



UMIT

private universität für gesundheitswissenschaften, medizinische informatik und technik
university for health sciences, medical informatics and technology

Towards an interactive, spatially standardized, gene expression database of the embryonic heart

Bakkalaureatsarbeit von: Schafferer Simon und Weidenholzer Benjamin

Im Rahmen des Studiums *Biomedizinische Informatik* an der Privaten Universität für
Gesundheitswissenschaften, Medizinische Informatik und Technik

Betreuer:

Univ.-Prof. Dr. Ammenwerth Elske

Bouke A. de Boer, MSc.

Jan M. Ruijter, PhD

Frans P. Voorbraak, PhD

Betreuerbestätigung

Hiermit bestätige ich, die vorliegende Abschlussarbeit betreut zu haben, und ich befürworte damit die Abgabe der von mir insgesamt positiv benoteten Arbeit.

.....

Datum und Unterschrift des Betreuers

.....

(Name des Betreuers in Blockbuchstaben)

Annahme durch das Studienmanagement

am:

von:

Abstract

The research of the Heart Failure Research Center (HFRC), a department of the Academically Medical Center in Amsterdam (AMC) focuses on genetic factors causing Congenital Heart Defects (CHD). CHD are structural problems arising from abnormal formation of the heart or the major blood vessels and the most common cause for infant death.

To find genes involved in the development, embryos are stained and cut into slices, but so the 3D anatomical context is lost. To recover this 3D context, at the HFRC a program called TRACTS (TRace the Anatomical Context of Tissue Sections) is developed, which automatically fits the stained slices with the gene expression in a 3D reference model and returns a visualization including the 3D model with the fitted slice. To make such an improved atlas accessible through the Internet would open new perspectives to the research on the embryonic heart.

So the main goal of this thesis is to develop a web interface that makes TRACTS available through the Internet, to give users the location of the submitted sections. Furthermore TRACTS should be used to gather gene expression information of heart sections and store them in a database to get an improved atlas of heart development.

Interviews with later users were performed to identify the requirements. Thereafter the database scheme and the website structure were developed. The required software to implement the web interface was found via internet analysis.

An almost complete and fully working web application was implemented, which offers the possibility to have a 3D localization and visualization of gene expression patterns of the embryonic heart accessible through the Internet. Furthermore this prototype enables to collect data in an extendable database. This is a big advantage for the research, because till now no such web interface was available.

The researchers of the AMC are now able to develop a leading gene expression database in the field of embryonic heart research.

Zusammenfassung

Die Forschung am „Heart Failure Research Center“ (HFRC), ein Institut am „Academically Medical Center“ in Amsterdam (AMC) befasst sich mit genetischen Faktoren, welche angeborene Herzfehler (CHD) verursachen. Dies sind strukturelle Probleme, die bei abnormaler Entwicklung des Herzens bzw. der Hauptblutgefäße entstehen und die häufigste Ursache für Totgeburten darstellen.

Die Forschung an Genen, die an der Entwicklung des embryonalen Herzens beteiligt sind, erfordert das Anfertigen von Schnittbildern, wodurch jedoch der dreidimensionale anatomische Kontext verloren geht. Um die Position eines Schnittbildes in seiner ursprünglichen anatomischen Struktur wieder zu finden, wird am HFRC das Programm TRACTS (TRace the Anatomical Context of Tissue Sections) entwickelt. Dazu werden die Schnittbilder in ein Referenzmodell eingepasst, das anschließend visualisiert wird. Einen solchen dreidimensionalen anatomischen Atlas über das Internet zugänglich zu machen, würde neue Perspektiven für die embryonale Herzforschung eröffnen.

Das Ziel der Arbeit ist es, ein Web-Interface zu erstellen, welches TRACTS über das Internet verfügbar macht. Dies ermöglicht dem Benutzer die hochgeladenen Schnittbilder im dreidimensionalen Raum zu betrachten. Ferner sollte TRACTS dazu verwendet werden, genexpressionsspezifische Daten der Schnittbilder des Herzens zu sammeln. Diese werden in einer Datenbank gespeichert, um einen verbesserten Atlas embryonaler Herzentwicklung zu erhalten.

Um die Anforderungen zu ermitteln, wurden Interviews mit zukünftigen Benutzern durchgeführt. Darauf basierend wurde das Datenbankschema und die Struktur der Website erstellt. Die benötigte Software zur Realisierung des Projektes wurde bei einer umfassenden Internet-Recherche ermittelt.

Es konnte eine vollfunktionsfähige und fast vollständige Webapplikation erstellt werden, die eine Lokalisierung und Visualisierung in einem dreidimensionalen Modell von “gene expression patterns” des embryonalen Herzens über das Internet erlaubt. Darüber hinaus ermöglicht dieser Prototyp das Speichern von Daten in einer jederzeit erweiterbaren Datenbank. Dies ist der erste derartige Atlas und stellt somit einen wesentlichen Fortschritt für die Forschung dar.

Die Forscher am AMC sind nun in der Lage, eine wegweisende Genexpressionsdatenbank für die Erforschung von Embryoherzen bereit zu stellen.

Structure

The project was a collaboration of three students, Simon Schafferer, Benjamin Weidenholzer and Klemens Woertz. It could be split in three independent parts:

1. Web interface by Simon Schafferer
2. Database by Benjamin Weidneholzer
3. 3D Visualization by Klemens Woertz

This thesis contains only part one and two. The third one will be published in a separate thesis describing the 3D visualization, the interface between TRACTS and the web application in detail.

Part I commonly written by Benjamin Weidenholzer and Simon Schafferer

Part II written by Simon Schafferer

Part III written by Benjamin Weidenholzer

Part IV commonly written by Benjamin Weidenholzer and Simon Schafferer

CONTENT

Abstract	3
I. Common part	11
1. Introduction	11
1.1. Congenital heart defects and genes.....	11
1.2. Data production and the loss of 3D context.....	13
1.3. Recovering 3D context	14
1.4. Project specification	16
2. Basics and state of technology	17
2.1. Biological background of the embryonic heart.....	17
2.2. From 3D computer reconstructions to TRACTS.....	18
2.3. TRACTS.....	18
3. Analysis of gene expression databases.....	20
3.1. Analysis of GenePaint.....	21
3.2. Analysis of the Edinburgh Mouse Atlas Project (EMAP)	26
3.3. Comparison of GenePaint and EMAP	31
4. Methods.....	32
4.1. Requirement analysis.....	32
5. Results.....	34
5.1. Summary of the interviews	34
5.2. Use case analysis	42
5.3. Requirements specification	44
II. Web interface	46
1. Basics and state of technology	46
1.1. Motivation for a web interface	46
1.2. WWW and HTML	46
1.3. Cascading style sheets	47
1.4. Web server.....	47
1.5. Client-server architecture	48
1.6. Multiple tier architecture	48

1.7.	Dynamic web pages: client-side scripting.....	49
1.8.	Dynamic web pages: server-side scripting	49
2.	Methods.....	53
2.1.	Diagram creation programs.....	53
2.2.	Analysis of GenePaint and EMAP	53
2.3.	Requirements analysis	54
2.4.	Navigation diagram	55
2.5.	Form based authentication.....	56
2.6.	MD5 encryption	57
2.7.	Software architecture	58
3.	Results.....	59
3.1.	Architecture of the web application	59
3.2.	Navigation diagram	61
3.3.	Security	62
3.4.	Creation of an embryo.....	64
3.5.	Adding of an image	65
3.6.	Upload of an image	65
3.7.	Visualization	67
4.	Discussion	68
4.1.	The accomplished goals.....	68
4.2.	Architecture	68
4.3.	Software specification	69
4.4.	Future navigation frame	69
4.5.	Generalization of the project	70
III.	Database	71
1.	State of technology	71
1.1.	Principles of a database	71
1.2.	Architecture of a DBS.....	72
1.3.	Data models	72
1.4.	Integrating a relational database into an object-based application.....	73

2.	Methods.....	76
2.1.	Entity-relationship-model.....	76
2.2.	Transformation of the E/R-model into a relational database scheme.....	77
3.	Results.....	78
3.1.	E/R-Model	78
3.2.	Entities	79
4.	Software specification.....	83
4.1.	DBMS: MySQL.....	83
4.2.	ORM in Hibernate	83
4.3.	The Hibernate Query Language (HQL)	84
4.4.	Most important Hibernate classes	84
4.5.	Storage of images	88
5.	Discussion	89
5.1.	The prototype	89
5.2.	Future perspectives and discussion	90
IV.	Common part.....	91
1.	Common Discussion.....	91
1.1.	Significance of the project	91
1.2.	Generalization of the project	91
1.3.	Requirement analysis interviews.....	91
1.4.	Future work of the AMC	91
V.	Appendix.....	92
1.	Data Dictionary	92
2.	Questionnaire	97
2.1.	General questions	97
2.2.	Questions concerning genePaint and EMAP	97
2.3.	Questions concerning the web interface	97
VI.	List of figures	99
VII.	List of tables.....	101
VIII.	References	102

Eidesstattliche Erklärung

Hiermit versichere ich, Simon Schafferer, geboren am 26.06.1985 in Innsbruck, dass die vorliegende Bakkalaureatsarbeit von mir selbständig verfasst wurde. Zur Erstellung wurden von mir keine anderen, als die angegebenen Quellen und Hilfsmittel verwendet.

Hall in Tirol, 10. Oktober 2008

(Simon Schafferer)

Eidesstattliche Erklärung

Hiermit versichere ich, Benjamin Weidenholzer, geboren am 17.02.1985 in Salzburg, dass die vorliegende Bakkalaureatsarbeit von mir selbständig verfasst wurde. Zur Erstellung wurden von mir keine anderen, als die angegebenen Quellen und Hilfsmittel verwendet.

Hall in Tirol, 10. Oktober 2008

(Benjamin Weidenholzer)

I. COMMON PART

1. INTRODUCTION

Congenital heart defects and genes

Congenital heart defects (CHD), are structural problems arising from abnormal formation of the heart or the major blood vessels.

In 2002, in the United States, the prevalence of congenital cardiovascular disease was estimated to range from 650.000 to 1.3 million. Nine defects per 1.000 live births, or 36.000 infants, are expected to die each year. This makes congenital cardiovascular disease to the most common cause of infant death from birth defects: more than 30 percent of infants who die from a birth defect have a heart defect [2].

These data make obvious, that CHD constitute a serious medical problem. Therefore researchers want to reduce the prevalence of CHD by searching for the factors involved. There are some non-genetic factors for congenital defects, like rubella infection during pregnancy.

However, research of the HFRC (Heart Failure Research Center), a department of the AMC (Academically Medical Center, Amsterdam) focuses on genetic factors causing CHD. So there is more information on genetic factors contributing to CHD required.

There are several possibilities to get the required insight to understand the genetic causes of CHD:

1. Genetic linkage analysis

The first approach is the so-called “genetic linkage analysis”. Researchers study living individuals with congenital heart diseases in consideration of their pedigree. They evaluate statistically and describe the involved genes.

2. Looking at the embryonic heart to find genes involved in the development

Another prevalent method is studying specifically the formation of a developing embryonic heart resulting in seeing which genes are involved and how they take part in the cardiac development. This method concentrates on the functionality of genes.

3. Treating genes

Mutating, “knocking in” or “knocking out” a gene is another commonly used method. Mutation will lead to a different gene product while “knocking in” results in an over-expression of a gene. “Knocking out” ends up in a reduction of the normal amount of a gene product. After each of these modifications, researchers look at the effect on the development of the heart.

All three methods are applied by the HFRC. The current project focuses on the second method.

Data production and the loss of 3D context

For reasons being explained later, embryos are cut into sections, but this inevitably causes the loss of the 3D context.

