigation of Rooms and Show-cases	complex	15
Over All		**117**

Use Case Type	Descripton	Factor
simple	fewer then 5 analysis classes	5
average	5 to 10 analysis classes	10
complex	more than 10 analysis clases	15

4.1.1 TCF (technical complexity factor)

Factor number	Weight	Assigned Value	Extended Value	Reason
T1	2	4	8	
T2	1	3	3	limited by human input
T3	1	3	3	needs to be efficient because of VM
T4	1	1	1	easy processing
T5	1	1	1	
T6	0.5	0	0	no installing
T7	0.5	5	2.5	must be easy to use
T8	2	0	0	no
T9	1	3	3	sure
T10	1	5	5	must be
T11	1	3	3	no special security
T12	1	0	0	
T13	1	0	0	

Tfactor = 8+3+3+1+1+0+2.5+0+3+5+3+0+0 = 29.5
TCF = 0.6 + (0.01 * 29.5) = **0.895**

4.1.2 EF (environmental factor)

Factor number	Weight	Assigned Value	Extended Value	Reason
F1	1.5	5	7.5	most of team familiar
F2	0.5	5	2.5	most of team are programmers
F3	1	4	4	most of team are programmers
F4	0.5	4	2	
F5	1	4	4	good
F6	2	4	8	we espect no changes
F7	-1	0	0	
F8	-1	0	0	no

EFactor = 7.5 + 2.5 + 4 + 2 + 4 + 8 + 0 + 0 = 28
EF = 1.4 + (-0.03*28) =**0.56**
F1 - F6: Count (value $<$ 3) : 0
F7 - F8: Count (value $>$ 3) : 0

4.1.3 UCP (use case points)

UUCP	117
TCF	0,895
EF	0,56
UCP	58,64
Man Hour	**1173**

UCP = UUCP * TCF * EF
Man Hour = UCP * 20

4.1.4 Costs

Man Week :	32
Man Hours :	1280

One Man Hour costs 1000 ATS

1280 Man Hours a 1000 ATS = 1.280.000 ATS

4.2 Estimation based on XP

4.2.1 Programming

This section estimates the programming effort needed for all functional units. Estimates are measured in "ideal implementation days (IID)" or "ideal implementation weeks (IIW)", i.e. days which a programmer would need if he could work uninterrupted.

- **Basic Data Entry**

Login	0.5 IID
Entry Menu	0.5 IID
Input Form	2 IID
Save Input	1 IID
Update, Delete	1 IID
Search	1 IID
Search Result	1 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 2 IIW.

- **Search and Display**

Search Form	1 IID
Search Result List	1 IID
Display Artefact	1 IID
Display Relation	0.5 IID
Download Link	0.5 IID
Login	0.5 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 1 IIW.

- **Adding Documents**

Document Upload (adding internal link)	1 IID
Adding External Links	0.5 IID
Login	0.5 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 1 IIW.

- **Relations**

Add Relations	1 IID
View Relation List	1 IID
Update, Delete	1 IID
Login	0.5 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 1 IIW.

- **User Administration**

View Login List	1 IID
Edit Login	2 IID
Login	0.5 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 1 IIW.

- **Virtual Museum**

Room Administration	5 IID
Show-case Administration	4 IID
Room Display	1 IID
Show-case Display	1 IID
Navigation	1 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 3 IIW.

- **Exhibition CD**

XML Export	3 IID
XML to HTML Conversion	3 IID
ZIP-Download	1 IID
Login	0.5 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 2 IIW.

- **Exhibition Catalogue**

XML Export	3 IID
XML to PDF Conversion	3 IID
ZIP-Download	1 IID
Login	0.5 IID

Needed ideal implementation weeks

Rounding to full working weeks, this functional unit needs about 2 IIW.

4.2.2 Graphics

Design and layout for the following parts will be needed (estimation is very rough as requirements are still very unclear).

Search and Display Artefacts	1 IIW
Virtual Museum	3 IIW
Exhibition CD	2 IIW
Exhibition Catalogue	2 IIW

Needed ideal implementation weeks

Rounding to full working weeks, graphics need about 8 IIW.

