PS Software-Engineering

SE-001-2000 9th November 2000

ARIS

Report

AUTHOR

@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

, rex@themail.com, gerhard.froner@sbg.ac.at,

ARIS

Author
Department of Computer Science
University of Salzburg
A–5020 Salzburg, Austria
@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.

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.

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: 1

Rationale: Any important data-type must be supported.

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 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: Videos might be needed for data-display.

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).

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.

1.8 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, ...

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 Resonse 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 Finacial 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.

Appendix A

Protocolls

A.1 18 October 2000

Protokoll-Author: Author

Teilnehmer: Teilnehmer

- **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

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: Author

Participating persons: Participants

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.

A.3 30. October 2000

Person responsible for protocoll: Author

Participating persons: Participants

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.