Gene activity can be visualized in sections by labeling the gene products (mRNA and proteins). There are two staining procedures to make them visible:

1. In situ hybridization

With this method researchers detect mRNA with a Dioxigenin (DIG)-labeled-RNA probe. After this, a DIG-antibody is coupled to the DIG-labeled mRNA probe. Since this DIG antibody is conjugated to alkaline phosphatase it can be visualized with an enzyme-reaction [7].

2. Immuno histo chemistry

Indirect immunohistochemistry is based on the use of a primary antibody against a specific protein and a secondary antibody coupled to a fluorochrome, enzyme or biotin. The secondary antibody specifically recognizes the primary antibody and permits its localization using either illumination with a unique wavelength or an enzymatic reaction [6].

Hence staining embryonic tissue with specific markers is a common approach used to study genes involved in cardiac development. As the result of this staining a so-called gene-expression-pattern is visible in the sections.

The staining process requires embryonic tissue, which can only be obtained by the use of model organisms which have a similar process in cardiac development to humans. Therefore embryonic research of the HFRC focuses on mice, chicken or rats.

Beside sectioning embryos, there are no other appropriate methods available. Gene specific staining can also be applied to whole mount embryos, which can be visualized in 3D with OPT (Optical Projection Tomography) [9]. However, whole-mount-staining is limited to early developmental stages due to penetration problems with the increasing size of the embryos. Appropriate gene-expression patterns can be achieved until embryonic day (ED) 11.5 for mice and till Hamburger Hamilton (HH) stage 20 for chicken. Limited penetration results in unreliable gene-expression patterns for larger embryos [8].

On the other hand the required microscopic level of detail cannot be obtained by other imaging methods used in medicine like e.g. MRI (Magnetic Resonance Imaging) or PET (Positron Emission Tomography) [1]. Furthermore, these methods

do not provide the specific labeling of gene products. So, in fact the output of these methods does not contain any gene expression information.

Indeed, sectioning the embryo provides a detailed insight into gene expression patterns, but the anatomical context of the embryonic heart is lost. To understand the extensive process of cardiac development with its complex interactions between different genes, an anatomical 3D-context is needed. Therefore a 3D reconstruction consisting of stained embryonic sections for every developmental stage has to be generated.

Recovering 3D context

A 3D reconstruction can only be generated under the condition that all sections are stained and available. To meet this requirement, the department of Anatomy & Embryology and the department of Medical Informatics at the AMC have been developing a 3D reference model as basis for an application called “TRACTS” (TRace the Anatomical Context of Tissue Sections), which provides an anatomical 3D context for small sample sets.

This 3D reference model is based on a complete series of EFIC-generated (Episcopic Fluorescence Image Capturing) sections of an embryonic mouse heart. The advantage of EFIC is the production of an image series with no section deformation which is perfectly aligned [3]. Based on tens of thousands of calculated sections through this reference model, TRACTS is able to automatically fit 2D sections of an embryonic mouse heart into their 3D context. It fits the stained input section to the best fitting virtual section of the reference model. As a result a visualization of the best fitting reference section, mapped onto the 3D reconstruction and the stained 2D input section are shown. A finished MATLAB-prototype of TRACTS is already available [5].

TRACTS has been developed with two goals in mind:

1. Supply researchers with an idea of the localization of an expressed gene in an anatomical context.
2. In future TRACTS should be used to gather gene expression data of individual sections that are fitted in the 3D reference model to get a gene expression atlas of a developing mouse heart. This would be a benefit, because existing gene-expression atlases do not describe the 3D context of an embryonic mouse heart in sufficient detail.

To make such an improved atlas accessible over the Internet would open new perspectives to research on the embryonic heart.

Project specification

The challenge of this project is to develop a web interface that makes TRACTS accessible over the Internet to give users the best location of their individual sections. TRACTS should be used to gather gene expression information of heart sections into a database to get an improved atlas of heart development.

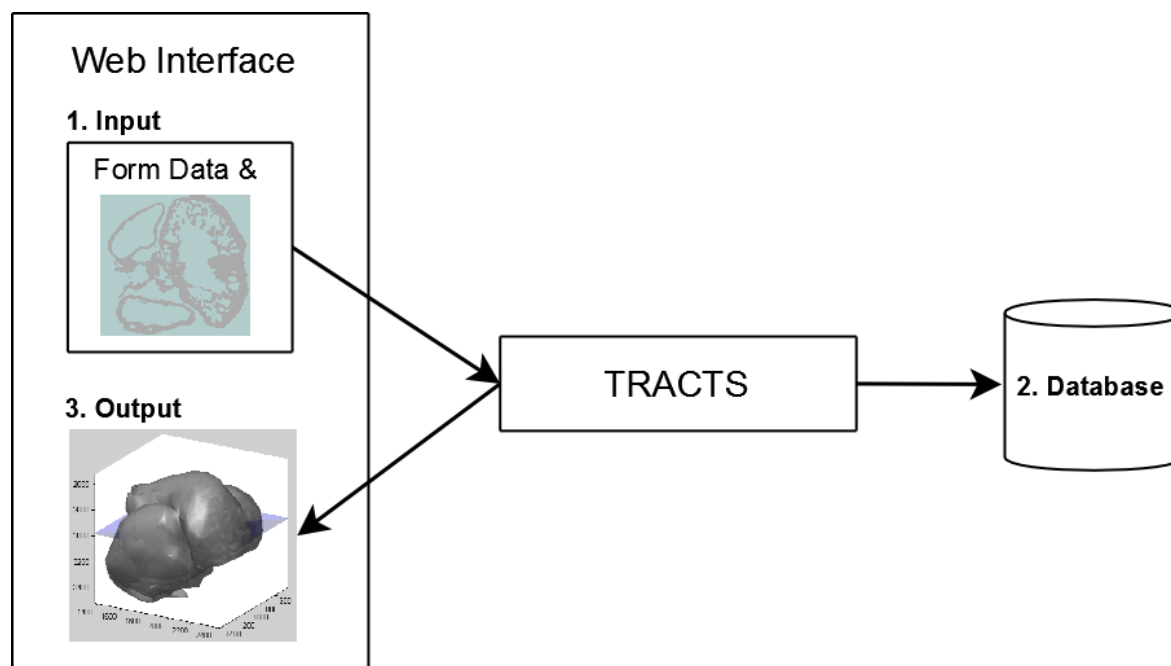


Fig. 1 Simplified overview of the project

As depicted in figure 1, it is possible to split the project into three separate parts:

1. An online input **interface** with the required input fields and calling TRACTS.
2. A **database** to store the collected data.
3. An appropriate **visualization** of the output results of TRACTS.

In fact the main challenge of the project is to incorporate TRACTS into a web interface.

2. BASICS AND STATE OF TECHNOLOGY

Biological background of the embryonic heart

The heart of mature birds and mammals consists of 4 chambers, which are divided into two parallel pairs: the right and left atrium are exclusively connected to the right and the left ventricle. The right side of the heart serves the pulmonary circulation whereas the left side of the heart handles the systematic circulation. It is important to consider the enormous morphological changes of an embryonic heart. During the embryonic development, while the heart has to function completely, a dual circuited four-chambered heart is formed out of a single-circuited heart tube (figure 2). For instance a mouse heart grows from 250 μm at ED 8.5 to 1.5 mm at ED 15.5. Therefore it is not easy to analyze and understand cross sections through a developing heart [5].

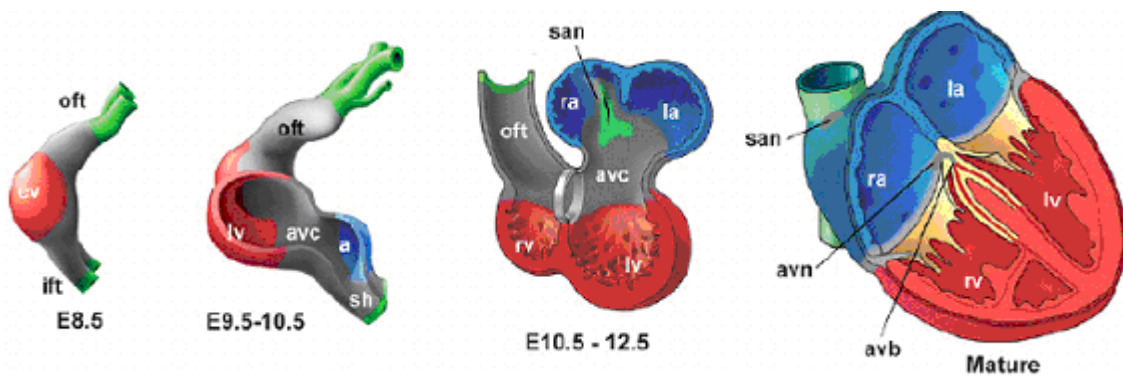


Fig. 2 Schematic illustration of the development of an embryonic heart of a higher vertebrate. a = atrium, v = ventricle, l = left, r = right. The other labels are not relevant for this paper.

From 3D computer reconstructions to TRACTS

In research of the cardiac development it is unavoidable to use serial sectioned biological material for 3D computer reconstructions, because of the required level of detail and the limited penetration of staining agents into tissues [1]. Often only a limited number of sections are stained by the researchers for specific proteins or mRNA. Furthermore, in many cases the sections are of unknown orientation and not exactly timed. So there is a lack of information on their anatomical context, which makes an interpretation of sections difficult. To provide information of the anatomical context an application called TRACTS has been developed by the departments of Anatomy & Embryology and Medical Informatics of the AMC. TRACTS fits the submitted 2D section in a 3D reference model of the heart and gives back its coordinates and a 3D visualization. This offers the possibility of viewing sections located in the heart [5].

TRACTS

TRACTS consists of a database containing tens of thousands of virtual cross sections of the reference model, which are made by taking a step size of one voxel over the three central axes of the sections. At every position 64 cross-sections are computed on the basis of tilting angle and tilting direction. Positional and size information are stored in a lookup table.

The basic version of TRACTS (the extended version is being used right now) is based on a pixel-based brute force approach, which compares the input 2D sections to the database of virtual 2D cross sections. Only inputs similar in size, about 20% deviation with the cross sections of the reference model are computed. TRACTS compares input sections with cross sections of the reference model based on the Euclidean distance discussed more detailed in [5]. To decrease the computation time and increase the positive fit results of TRACTS some features have been added which are not explained further [5].



Fig. 3 From left to right: An image of a histological section, a resized thresholded binary image, its contour, and its distance transformed image. a = atrium, v = ventricle, l = left, r = right. Note that at this stage the systematic and pulmonary circulation are not yet separated.

A pre-processing, described in [5], has to be carried out before the input section is compared to the reference model. The resulting images are compared with the reference model and the lowest distance indicates the best match.

However the computation time is approximately one minute per section, which is suitable for the use of the program. Currently the program features only ED11.5 mouse embryos, but the program will be extended to have the ability to fit sections of other age and species, like chicken. An improvement of the distance computation and a reevaluation of the chosen resolution of the reference database will be done in future [5].

3. ANALYSIS OF GENE EXPRESSION DATABASES

The purpose of this chapter is to analyze existing projects in order to get an overview on implementation of issues to point out advantages or disadvantages. This is necessary in order to prevent mistakes and also to avoid that the project of the HFRC coincides with other projects.

Many projects are working on gene expression during the development of an embryo. The rat and the mouse are the main mammalian reference models. The chicken, the zebra-fish and the fruit-fly are substantial for non-mammalian species.

Gene expression databases for the mouse

GenePaint [10]	http://www.genepaint.org
EMAP [11]	http://genex.hgu.mrc.ac.uk/Emage/database/emageIntro.html
GXD Gene Expression	http://www.informatics.jax.org/expression.shtml
Allen Brain Atlas	http://www.brainmap.org/

Gene expression databases for non-mammalians

Zebra-fish	ZFIN	http://zfin.org
Fly	Flybase	http://flybase.bio.indiana.edu
Chicken	Geisha	http://geisha.arizona.edu/geisha/

Discussing all mentioned databases would go beyond the scope of this paper. So only the most important gene expression databases for the mouse embryo, EMAP and GenePaint, are outlined. This description is based on a practical analysis of the web interfaces by using the online manuals.