4.2.3 System Administration

Setup of the servers (including hardware and software configuration) will roughly need 1 IIW.

4.2.4 Summarized Cost Report

This section summarizes the estimated costs of the project. We offer the following hourly rates:

Programming	ATS 1.000,–
Graphics	ATS 1.200,–
System Administration	ATS 800,–

Unit	Estimated Effort	Price (Effort * hourly rate)
Basic Data Entry	2 IIW	ATS 80.000,–
Search and Display	1 IIW	ATS 40.000,–
Search and Display Graphics	1 IIW	ATS 48.000,–
Adding Documents	1 IIW	ATS 40.000,–
Relations	1 IIW	ATS 40.000,–
User Administration	1 IIW	ATS 40.000,–
Virtual Museum	3 IIW	ATS 120.000,–
Virtual Museum Graphics	3 IIW	ATS 144.000,–
Exhibition CD	2 IIW	ATS 80.000,–
Exhibition CD Graphics	2 IIW	ATS 96.000,–
Exhibition Catalogue	2 IIW	ATS 80.000,–
Exhibition Catalogue Graphics	2 IIW	ATS 96.000,–
Server Setup	1 IIW	ATS 32.000,–

Production costs

The full production will cost ATS 936.000,–.

4.2.5 Hardware

The hardware will cost ATS 100.000,–.

4.2.6 Running Costs

Server-housing, Internet-connectivity and support will cost ATS 180.000,– per annum.

4.2.7 Overall Price based on XP estimation

The overall project price is **ATS 1,036.000,–.**

4.3 Summary

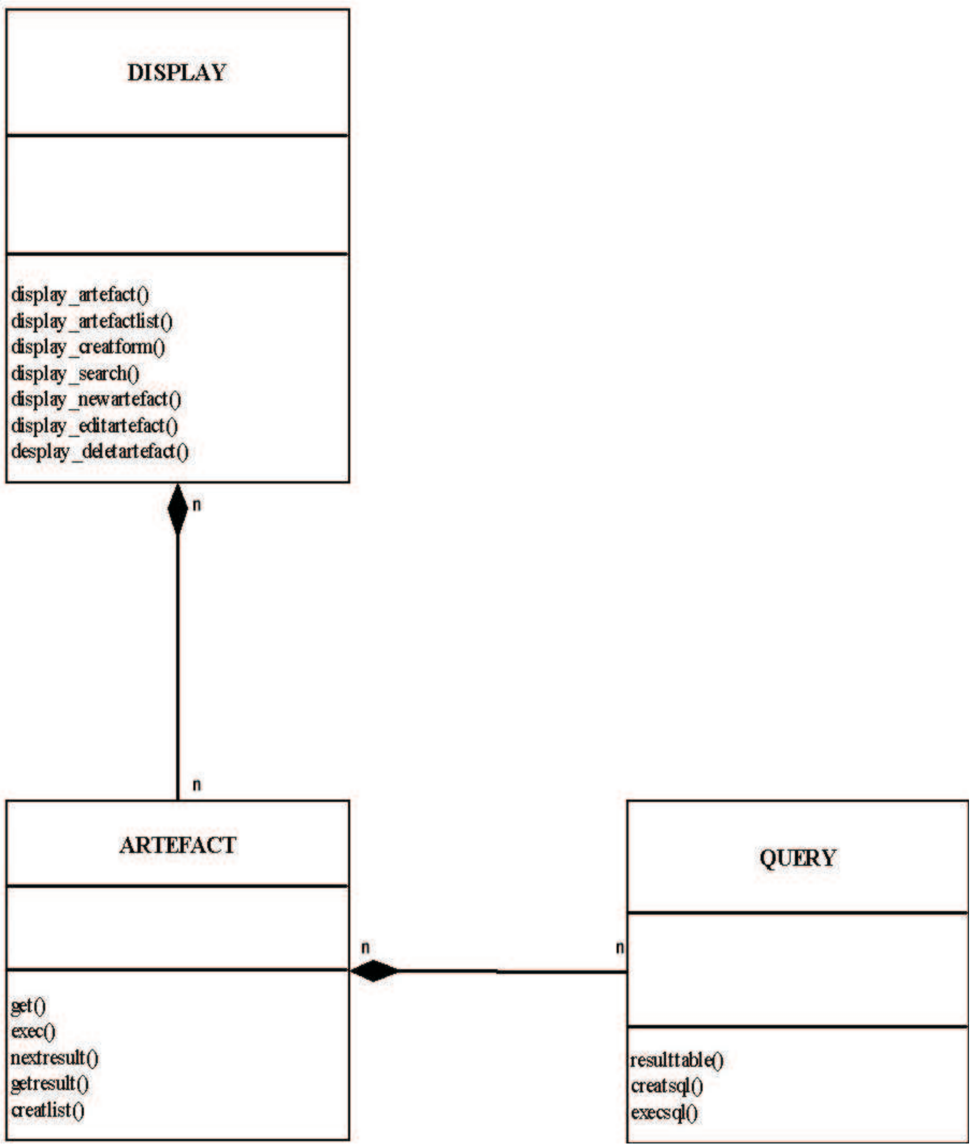
Based on two different types of estimations we now can offer the overall price based on the average of both methods.