Analysis of GenePaint

GenePaint is essentially a digital atlas of gene expression patterns in the embryonic mouse. For the research on the embryonic heart GenePaint is not precise enough, but it is sufficient to get an idea where a gene is expressed in the whole embryo. The main focus is on the ED 14.5 mouse, but it also features the ED 10.5 (only the head) and the ED 15.5 mouse. Furthermore newborn and adult mouse brains (day 7 and 56) are provided.

Database queries

The “**simple search**” enables the user to search for gene symbols, accession numbers and GenePaint set IDs.

The “**text search**” extends the simple search providing the option to search for gene name, gene symbol, gene description, and LocusLink ID, GenBank accession number or GenePaint set ID. Search terms can be combined via the Boolean operators “AND” or “OR”.

The “**expression pattern search**” uses annotations, outlined in the following, to search for genes expressed in specific organs or regions of interest.

The “**sequence homology search**” evaluates the similarity of the submitted DNA sequence to the sequences in the database by using a BLAST tool.

Displaying search results

Submitting a **text query**, e.g. for the gene symbol “Pax6”, returns the following table:


RESULTS							
Set ID	Accession no.	Gene	Tissue	Stage	Strain	Views	
HB182	X63963	Pax6: paired box gene 6	Head	E15.5	C57BL/6	Thumbnail	Set Viewer
HB304	X63963	Pax6: paired box gene 6	Brain	P7	C57BL/6	Thumbnail	Set Viewer
MH25	X63963	Pax6: paired box gene 6	Embryo	E10.8	NMRI	Thumbnail	Set Viewer
MH454	 X63963	Pax6: paired box gene 6	Embryo	E14.5	NMRI	Thumbnail	Set Viewer
Show	20	4 Hits	Page 1 of 1 select page: 1				

Fig. 4 Result of a text query for the gene “Pax6”

The links in the “Views” column allow viewing all images associated with the data set. High resolution images can be previewed by thumbnails.

Submitting the sequence of the “Fbox 2” protein in the “**sequence homology search**”, which can be derived at the Genbank website, gives back the following result table:

RESULTS							
Score	Set ID	Accession no.	Gene	Tissue	Stage	Strain	Views
190	HB102	XM_131839	Fbxo2: F-box only protein 2	Head	E15.5	C57BL/6	Thumbnail Set Viewer
190	HB273	XM_131839	Fbxo2: F-box only protein 2	Brain	P56	C57BL/6	Thumbnail Set Viewer
190	HB274	XM_131839	Fbxo2: F-box only protein 2	Brain	P7	C57BL/6	Thumbnail Set Viewer
190	MH115	XM_131839	Fbxo2: F-box only protein 2	Embryo	E14.5	NMRI	Thumbnail Set Viewer
190	MH216	XM_131839	Fbxo2: F-box only protein 2	Embryo	E10.5	NMRI	Thumbnail Set Viewer
190	MH217	XM_131839	Fbxo2: F-box only protein 2	Embryo	E10.5	NMRI	Thumbnail Set Viewer
101	EG559	NM_176848	Fbxo2: F-box only protein 2 (Fbxo2)	Embryo	E14.5	C57BL/6	Thumbnail Set Viewer
Show	20	7 Hits	Page 1 of 1 select page: 1				

Fig. 5 Result of a sequence homology search for the protein “Fbox 2”

The “BLAST Score”, which specifies the degree of homology between two DNA sequences, is shown in the left most column in the table. A graphical representation of the alignment of the query and template-sequences are also available.

Visualization

To enable viewing of images approaching the high resolution of the original data, a so-called “**Zoom View**” is used.

The viewer runs with any browser, without using an applet or installing a plug-in, best for slow computers, low bandwidth or uncommon operating systems.

Another option is to use the applet or the plug-in. However, this needs some premises and more computing power, but in return the handling of zooming and sliding is more convenient.

Annotation

The annotation of anatomical structures makes queries in regions of interest possible. Gene expression patterns are annotated by experts. The gene expression levels are scored automatically according to a standard scheme. GenePaint differentiates between three levels (weak, medium, strong) and three types of expression (ubiquitous, regional, scattered).

About 100 anatomical structures have been annotated and hierarchically organized.

Maps

There are also two types of maps for the identification of the localization of gene expression available:

1. Embryo maps

To provide the possibility to describe the localization of a gene, an anatomical annotation for sagittal sections of the embryonic mouse at ED 14 has been developed. This annotation can be viewed by sliding through the sagittal sections. But there is no direct connection with gene expression patterns.

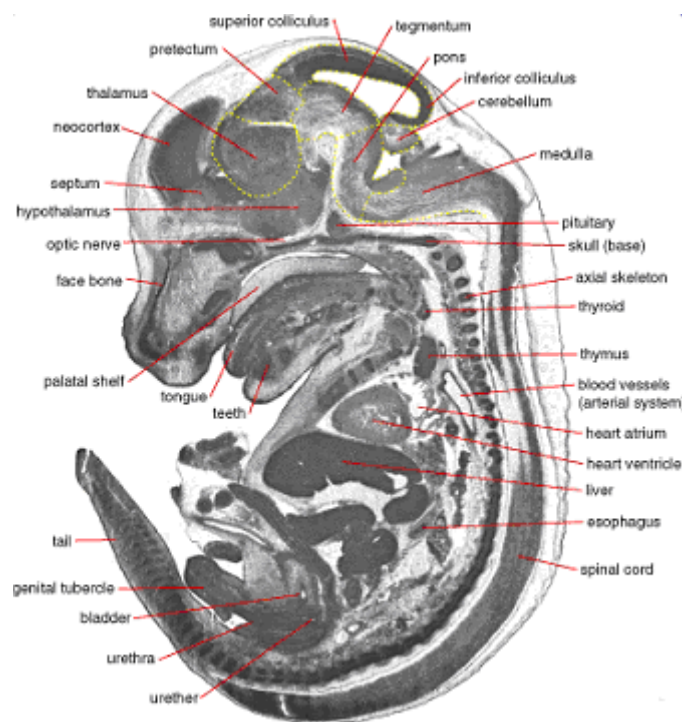


Fig. 6 Example of an embryo map: Mouse E 14.5 with distance from midline: 0.2 mm, left slide. Annotated regions are marked with a red pointer. Yellow lines are borderlines between the different brain regions.

2. Brain maps

The initial identification of regions of gene expression are supplemented by maps of brains of the embryonic mouse ED 15.5 and the postnatal mouse (P 7 and P 56). These maps label major brain regions and important nerve tracts.

Other features

The genes and specimens, which are currently in the database, are stated in a comprehensive gene list.

GenePaint also provides a service for requesting in situ hybridization for specific cDNAs. Thereby expression patterns for several hundred genes per year could be determined.

Analysis of the Edinburgh Mouse Atlas Project (EMAP)

The “EMAP Digital Atlas” portrays the development of the embryonic mouse. This atlas offers 3D computer models of mouse embryos at different developmental stages combined with hierarchical ontologies of anatomical terms. This atlas serves as a framework for the **Edinburgh Mouse Atlas of Gene Expression (EMAGE)**, database.

Database queries

The queries offered by EMAP are very comprehensive. It is possible to search for data according to the region/component where genes are expressed combined with the age of the embryo.

So the search interface can answer the following questions in context to a particular stage of development:

- Which genes are expressed in selected components?
- Which genes are expressed in selected regions?
- Which regions express genes?
- Which components express genes?

There are two types of **spatial queries** available: searching lateral views of whole mount stained embryos or searching within the 3D space of an embryo.

To search for specific domains in whole mount embryos it is either necessary to specify a region of interest in the standard embryo, or to use a predefined region for the search.

Three dimensional data can be searched by sliding through the slices with the “Slice Chooser” or just by specifying the coordinates.

Annotation

Text based

For every Theiler stage a hierarchically organized text based anatomical ontology is available on EMAP. Such ontology allows differentiating between diverse histological domains. This serves as a framework for researchers to depict unambiguously the location of an expressed gene.

Structure based

For every Theiler stage between TS 07 and TS 20 and also for TS 26, EMAP provides a 3D representative computer model. These models can be rotated in 3D space to allow visualization of surfaces. They also enable researchers to “virtually section” any plane to visualize internal anatomical structures.

Firstly each anatomical domain at each stage is painted manually by experts. Afterwards the labeled domains can be grouped. Thus a 3D reference model is created. These domains are linked with the anatomical ontology:

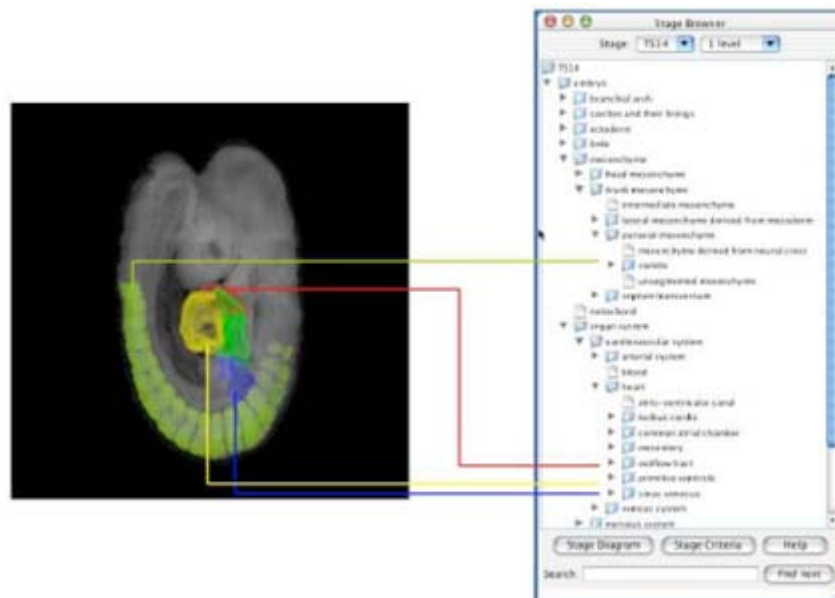


Fig. 7 3D Model with text anatomy descriptions

Displaying search results

As mentioned before, queries on EMAP are comprehensive. So in the following only one text and one spatial search is exemplified.

Text search

Selecting the cardiac muscle component of the mouse TS 14 in the component tree:

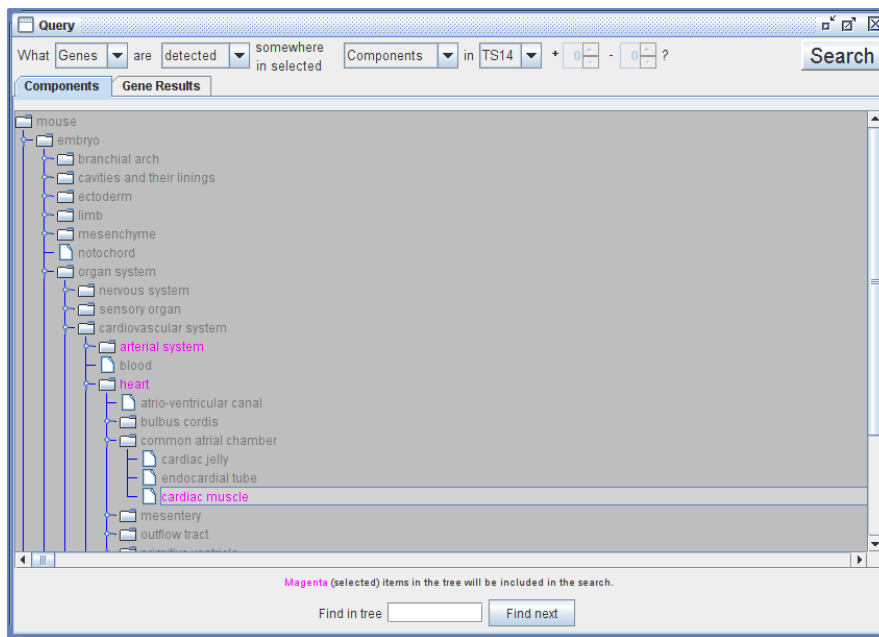


Fig. 8 Example of a text search

The screenshot shows the 'Query' window with the same search query. The 'Gene Results' tab is active, displaying a table of 39 assays with positive results. The table has columns for Gene, Rank, ID, Stage, Pattern Clarity, and Show Image. The gene 'Tbx5' is highlighted in blue.

Gene	Rank	ID	Stage	Pattern Clarity	Show Image
Foxc1	0,000	EMAGE:3651	TS14	***	<input type="checkbox"/>
Gata3	0,000	EMAGE:573	TS14	***	<input type="checkbox"/>
Gata4	0,000	EMAGE:3314	TS14	***	<input type="checkbox"/>
Hhex	0,000	EMAGE:3341	TS14	**	<input type="checkbox"/>
Hhex	0,000	EMAGE:3477	TS14	**	<input type="checkbox"/>
Htr2b	0,000	EMAGE:3441	TS14	**	<input type="checkbox"/>
Kdr	0,000	EMAGE:489	TS14	**	<input type="checkbox"/>
Kdr	0,000	EMAGE:881	TS14	**	<input type="checkbox"/>
Lmo2	0,000	EMAGE:1034	TS14	**	<input type="checkbox"/>
Mei2c	0,000	EMAGE:495	TS14	***	<input type="checkbox"/>
Myl2	0,000	EMAGE:709	TS14	***	<input type="checkbox"/>
Myl2	0,000	EMAGE:942	TS14	***	<input type="checkbox"/>
Myl7	0,000	EMAGE:945	TS14	***	<input type="checkbox"/>
Ncam1	0,000	EMAGE:3267	TS14	**	<input type="checkbox"/>
Nlx2-5	0,000	EMAGE:609	TS14	***	<input type="checkbox"/>
Nppa	0,000	EMAGE:4022	TS14	***	<input type="checkbox"/>
Pbx1	0,000	EMAGE:360	TS14	***	<input type="checkbox"/>
Pbx2	0,000	EMAGE:344	TS14	***	<input type="checkbox"/>
Ptch1	0,000	EMAGE:311	TS14	**	<input type="checkbox"/>
Smad2	0,000	EMAGE:3274	TS14	**	<input type="checkbox"/>
Smad3	0,000	EMAGE:3275	TS14	**	<input type="checkbox"/>
Snai2	0,000	EMAGE:3239	TS14	***	<input type="checkbox"/>
Sox2	0,000	EMAGE:320	TS14	***	<input type="checkbox"/>
Tbx5	0,000	EMAGE:4077	TS14	***	<input type="checkbox"/>
Tdgl1	0,000	EMAGE:477	TS14	...	<input type="checkbox"/>
Tie4	0,000	EMAGE:3612	TS14	...	<input type="checkbox"/>
Vcam1	0,000	EMAGE:1278	TS14	***	<input type="checkbox"/>

Fig. 9 Result of a text search

Spatial search

Defining a search region (magenta labeled) with the “paintbrush tool” in the whole mount mouse embryo TS 14:



Fig. 10 Example for spatial search

This search gives back a similar set of genes as the text search example. By selecting one of the genes, which were found by both searches, e.g. *Tbx5* the following window is displayed. It shows the spatial mapping of the gene, the associated picture of the whole-mount embryo and the site of expression in the ontology tree:

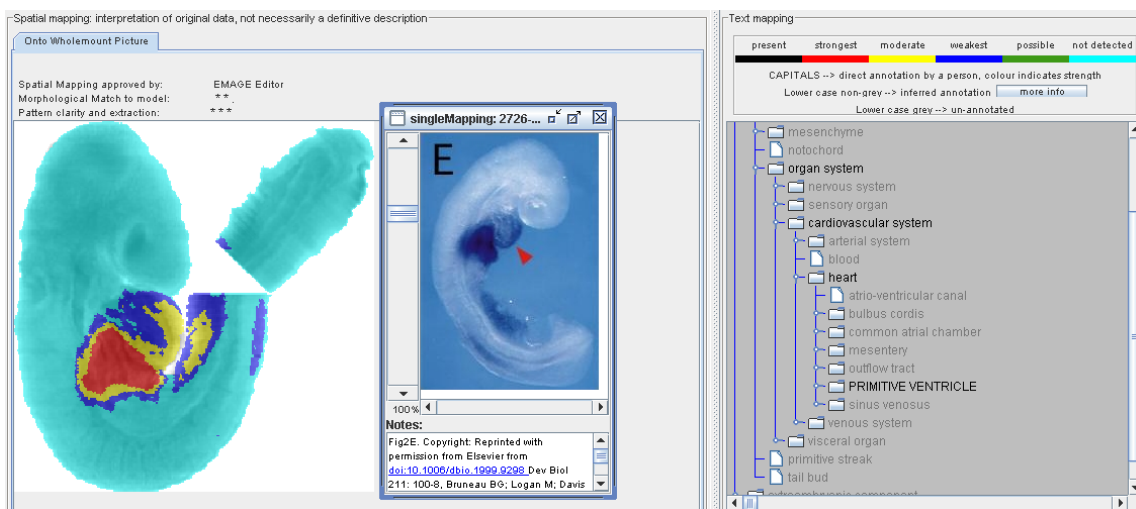


Fig. 11 Result of a spatial search

Visualization

There are several tools available to access the EMAP Mouse Atlas online. The most important is the “**Section Browser**”. This browser gives the possibility of viewing frontal, sagittal or transverse virtual sections from the 3D EMAP embryo models. It also enables to retrieve the name of a tissue within the displayed sections using your cursor or vice versa to find the location of a tissue within a section by its name. The Section Browser is also connected to Jackson Laboratory’s GXD database, which allows to look for information on the genes expressed in a particular tissue.

“**EmbryoView**” shows the reference model in 3D and lets the user change the view interactively. Viewing the rotating model in an mpeg-movie offers another chance.

“**Section movies**”

All of the virtual transverse, sagittal and frontal sections can be viewed as mpeg-movies.

“**High resolution section images**”

There is also the option to view digital images, which have been used for the embryo models, by accessing them through their reference number in the set.

Gene expression database

To store data in the EMAGE database a researcher has to submit his locally stored data to EMAGE. There it is evaluated by EMAGE editors and afterwards the information can be publicly accessed. The data that are stored in the database is text or spatial data.

Comparison of GenePaint and EMAP

At first sight GenePaint and EMAP seem to be completely different, but indeed they are serving the same purpose. However, for the research on the embryonic mouse heart, both of them are too inaccurate, which was mentioned by several researchers during the interviews (see section 5.1). Nevertheless they provide a good overview on gene expression patterns and give an idea whether a gene is expressed in the heart.

Database queries

GenePaint only supports text queries. The editors of EMAP annotate gene expression patterns in 3D space and link them to the 3D mouse atlas. So EMAP is able to provide more comprehensive queries by searching for genes relating to their region of expression or by selecting voxels in the 3D anatomical atlas.

Visualization

The possibilities for visualization of gene expression patterns offered by EMAP are one of the major advantages in comparison to GenePaint. The expression pattern of a gene can be viewed in a fully interactive 3D model. In GenePaint, however, genes are not visualized in the 2D embryo maps and are even not directly connected to them.

Annotation

GenePaint does not only provide annotation of the embryonic mouse, but also of the adult mouse brain. The different anatomical regions are marked in embryo maps, a set of 2D sagittal sections. These embryo maps seem to be more an anatomical dictionary. EMAP however offers ontology and a 3D anatomical atlas.

4. METHODS

Requirement analysis

The requirements analysis is the first step in the development of a software system. The goal is to elicit, analyze and record all demands of future users of the system. The result of this development stage is the requirement specification.

Interviews

Interviewing the later users of a software system is a known good strategy for gathering information to identify the requirements. For this project a questionnaire was elaborated to gather suggestions and opinions from potential users. It seemed that it was best to use open non-leading questions, in order not to restrict the possible answers with closed questions. Moreover, closed questions would have required an expert to prepare the questions and complete them. The open questions helped to get a deep insight into the project and a better understanding of the needs of the users.

The interview guideline is the basis for the requirements analysis and the information gathered via interviews is used to plan the use case diagram.

The guideline was structured as follows:

- The guideline starts with some general icebreaker questions concerning the research projects and the use of GenePaint and EMAP by the interviewed persons. This first part consists of five questions.
- The following part deals with EMAP and GenePaint in detail and consists of four questions.
- The last part concerns the project of the AMC, which includes the functionalities, the product data and the future perspectives of the project. This part consists of 11 questions.

The interviews were held orally, except one was performed via email. The full version of the questionnaire can be found in the appendix.

Use case analysis

Motivation

This method is used to identify requirements and to get an overview of all possible interactions of users with a software system.

For this project it serves the purpose of confirming the completeness of the requirements, which were determined through the preceding interviews. Therefore a brief use case analysis is sufficient.

Definitions

A **use case** describes an interaction with the system caused by an actor to achieve a certain business goal [14].

An **actor** is a person or a role which interacts with a system [14].

5. RESULTS

Summary of the interviews

The researchers were chosen on the basis of their insight of gene expression databases. Moreover these experts will be potential users of the program that should be developed.

Table 1 Interviewed researchers

Interviewed person	Role in the institute	Current research
Jan Ruijter	Biometrist and data analyst	<ul style="list-style-type: none">• Measurement of biological data• Supporting other projects• Development of databases
Alexander Soufan	Staff Adviser	<ul style="list-style-type: none">• Imaging of the heart• Web development
Wim Aanhanen	PhD student in medicine	<ul style="list-style-type: none">• Research on the heart with a focus on mouse hearts
Willem Hoogaars	Postdoc	<ul style="list-style-type: none">• Research on duchenne muscular dystrophy
Antoon Moorman	Head of the HFRC, Anatomy and Embryology Coordinator of the Heart Repair and CHeartED projects	<ul style="list-style-type: none">• Genetic annotation of the reconstruction of the embryonic heart
Maurice van den Hoff	Associate Professor	<ul style="list-style-type: none">• Research on the late myocardium• Finding out reasons for heart failures on the basis of knocking out genes
Philip Barnett	Leader of a research group, which tries to find protein partners with a main focus on the interactions with DNA	<ul style="list-style-type: none">• Mutation in TBX 5• Interaction of SOX Proteins• Finding TBX3 in the mouse heart

General questions concerning GenePaint and EMAP

The overall opinion was that the existing databases EMAP and GenePaint are not precise enough concerning the annotation, although the resolution of embryonic images is sufficient. Therefore it is not possible to distinguish the location of gene expression in the exact part of the heart. As a result no one uses GenePaint or EMAP on a regular basis, only one of the interviewed researchers uses GenePaint once or twice a month.

There are some other databases, which are used by the researchers for example GEISHA, which is a database for gene expression patterns in chicken.

Conclusion

- The annotation of these projects is not precise enough

Questions concerning the project of the HFRC

A problem is that many researchers are not familiar with the anatomy of a developing mouse heart. Hence, a determination of the exact position of gene expression patterns has to be made by an expert on mouse hearts. “More detailed structural information of sections linked to spatiotemporal expression patterns of genes would be a great advantage to identify specific structures (such as the chambers, valves, septa or conduction system) in which specific genes are expressed at any point in development.” (W.H.)