The project price is ATS 1,158.000,– or EUR 84.155,–.

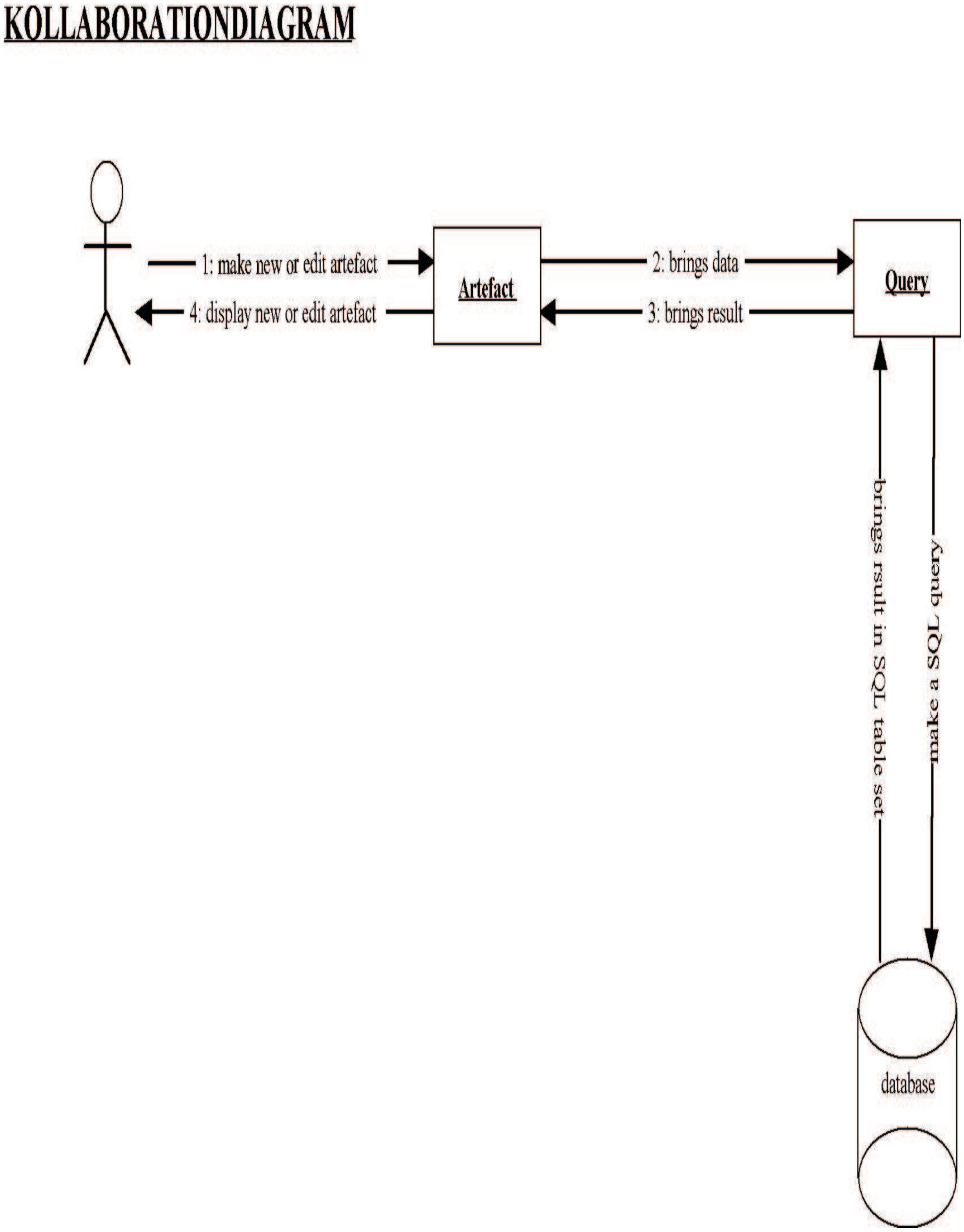
4.4 Diagrams

4.4.1 Classdiagram

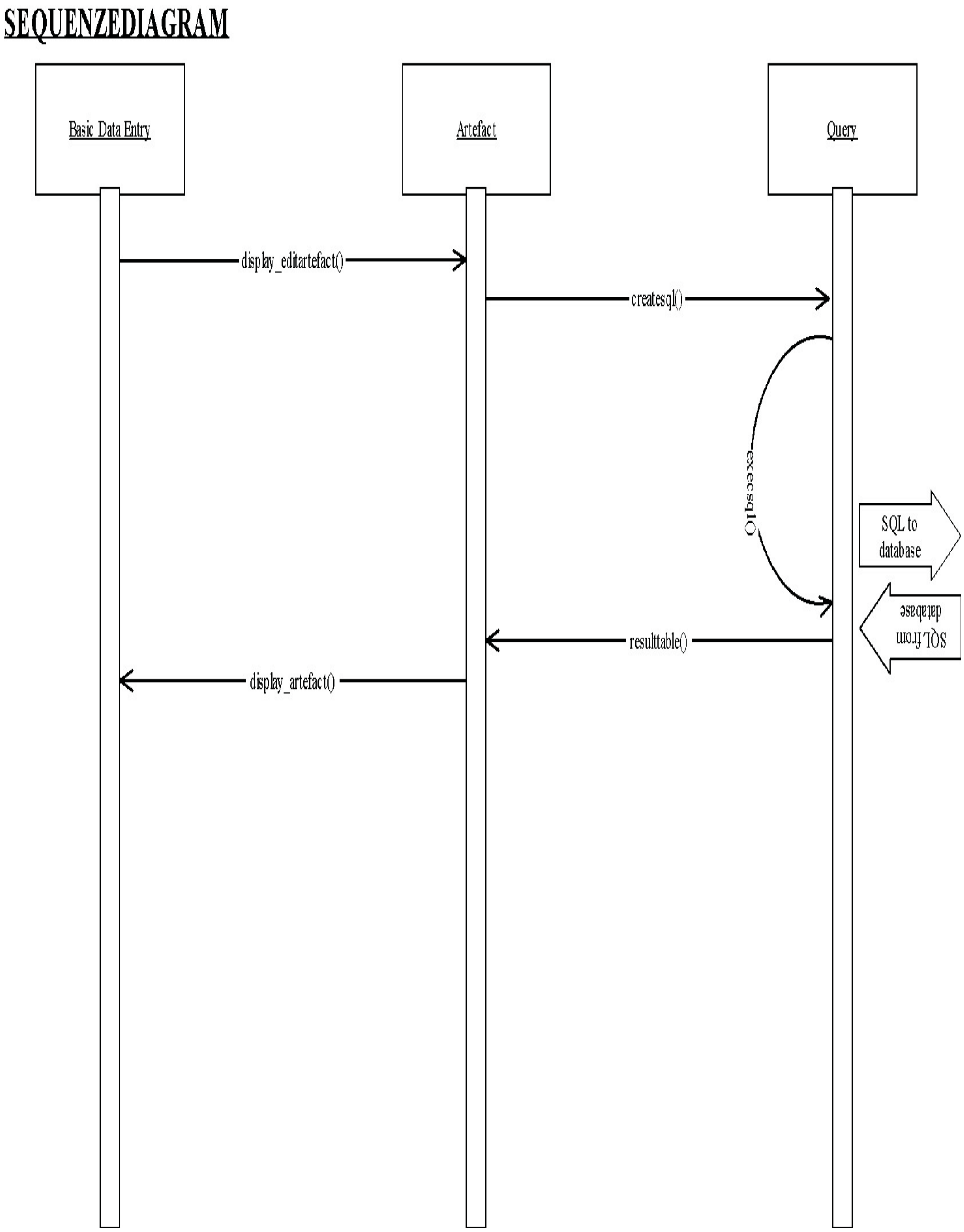
CLASSDIAGRAM



4.4.2 Kollaborationsdiagram

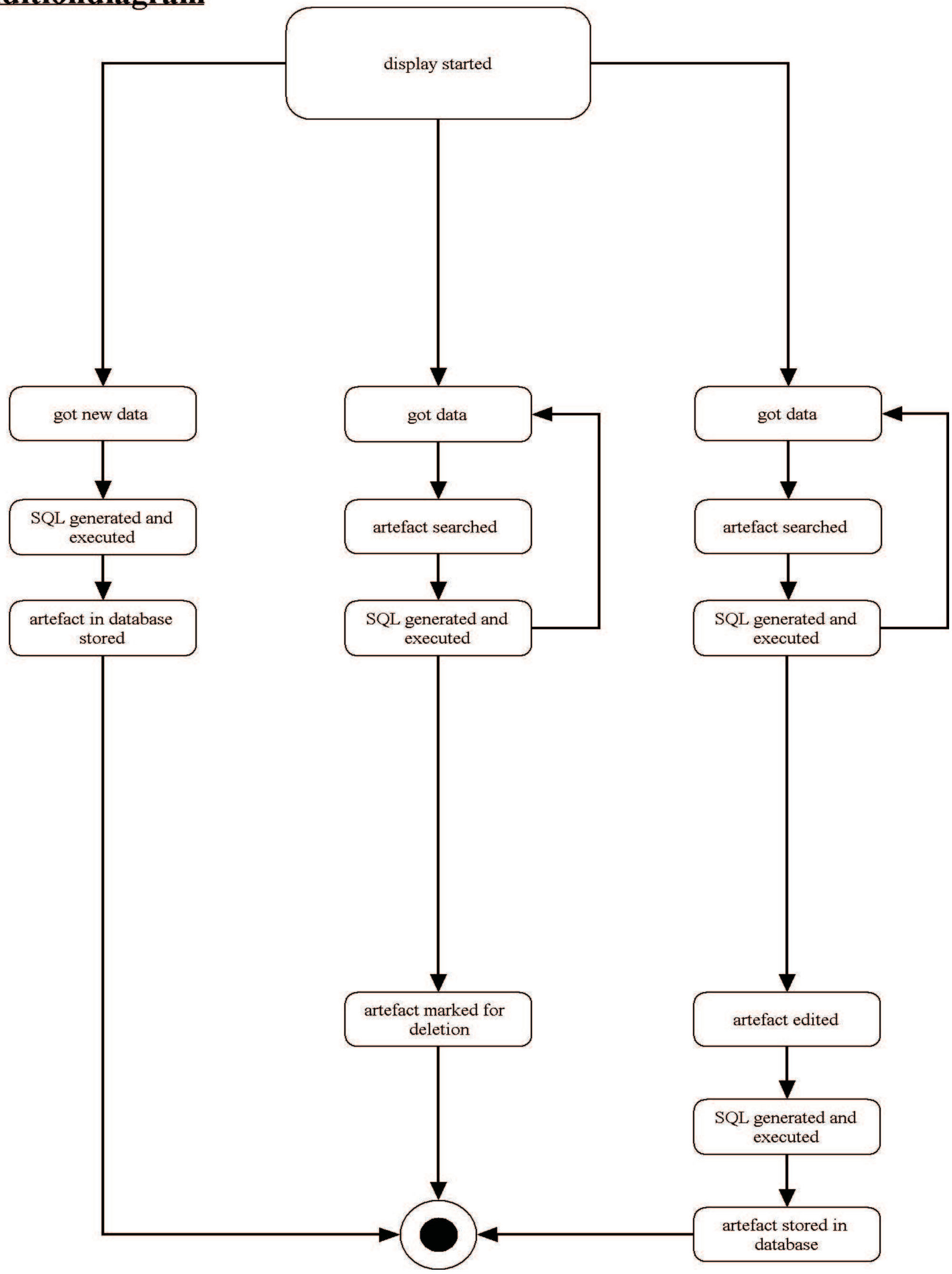


4.4.3 Sequenzdiagram



4.4.4 Conditiondiagram

Conditiondiagram



Appendix A

Protocolls

A.1 18 October 2000 13:15

Protokoll-Author: Gerhard Froner

Teilnehmer: Gerhard Froner, Roland Schwaiger

- **Administratives:**
  Es wurde darauf hingewiesen, daß auch die persönlichen Dokumente in TEX ausgearbeitet werden müssen.

- **Projektbeschreibung:**

  Funkionale Anforderungen:

  - **Datenerfassung: Zur Zeit werden die Rohdaten auf Karteikarten, die formularmäßig aufgebaut sind, erfaßt. Auf diesen werden z.B.: Fundort, Fundzeit, Abmessungen,... aufgenommen. Die gesammelten Karteikarten werden nach Inventarnummern sortiert und abgelegt. Es werden keine Daten elektronisch verarbeitet. Daraus folgt, daß kein Import von Altdaten erfolgen muß.**
    Die wissenschaflichen Daten setzen sich aus Dokumenten, Literaturverweisen, Bezüge auf andere Fundorte, Interpretationen,usw. zusammen.
    Ein Upload der Daten wird nicht gefordert.

    Unterschied zwischen Erfasser und wissenschaflichen Erfasser:

    Die Erfasser sind eine Untermenge der wissenschaftlichen Erfasser. Es sollten aber auch weitere Untermengen(z.B.: Studenten u.w.) überlegt werden. Daraus ergibt sich eine dynamische Steuerung der Benutzerrollen, d.h. verschiedene Benutzergruppen mit verschiedenen Rechten sollten z.B.: verschiedene Frontends haben. Auf Grund der verschiedenen Benutzergruppen ist es unerläßlich einen logon progress einzurichten.

- Verwaltung: Von unserer Seite sollte ein Datenbankhosting betrieben werden.
  Die zu erwartenden Datenmengen sind wie folgt:
  * 400 Bilder p.A.
  * 400 Zeichnungen p.A.
  * 10 Filme 10sec. p.A.

  Die im Jahresschnitt zu verwaltenden Personen:
  * 1 Erfasser
  * 1 wissenschaflicher Mitarbeiter
  * 1 studentischer Mitarbeiter

  Die Berechtigungen werden vom Auftraggeber angelegt und verwaltet. Es wurde aber ausdrücklich darauf hingewiesen, daß keine Personalverwaltungsdatenbank gefordert ist. Die Sicherheit bei Datenübertragungen kann niedrig sein.

Nichtfunktionale Anforderungen:

- Hardware: Unsere Hardware besteht aus zwei Maschinen, nämlich einen WEB-Server und einen Datenbankserver. Technische Daten: PIII 700Mhz, 512MB RAM und 50GB Platte. Dazu seht uns für Backuptätigkeiten ein Streamerlaufwerk zu Verfügung.

  Der Kunde besitzt eine ADSL - Anbindung und ein Netzwerk mit Standleitungen. Auf jedem Einzelplatzrechner ist ein WEB - Browser verfügbar. Mitarbeiter des Kunden arbeiten nicht mit einem Notebook und den dazugehörigen Kommunikationsmitteln.

- Software: Bei der Entscheidung über die nötige Software haben wir freie Hand. Beim Kunden darf keine weitere Software installiert werden.