TRACTS enables to view a slice of the heart in the correct anatomical context, further the exact position of gene expression patterns is determined automatically. This would be a major improvement in the research on the embryonic heart, “because if you want to understand mechanisms you have to understand where the gene is expressed spatially, it is not sufficient only to say it is expressed in the heart” (A.M.).

Conclusion

- The spatial temporal information is necessary to understand mechanisms

Based on the functionalities of TRACTS there were several characteristics mentioned that should be provided by the web interface, the database and the visualization of results.

Questions concerning the functionalities of the web interface

Access Restriction (login)

Every interviewed researcher agreed that the visualization of the output results of TRACTS has to be accessible for public. The web interface should provide a service for researchers around the world. Nevertheless a login area has to be created for the website with a free registration, because of quality assurance.

A suitable solution for the quality assurance would be that an uploaded image is stored in a private or temporary database in case of a missing PubMed ID. If the PubMed ID is present the image is automatically stored in the public database **(A.S.)**.

PubMed is a free search engine for accessing the MEDLINE database of citations and abstracts of biomedical research articles [12]. The PubMedID is a unique ID for every article, which is published. The first approach to offer a quality assurance for the uploaded image was, to ask a researcher for the PubMedID when he uploads an image. On the basis of the article relating to the stored PubMedID it can then be proved that the information of the uploaded image is of good quality.

However, when a paper is published, it has already been reviewed and the use of the upload to see where the section lies in the heart is not necessary anymore. So the quality assurance for the unpublished images has to be handled by an administrator.

One of the interviewed researchers also mentioned: “at the beginning only members of the AMC are allowed to upload images and later it should be open” **(M.v.d.H.)**. The results of data mining should not be visible for public, because of competition.

Conclusion

- A login area with a free registration has to be created.
- The display of the search result should be possible without a registration.
- An administrator has to handle the uploaded images, because a quality assurance via PubMed ID is not possible.

Consequently in the use case diagram there have to be at least three actors. As mentioned in the interview a restricted image upload is needed but the results of the search should be public. Also there has to be a researcher who administrates the database. As mentioned above, a temporary database will not be created.

Three actors in the use case diagram:

- submitting researcher
- retrieving researcher
- administrator

Ontology, annotation and quality assurance of the gene expression data

There should be an anatomical annotation providing a rough overview, because a detailed annotation of the embryonic mouse heart would lead to disagreement between experts. “The AMC should provide a low level annotation. It is sufficient for example to globally annotate the ventricles, instead of a detailed categorization.” (M.v.d.H.)

There are dissensions about the use of an ontology between the members of the HFRC: “There is no ontology planned for the near future, but it is planned to annotate compartments based on the gene expression in a later stage.” (J.R.) “Nevertheless an ontology based on gene expression patterns underestimates the dynamic development.” (P.B.)

In contrast to the dissensions about the ontology everybody agreed on the use of the GenBank symbol to search for expressed genes. The interviewed researchers also agreed on the support of the staining methods ISH, ICC and reporter genes by the HFRC.

Conclusion

- A low level annotation should be provided.
- There is no ontology planned in the near future.
- When searching for genes, the official GenBank symbol should be used.

Description of data production

There were two contradictory statements concerning the information displayed on the website. “An explanation of how the data is produced should be available on the web interface”. **(J.R.)** “It is not necessary to store information on the website how the image is created. A description can be found using the PubMed ID.” **(A.M.)**

Conclusion

- No additional information about staining procedures or image acquisition will be provided on the website.

Design and visualization of output results

The design of the website should be the same as the one of the HFRC, which is currently being developed by **(A.S.)** to associate the AMC when using TRACTS. An interactive view like on EMAP with a higher resolution of the heart, further a focus on certain areas would be a good solution. If there is a movie on the site as output result of TRACTS that shows the slice in the 3D reference model from all perspectives, it should be possible to download it.

Conclusion

- The existing cascading style sheet of the HFRC site will be used for the web design.
- An interactive view should be available on the website.
- If a movie is shown on the website, there has to be the possibility to download it.

Future perspectives

A suggestion for an advanced version of the website from **A.S.** is: “If you put in a section with a new gene stained, it would be fine if there is a possibility to view the stained genes that are already in the database, so you get information about interactions of the genes”.

“The recognition of the age of the embryo from the program, for example the Theiler Stage or the embryonic day, would be another feature for the future”. (**P.B.**)

Later the program should be extended to different species like the chicken, or the snake and maybe linked to the database of GenePaint or EMAP.

Conclusion

- The database and the website should be extendable.

Description of product data

The product data is the basis for the data fields on the website. It is the most important information that has been elicited during the interviews. A basic version of the product data was created before the researchers were interviewed. The advantage of a basic version was that the researchers could tell what was missing and which fields had to be changed.

The resulting product data is listed as follows:

Table 2 PD1: Information describing submitted images

Data	Required	Optional
Species (mouse/chicken)	X	
ED*	X	
Appropriate age scale (HH/TS)**		X
Transgenic/Wildtype	X	
Expressed Gene	X	
Stain	X	
Treatment		X
PubMed ID of the associated paper		X
Strain		X
Thickness of sections		X
Dimensions of the embryo		X

* ED = Embryonic Day

** HH = Hamburger Hamilton Stage (chicken) / TS = Theiler Stage (mouse)

Table 3 PD2: The image and its attributes

Data	Required	Optional
Resolution ($\mu\text{m}/\text{pixel}$)	X	
Slice thickness		X
Number of Slices		X
Position of the slice		X

Table 4 PD3: Information describing submitting persons

Data	Required	Optional
First Name	X	
Last Name	X	
Email address	X	
Institution	X	
Password	X	

Use case analysis

The “use case diagram” is used to give a brief overview of possible interactions of a user with the system [13].

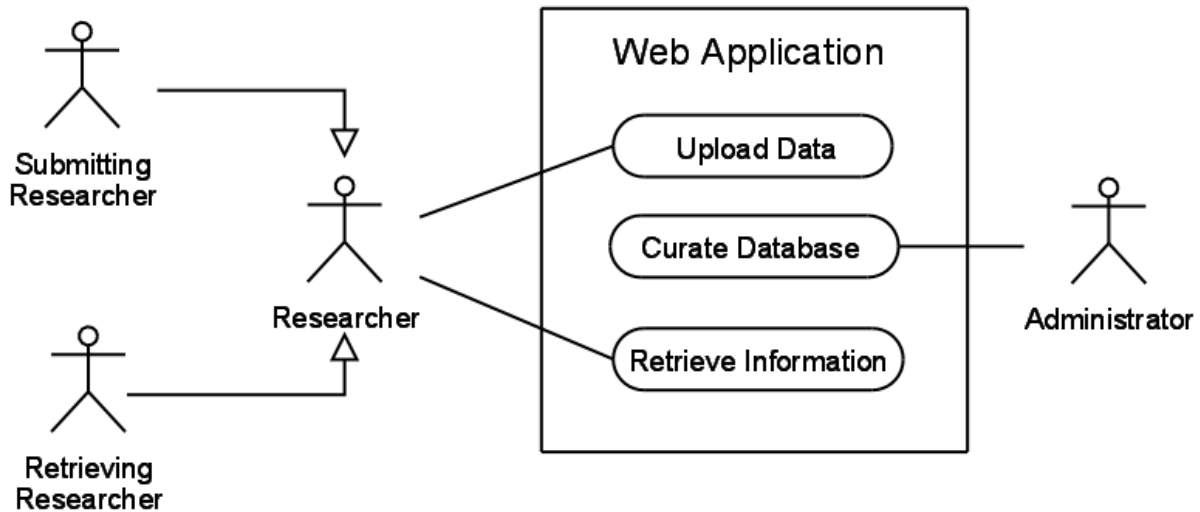


Fig. 12 Use case diagram

Use case 1: Upload data

Actor: Submitting researcher

A researcher wants to upload an image on the server. If the researcher is not registered, a registration has to be done first. Once a successful registration is performed the researcher has to login for uploading an image. After filling in at least the required fields, the image can be attached. When the image is uploaded, TRACTS is able to fit the section in the 3D reference model. In case TRACTS is successful, the result is displayed on the website. Later the researcher can log out and leave the website.

Use Case 2: Retrieve information from the database

Actor: Retrieving researcher

A researcher wants to retrieve information from the website, therefore visits the website and opens the search section. After filling in the required search parameters as for example a gene expression pattern, the search is submitted. Following the found embryos are displayed with thumbnails on the website, which can be later visualized by TRACTS.

Use Case 3: Curate database

Actor: Administrator

An employee of the HFRC wants to ensure the quality of the submitted data. The researcher visits the website and logs in as administrator. Afterwards the not yet curated data has to be selected and validated. In case of an approved embryo by the administrator, the data is set visible.

Requirements specification

The main goal for fitting 2D sections of the embryonic mouse heart in a 3D reference model (TRACTS) is, to make an existing MatLab application available through the Internet. It should be possible to submit images, to store these images and according data in a database and to visualize the results of TRACTS in an appropriate way.

Subproject requirements

1. Web interface

- 1.1. Input of text information and images
- 1.2. Quality assurance for the submitted data
- 1.3. Login procedure
- 1.4. Determine a ROI

2. Database

- 2.1. Find an appropriate database management system
- 2.2. Evaluate a database scheme
- 2.3. Integrate the database in a web application
- 2.4. Find a solution for the storage of images

3. Visualization

- 3.1. The Program returns where the submitted image best fits in the reference model.
- 3.2. This result should be visualized by showing an interactive 3D visualization of the reference model.
- 3.3. There should also be an overlay of the submitted image in the interactive 3D visualization of the reference model.

Requirements for the HFRC

Some elicited requirements have to be dealt by employees of the HFRC:

1. A low level annotation should be provided.
2. TRACTS should be able to create a short movie, where the 3D reference model with the placed section is shown. This can be used by researchers for their presentations.
3. TRACTS should be extended to enable the handling of other file formats like TIFF as input images.
4. Queries for the database should be created.

Software specification

- Eclipse 3.2 is an integrated development environment for Java
www.eclipse.org
- JABuilder
The JABuilder is a commercial compiler from MATLAB. It compiles MATLAB files into jar files (executable java files), which can be easily be integrated in Java-based applications.
- Third party libraries:
Hibernate Core 3.3.1.GA: <http://www.hibernate.org/6.html>
Commons fileupload 1.2.1: <http://commons.apache.org/fileupload/>
com.oreilly.servle cos-05Nov2002: <http://www.servlets.com/cos/>
MySQL Connector/J 3.1:
<http://dev.mysql.com/downloads/connector/j/3.1.html>
JA Builder 2.0.1

II. WEB INTERFACE

1. BASICS AND STATE OF TECHNOLOGY

Motivation for a web interface

Research in the field of molecular biology creates a lot of information, which has to be stored somewhere. Therefore, there is a main focus on databases in the field of Bioinformatics. Due to the huge amount of research data gathered, it is not favorable to have the data stored locally in a database. Furthermore, the information should be shared with researchers around the world to increase the effectiveness of the research.

The goal of our project is to create a web interface for making the results of the research from the AMC available for public. A web interface providing detailed information of an embryonic mouse heart is currently not offered and should be realized within this project.

The present chapter describes the basics and definitions in order to understand the development of a web site. Only the most important technologies in the field of web development concerning this project are described as follows.

WWW and HTML

The **World Wide Web** (WWW) created by Sir Tim Berners-Lee, a computer scientist at CERN, is a system based on hypertext documents that are accessed through the Internet. The navigation through the documents is handled via hyperlinks. The markup language for the structure of documents, like headings, paragraphs etc., in the WWW, is called **Hyper Text Markup Language** (HTML). HTML provides text based documents with interactive forms, embedded images and other objects. It can also describe the appearance and semantics of a document and it possible to include embedded script language code like JavaScript, which is explained later [15][16].