Abscließend zum Them XML: XML dient ausschließlich als Schnittstelle zu Geräten und zu HTML.

A.2 24. October 2000, 17:00-19:00

We met for the handover of the project documents to the new project leader, and to discuss the next steps to serve our customers needs.

Person responsible for protocoll: Harald Spanring

Participating persons: Rene Elsässer, Gerhard Froner, Rudolf Götlich, Harald Spanring

Topics talked about: After the handover of the project documents, the following topics where discussed:

- **Artefact file cards: We decided to ask the customer for a carbon copy of the file cards which are being used by the customer to record archaeological data, so we can get a quick impression of what attributes will be needed for the online data entry in ARIS.**
- **Document upload: It is still not clear if the customer wants to upload documents via the system, though recommended from our point of view to reduce support by our operators when the data collectors want to add documents to an entered artefact.**
- **Data quantity: Though the rough number of pictures and movies is specified, it's still not defined, how many artefacts will be entered into the system.**
- **User groups: The different user privileges will be implemented through user groups. We decided to suggest a static number of possible groups with specialiced priviliges which can not be altered online, unless the customer provides other requirements during project analysis.**
- **Hardware: The hardware specifications have been acknowledged as described in the latest customer-meeting-protocol.**
- **Software: As (server-)software decisions are our business, we decided to inform the customer as soon as we know to enable any (though unlikely) veto from the customer early.**
- **Browser: We recommend to limit the number of web-browsers which will be supported by our system and will discuss this issue with the customer (This might conflict with the XML-requirement).**
- **XML: It is still not clear if XML should be used with the user frontend, and if, if the customer is aware of the restrictions which apply.**

New assignments: Further product analysis and risc analysis.

Open questions: As mentioned above.

Next meeting: The next meeting will proceed on Oct. 30. 2000, 17:00.

A.3 30. October 2000, 17:00-19:00

Person responsible for protocoll: Harald Spanring

Participating persons: Naami Abderrahim, Rene Elsässer, Gerhard Froner, Rudolf Götlich, Harald Spanring

The the following topics where discussed:

Document language: We decided to write the group-document in english, individual solution documents and protocolls may be writen in german or english.

Functional requirements: We discussed our current corporate view of the requirements of the system to be build, and verified that the resulting group-document will represent all our ideas.

Risk analysis: We discussed common risks of software projects, based on the material provided by Roland Schwaiger.

Open questions: Most open questions are now formulated as asumptions within the functional requirements, so the customer can see our ideas and veto anything which we misunderstood.

Other open questions are:

- A carbon copy of the artefact card-files would still be a great help.
- We need definitions of image-formats, video-formats, etc. and if videos should be downloaded or played within the browser.
- The term 'document upload' has been created to differentiate between data-import (which is not needed) and document-upload, which should happen via the web-browser.
- It might be necessary to administer uploaded documents via a web-frontend, we have to ask the customer about it.

A.4 31. October 2000, 14:15-14:45

Person responsible for protocoll: Harald Spanring

Participating persons: Roland Schwaiger, Harald Spanring

The the following topics where discussed:

Settlement attributes: The customer enters the following attributes for settlements:

- Description (text)
- Location (text)

- GPS (coordinates)
- Size (m^2):
- Images (1-5 GIF or JPG)
- ⇒ tombs (hyperlink)
- ⇒ artefacts (hyperlink)

Tomb attributes: The customer enters the following attributes for tombs:

- Description (text)
- Location (text)
- GPS (coordinates)
- eventually hyperlinks to a GIS
- Allotment (number)
- Images (1-5 GIF or JPG)
- ⇒ artefacts (hyperlink)
- ⇒ settlement (hyperlink)

Artefact attributes: The customer enters the following attributes for artefacts:

- Inventory # (number)
- Name (text)
- Dimension (dim)
- Category (text)
- Description (text)
- Images (1-5 GIF or JPG)
- eventually audio data
- ⇒ tomb (hyperlink)
- ⇒ settlement (hyperlink)

Data quantity: The customer expects about 10 artefacts being entered per working day, tombs or settlements will not be entered very often, maximum 1 per day.