Cascading style sheets

Cascading Style Sheets (CSS) are used to describe the formation of a document written in a markup language like HTML. It can be used to define colors, fonts, layout, and other aspects of document presentation. With CSS a separation of the presentation and the content of a website can be achieved [17].

Web server

A web server can be a computer program that accepts HTTP requests from web clients, serving them HTTP responses along with optional data contents, or it can be a computer that runs the just described computer program. The most popular free web server is the Apache Tomcat [18].

Apache Tomcat

The Apache Tomcat is a Servlet container developed by the **Apache Software Foundation (ASF)**. Tomcat implements the Java Servlet and the **JavaServer Pages (JSP)** specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run" [19].

Client-server architecture

The client-server architecture separates service and request in a network. The Server, which is a computer program as the client, fulfills a request sent by the client with a proper response [20].

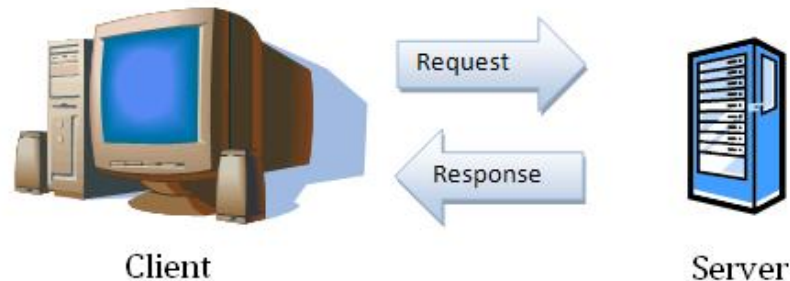


Fig. 13 Client-server architecture

Multiple tier architecture

The most important software architecture for websites is the three-tier architecture (figure 14), which was also used for the design of this web interface. The Three Tier architecture is considered to be a software architecture and a software design pattern that enables a programmer to use the tiers as separate modules, most often on separate platforms. This allows a software developer to replace or upgrade any of the three tiers independently [21].

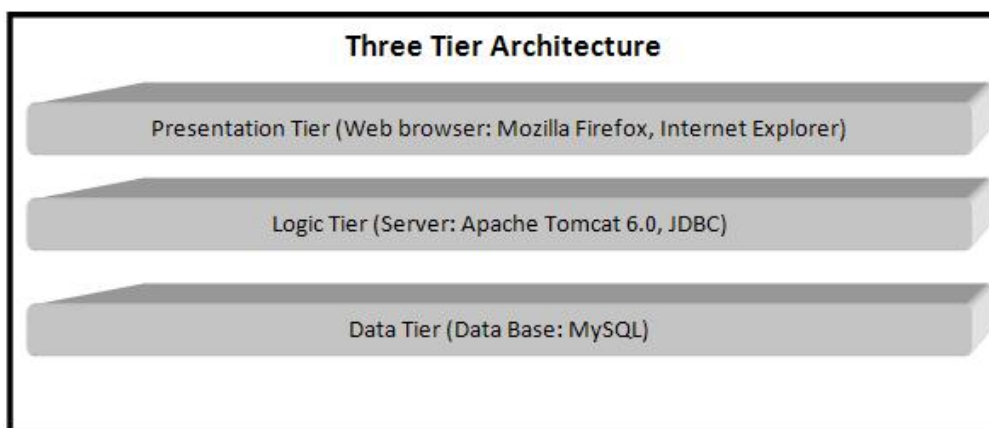


Fig. 14 Three Tier architecture: The software used in the thesis is shown in parentheses next to the different layers

Dynamic web pages: client-side scripting

Client-side scripting is used to change the interface of a specific web page, in response to mouse, keyboard or timing events. As stated in the name, the script runs on the computer of the client [22].

JavaScript

JavaScript is a scripting language used by client based web programming. Despite the name it is unrelated to the Java programming language, but it copies the Java naming convention [23].

Dynamic web pages: server-side scripting

Server-side scripting is used to change the state of a website after the request is sent to a server where it is processed. Based on this technology it is possible to develop interactive web sites involving the storage of data. The most important server-side technologies are discussed in detail along these lines.

Common gateway interface

The **C**ommon **G**ateway **I**nterface (CGI) is one of the first techniques for creating dynamic web content. A web server passes a request to an external program and the results will be sent to the client [25].

PHP

Hypertext preprocessor (PHP) former “**P**ersonal **H**ome **P**age” is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML [24].

Java Servlets

Java Servlets are a generic extension to a server – a Java class, which can be loaded dynamically to extend the functionality of a web server. Java Servlets run on a web server instead of CGI and are compiled via **Java Virtual Machine (JVM)** as usual Java programs. Therefore, Servlets are portable and do not need a virtual machine on the client.

The Servlet API, provided by Java Sun contains the packages `javax.servlet` and `javax.servlet.http`, which allows a software developer to add dynamic content to a web server using the Java platform.

A Servlet provides methods like `doPost` or `doGet` to handle HTTP-requests like GET or POST. There are several other methods provided by Servlets that are not discussed further.

Sample Java Servlet implementation for “Hello World”:

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
public class HelloWorld extends javax.servlet.http.HttpServlet implements
```

```
javax.servlet.Servlet {
```

```
    static final long serialVersionUID = 1L;
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {
```

```
        response.setContentType("text/html");
```

```
        PrintWriter out = response.getWriter();
```

```
        out.println("HTML");
```

```
        out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
```

```
        out.println("<BODY>Hello World</BODY>");
```

```
        out.println("</HTML>");
```

```
    }
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {
```

```
        doGet(request, response);
```

```
    }
```

```
}
```

In this sample a Servlet is shown which shows “Hello World” in a web browser. To simplify the presentation via HTML another technology, JSP was introduced by Java Sun [25].

JavaServer pages

JSP is the latest Java technology in web development to dynamically generate HTML XML etc. A JSP code is compiled into a servlet before it is executed. JSPs should be used for presentation.

Sample JSP implementation of “Hello World”:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%
//scriptlet: Some Java Code
String hello = "Hello";
%>
<html>
<head><title>Hello World</title>
</head>
<body>
<%!String world = " World"; %> <!-- declaration -->
<%= hello + world %> <!-- expression -->
</body>
</html>
```

In this sample a JSP shows “Hello World” in a web browser. There are several types of embedding java code in HTML code shown in this sample which are commented above [25].

2. METHODS

Diagram creation programs

The following modeling tools were used to plan the web site:

DIA

DIA is a diagram creation program for Linux, UNIX and Windows, released under the GPL license. DIA was used to design sequence diagrams [26].

ArgoUML

ArgoUML is the leading open source UML modeling tool and includes support for all standard UML 1.4 diagrams. UML was used to create the class diagram [27].

Microsoft Office

Microsoft Office 2007 was used to create the navigation diagram and to visualize the architecture of the program.

Analysis of GenePaint and EMAP

The study of existing web sites, which provide similar applications, is an important part in the planning of a web site. As a result the analysis of GenePaint and EMAP described in the common part (see section I.3), was used to design the web interface. The navigation of GenePaint and the information stored on these websites are the basis of the development of the web interface. The idea of EMAP to execute an application as a downloadable Java program on the computer of the client was not a proper solution for the AMC. However, it was required that the Matlab program should be executable from a browser.

Requirements analysis

The requirements analysis (see section I.4.1) is the basis of the functionalities and form fields that should be implemented. The conclusions of the interviews together with the use case analysis, the product data describe the requirements of the whole system and therefore also for the web interface. The requirement specification for the web interface can be looked up in the common result chapter.

The use case 1 (see section I.5.2) describes the process of uploading an image that serves as an abstract of the website-navigation.

The product data describes the form fields, shown in the screenshots of the website in the results part. In the result part of the database chapter (see section III.3), the product data is discussed in detail.

Requirements deduced from the interviews

- Access restriction (see section I.5.1)
A login area has to be created with a free registration, but the display of the search result should be possible without registration.
- Description of data production (see section I.5.1)
No additional information about staining procedures or image acquisition will be provided on the website.
- Design and visualization (see section I.5.1)
The existing CSS file of the HFRC site will be used for the web design and there should be an interactive view available on the website.
- Future perspectives (see section I.5.1)
The website should be extendable.

Software specification

The software specification describes the programming language and software used to develop the web interface.

- Java Servlet and JSPs

Due to the fact that the JA Builder (see section I.5.3) is written in Java, Java Servlets and JSPs were chosen as web programming language. In the beginning of the project PHP was used, but as was not a proper solution.

- Apache Tomcat

As Apache Tomcat is a free and a popular web server for Java based web sites, it was used in this project.

- JavaScript

JavaScript was chosen for checking form fields on the client side.

- CSS

As described above, the interviewed researcher suggested using the same design as the currently developed website of the HFRC.

Navigation diagram

A navigation diagram is used to get an overview of the pages that should be implemented. The diagram is discussed in the result section (see section 3.2) and shows the navigation through the website.

Form based authentication

To handle the requirement of the login area a secure solution had to be found. Form-based authentication allows customizing the error pages and the login screen, which are presented to the end user. It is a secure solution and a common method to protect web sites.

Explanation of the login procedure [28]

1. A client requests access to a protected resource
2. If the client is unauthenticated, the server redirects the client to a login page.
3. The client submits the login form to the server.
4. If the login succeeds, the server redirects the client to the resource. If the login fails, the client is redirected to an error page.

In figure 15 the actions of this process are shown

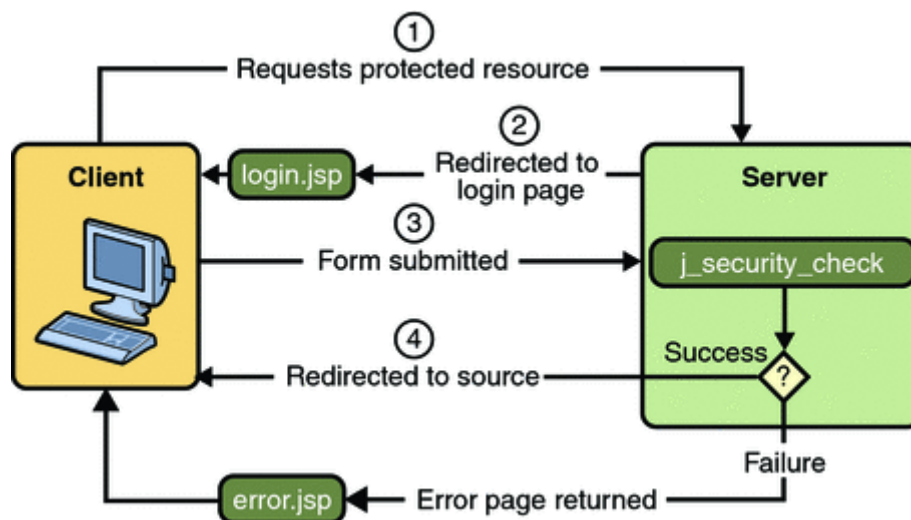


Fig. 15 Form Based Authentication

Example code for the form-based authentication:

```
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>Form-Based Authentication</realm-name>
  <form-login-config>
    <form-login-page>/login.jsp</form-login-page>
    <form-error-page>/error.jsp</form-error-page>
  </form-login-config>
</login-config>
```

The problem with form-based authentication is an unauthenticated target server and the lack of an encrypted password, since the content of the user dialog box is sent as plain text. To avoid this, the connection should be over **Secure Socket Layers (SSL)**¹. Therefore, it has been recommended to the HFRC to use a SSL connection, when the website is published [29].