User view restrictions: The anonymous user will never see the detailed description texts of artefacts, and individual artefacts may be configured by the collectors to be invisible to the user.

File formats: For images, the formats GIF and JPG will be used. For audio, the formats RealAudio and WAV will be used. For video, the formats RealVideo, MPG2 and AVI will be used. Scientific documents must be HTML or PDF files.

Media players: The user has to install any needed media-player by himself to view audio or video. For RealAudio/RealVideo, a browser- plugin should be offered by our pages.

Document upload: A document upload is acceptable for the customer, but direct entering of HTML-document would suffice, too. When entering HTML, the collector could add hyperlinks directly.

XML: No XML-browsers are needed, Output-formats of the webserver will be HTML, WML or PDF.

Browsers: We should support almost any browser, so no tricks are allowed which make pages unreadable for certain browsers, HTML 3.2 is ok. Even plain-text-browsers like lynx should be able to browse ARIS. Flash, javascript, etc. is not allowed.

Database: It is still not clear which database-engine we will use, as it is still unknown which data-types the database has to support. Current candidates: mysql, postgres, oracle. Our implementation should provide a implementation-layer which allows late changing of the used database-engine.

Homepage: The customer already has an own homepage, which will contain a link pointing to ARIS.

A.5 6. November 2000, 15:00-17:00

Person responsible for protocoll: Rudolf Göttlich

Participating persons: Naami Abderrahim, Rene Elsässer, Gerhard Froner, Rudolf Götlich, Harald Spanring

The the following topics where discussed: handover of the projekt document, report of the last meeting with Roland Schwaiger, the new task.

Task 3 Use Case Diagram: In working together we developed the solution of task 3.
Some points using the Use Case Diagram difficult for use. The group document was developed by all, the single documents developed after the group document in groups by two people.

A.6 8. November 2000, 14:00-14:45

Person responsible for protocoll: Rudolf Göttlich

Participating persons: Roland Schwaiger, Göttlich

The the following topics where discussed: Use Case Diagram, Use Case Description

Use Cases: Use Cases declares the functionality of the program, every point should not be declared technical, it should be declared in common words.

We have to revise the cases, especially the primary case.

Points to do :

Primary Cases: must have the priority I, primary cases are the same like in the Use Case Diagram.

3.2.4. Searching and Displaying Artefacts: Part in Searching and Displaying

3.2.5. Virtual Museum Walktrough: is a secondary case. Primary case is the prepare of the Walktrough

3.2.6. Exhibition CD Creation: is a secondary use case.

3.2.7. Exhibition Catalogue Printing: is a secondary case.

3.2.8. Administration: Part into system-administration and customer-administratio

Primary Cases: we have to control every primary case.

Bibliography: should be extended

Appendix B

Glossary

ASP: Application Service Provider. A company which provides applications on a central internet-server instead of installing software on the client's machines, commonly implemented via a web-server accessed by a web-browser, and often using a database-server as backend for data-storage.

HTML: HyperText Markup Language. Format of hypertext-documents used with the WWW.

HTTP: HyperText Transfer Protocoll. Protocoll used for communication between a web-browser and a web-server.

hypertext: An extension to multimedia-documents to enable navigation to related documents via simple mouse-clicks.

ISP: Internet Service Provider. A company that provides access to the internet.

PDF: Portable Document Format. A document format from Acrobat for portable documents, enabling easy printing.

web-browser: A client programm to access hypertext-documents from a web-server or a local filesystem. Popular web-browsers are the Netscape Navigator from Netscape and the Internet Explorer from Microsoft.

web-server: A server program providing hypertext-documents to the internet via HTTP.

WWW: The World-Wide-Web. An internet-based hypertext-system with multimedia-features.

XML: The eXtensible Markup Language. The so-called successor of HTML. Allows dynamic data-representation with ASCII-formated text-files using attribute-tags to markup data. In contrast to HTML, XML allows the definition of custom tags, thus enabling more flexible data-representation. XML-files can be converted to many different formats as HTML, PDF, ...