MD5 encryption

MD5 (**M**essage-**D**igest algorithm **5**) encryption is a widely used cryptographic hash function with a 128 bit hash value. It was designed by Ronald L. Rivest in 1991 [29].

¹ SSL: a secure encrypted communication protocol

Software architecture

In the conclusion (see section 1.5.1) of the requirements analysis is stated that the program should be extendable. To provide this expandability, at least a basic software architecture had to be realized. In the following the basics of the most important design pattern is described. The used software architecture is discussed in the result chapter (see section 3.1).

Model View Controller (MVC) pattern

- The **model** contains the data that should be represented and does not know anything about controller or view.
- The **view** is the presentation of the data which is stored in the model.
- The **controller** interacts with the user and changes the state of the model.

MVC2 pattern

Due to the HTTP-protocol (every time a state changes a browser must query the server to get information about the changed state) the MVC pattern has to be changed. Therefore the realization of the pattern in the web is called the MVC2 pattern (figure 16).

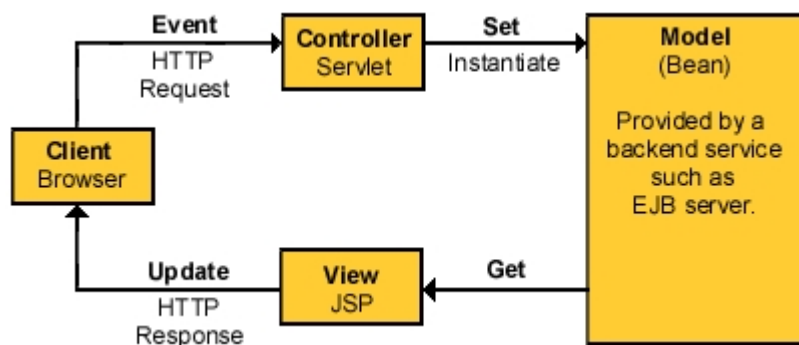


Fig. 16 MVC2 pattern

As shown in figure 16 a client sends a request to the server, which is handled by a controller (Servlet). This Servlet sets the state of the model and the view displays the changed state of the model [31] [32].

3. RESULTS

Architecture of the web application

The architecture of the program combines JSP with Servlets, allowing a separation of the view from the control. This strategy makes the program extendable and easier to maintain.

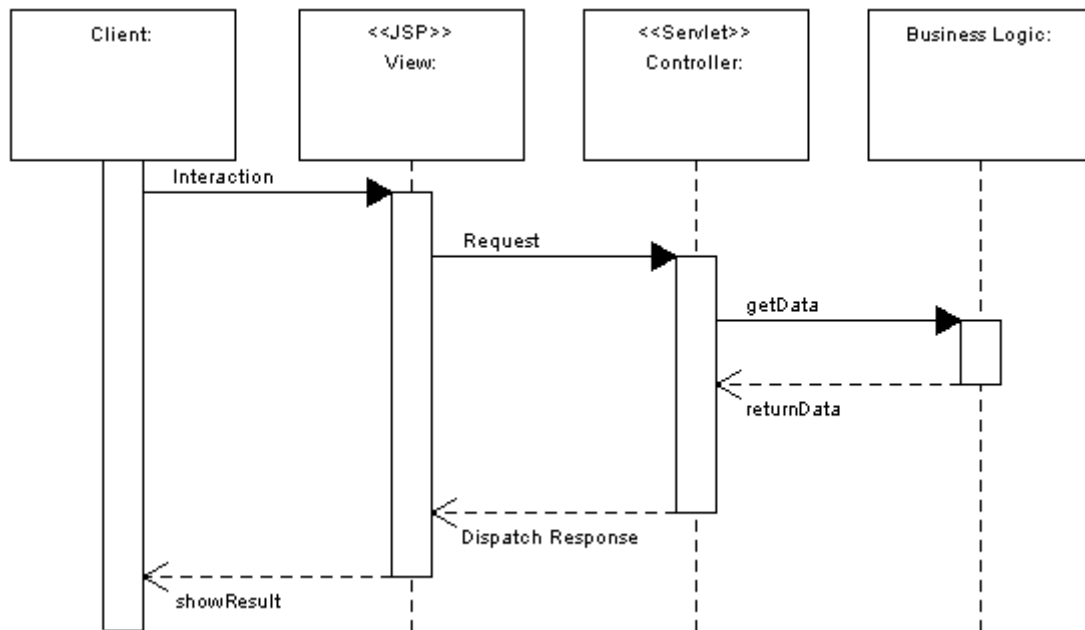


Fig. 17 Sequence diagram of the web application architecture

As shown in figure 17 the Servlet serves as a controller and handles the request. It calls the business logic to get the requested data and dispatches the response to the JSP to view the changed result to the client.

As demonstrated in figure 18, when the client interacts with the web application, first of all JavaScript checks this interaction. In case of a positive statement, the request is forwarded to the Servlet. The Servlet as a controller, checks the request again. Furthermore it is able to call the business logic to get the needed data and/or a helper class to process the request. After this is done the request is dispatched to the JSP where the result of the interaction is displayed to the client. The words in parenthesis illustrate the correlation to the MVC pattern.

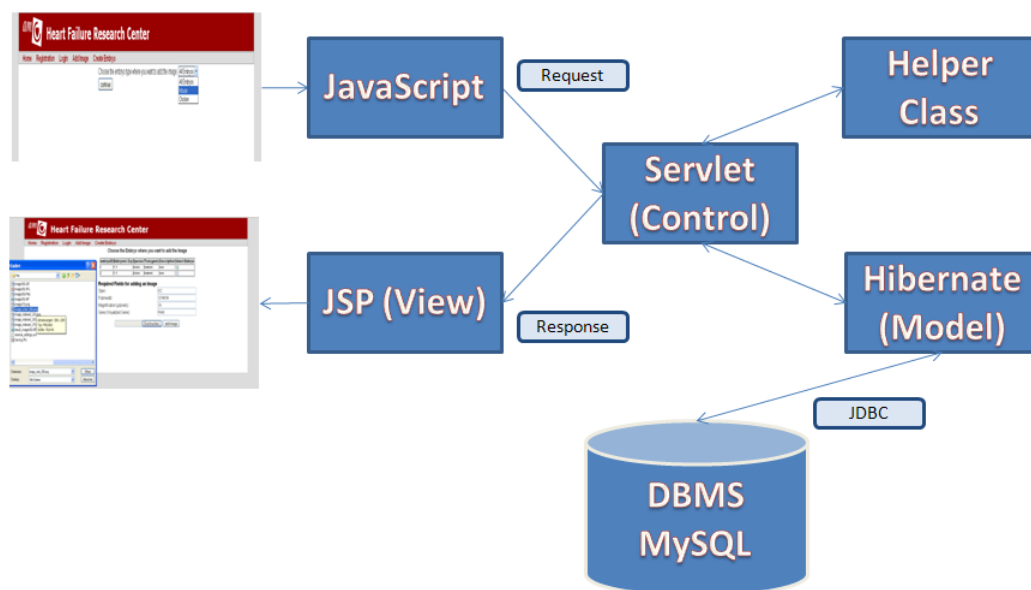


Fig. 18 Architecture of the web interface

Navigation diagram

Figure 19 shows the navigation through the websites of the prototype. In the chapter "discussion and future perspectives", an advanced navigation diagram is illustrated. The single websites are discussed as follows in detail.

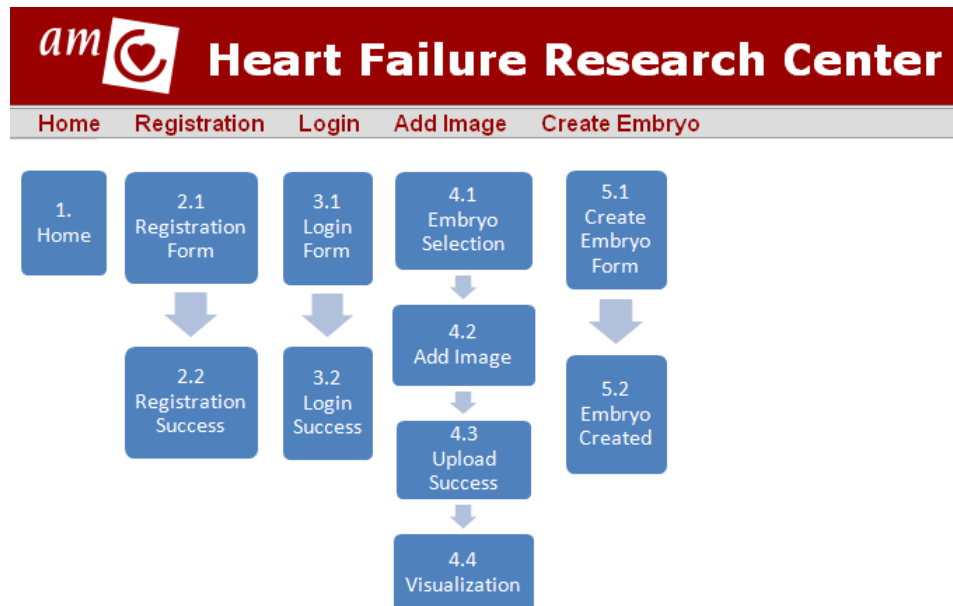


Fig. 19 Navigation diagram of the current web interface

Security

A login area had to be created, as already stated in the requirement specification (see section 1.4.1). Therefore, a form based authentication was implemented as described in the methods chapter (see section 2.5). To provide secure storage of the password, it is encrypted by MD5 encryption and stored in the database. Figure 20 illustrates the registration process.

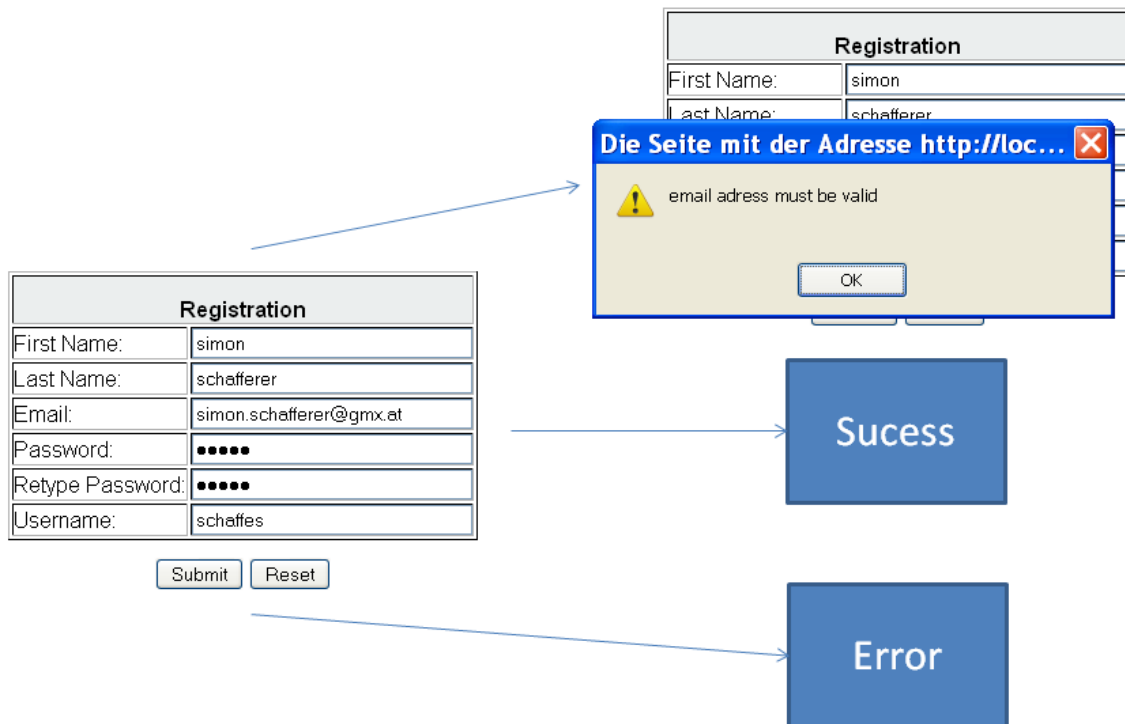


Fig. 20 Registration process

The user has to register, before an image can be uploaded. In order to do this it is necessary to fill out the registration form. The form fields are checked via JavaScript and in case of an empty field, passwords that do not match, or an invalid email address a pop up window informs the user. If the client disables JavaScript, the fields are also checked by the Servlet where the data is processed. The result of the check by the Servlet is either an error page or a successful forwarding to a "registration success" page and the storage of the data fields in the database. The password is first encrypted by a Java helper class before it is stored in the database as shown in figure 21.

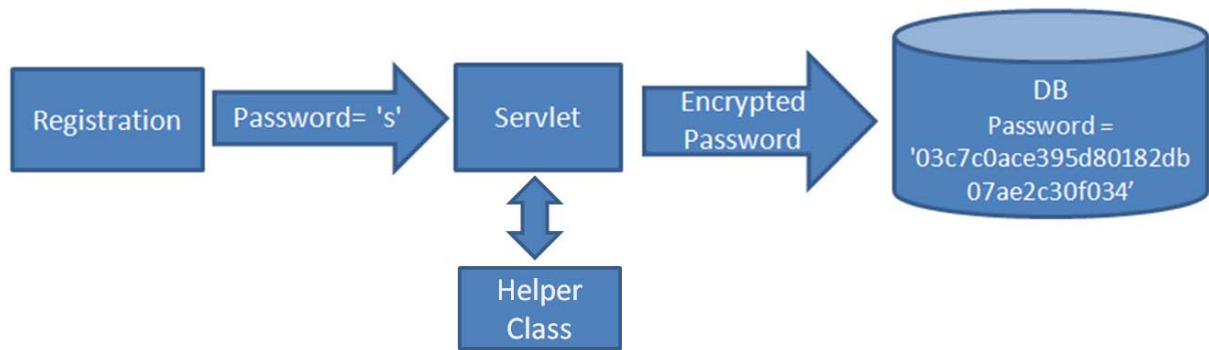


Fig. 21 Registration procedure via MD5 encryption

After a successful registration the password is sent to the Servlet where it is encrypted and then stored in the database as a 32 bit sequence. Later when a user wants to upload images and has already registered a login by filling out the login form as shown in figure 22 has to be made.

Username:
Password:

Fig. 22 Login interface

This login form is displayed every time a user wants to open a protected site for example the “image upload”. There is also a link to the login form, as shown in the navigation diagram (figure 19).

Creation of an embryo

The creation of an embryo shown in figure 23 is an essential step before an image can be added. Images can only be added to an embryo. It is not possible to upload an image that does not relate to an embryo. To the left a mouse embryo is selected and the age scale is adapted from “Hamburger Hamilton” to “Theiler stage”. The additional field “strain” is enabled and filled. This is realized via JavaScript as well as the checking of the fields if they are empty. On the right side a chicken embryo is selected with the appropriate “Hamburger Hamilton” scale. The information sent to the Servlet as a String is also checked there and the parameters are converted in the proper values for the storage in the database.

The figure displays two side-by-side web forms for creating an embryo. Each form has the following fields and controls:

- Choose Species:** Radio buttons for 'Mouse' and 'Chicken'.
 - Left form: 'Mouse' is selected.
 - Right form: 'Chicken' is selected.
- Embryonic Day:** Text input field containing '11.5'.
- Treatment:** Text input field containing 'treatment'.
- Description:** Large text area containing 'none'.
- Institution Embryo ID:** Text input field containing 'institutionEmbryoID'.
- Strain:** Text input field containing 'strain'.
- Hamburger Hamilton:** Dropdown menu with '8' selected.
- positive:** Dropdown menu with 'positive' selected.
- Theiler Stage:** Dropdown menu with '4' selected (left form) or '1' selected (right form).
- Buttons:** 'Submit' and 'Reset' buttons at the bottom.

Fig. 23 Creation of a mouse and a chicken embryo

Adding of an image

After the successful creation of an embryo an image can be added.

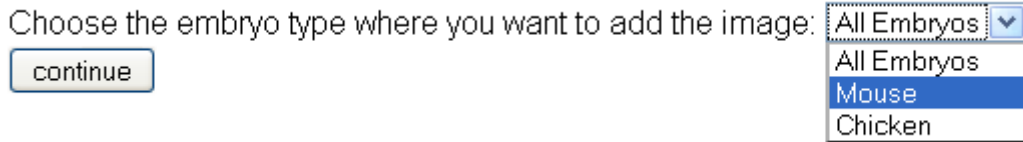


Fig. 24 Embryo selection

As shown in figure 24 an embryo type has to be selected first, before an image can be added. It is possible to display all the created embryos, the mouse or the chicken embryos. After the selection the state is forwarded to the “upload image” page.

Upload of an image

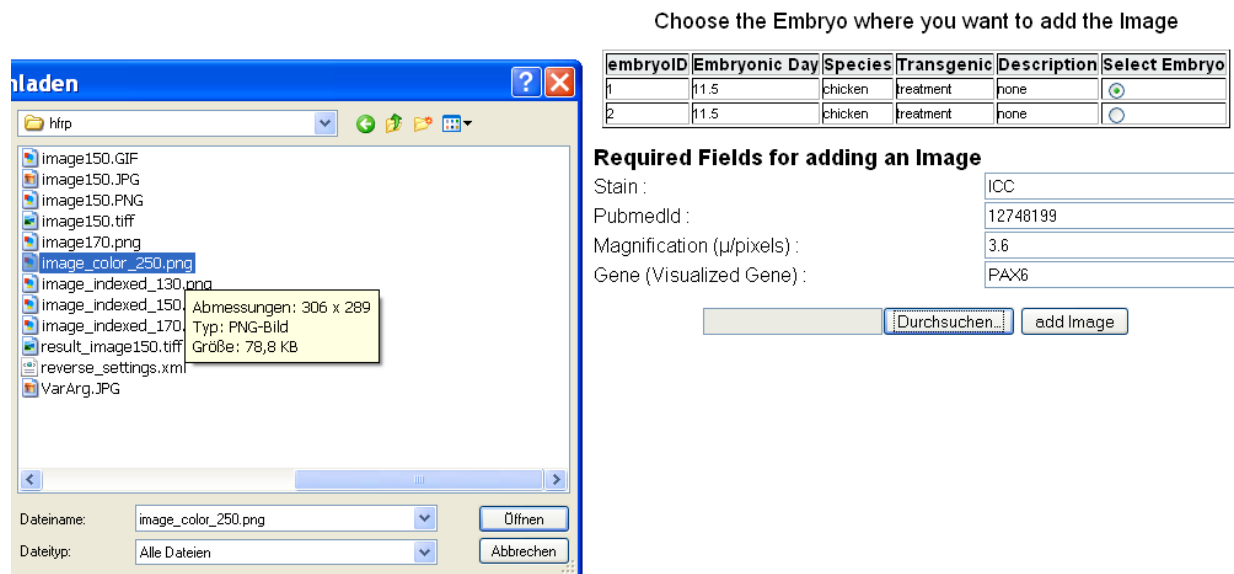


Fig. 25 Upload of an image

In figure 25 the result of the data base query of the selection from figure 24 is shown on the top right. The existing embryos are displayed in a table and an embryo has to be selected where an image should be added. After selecting an embryo the form has to be filled out as shown in the middle right and the image has to be uploaded (left). After the upload, the image, the state of the selection and the form fields are

passed to the Servlet, which stores all the information and calls TRACTS to visualize the fitted section. The forwarding of the form fields together with the uploaded image all presented on one page was solved with a third party library called “multipart parser”.

Visualization

After Matlab is called and the fitting of the section is computed, TRACTS returns a web figure, the resulting image and the overlay image that is stored in the database and forwarded to the view to visualize the result. The returned coordinates are also stored in the database, to compute the best fit only once per image. The storage of the values is discussed in chapter 0. The resulting image is the section that fits best of the reference sections and the overlay image shows the resulting and the uploaded image in one picture.

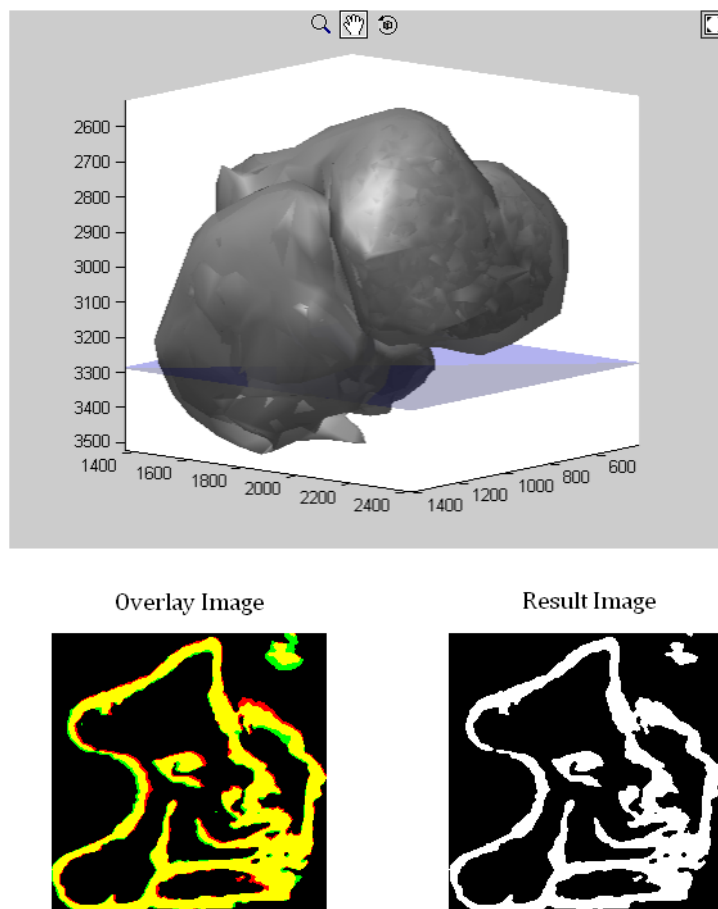


Fig. 26 Visualization of the result

In figure 26 the 3D visualization of TRACTS (top) with the resulting image (bottom right) and the overlay image (bottom left) are shown. The web figure is fully interactive, which allows rotation, zoom and shifting operations.

4. DISCUSSION

The accomplished goals

Table 5 The accomplished goals of the specified requirements.

1.1. Input of textual information and images	+
1.2. Quality assurance for the submitted data	+
1.3. Login procedure	+
1.4. Determine a ROI	-
1.5. Input of database queries	-

The accomplished requirements are marked with a plus on the right, whereas the unattained are marked with a minus.

As shown in table 5 the most important requirements were reached. The determination of a region of interest (ROI) was not implemented because of a lack of time and a low priority. Due to the architecture of the program, the missing database queries can be implemented without much effort in the near future.

Architecture

As described in the result chapter the architecture of this project relates to the MVC2 pattern, what poses the question, why a web application framework like Apache Struts or Java Server Faces was not used?

An argument why this kind of framework is not applied is the lack of experience in the field of web development combined with the small amount of time given. On the other hand a complex web application framework is beyond the scope of the developed program.

Software specification

Former PHP was used to implement the web application. After further analysis of the opportunities to call Matlab programs, the solution via Java was found. It would have been better to spend more time in the analysis of the different communication types, before concentrating on the programming language used to implement the web application. This cost a lot of time and also hindered the implementation of optional requirements like the determination of the ROI.

Future navigation frame

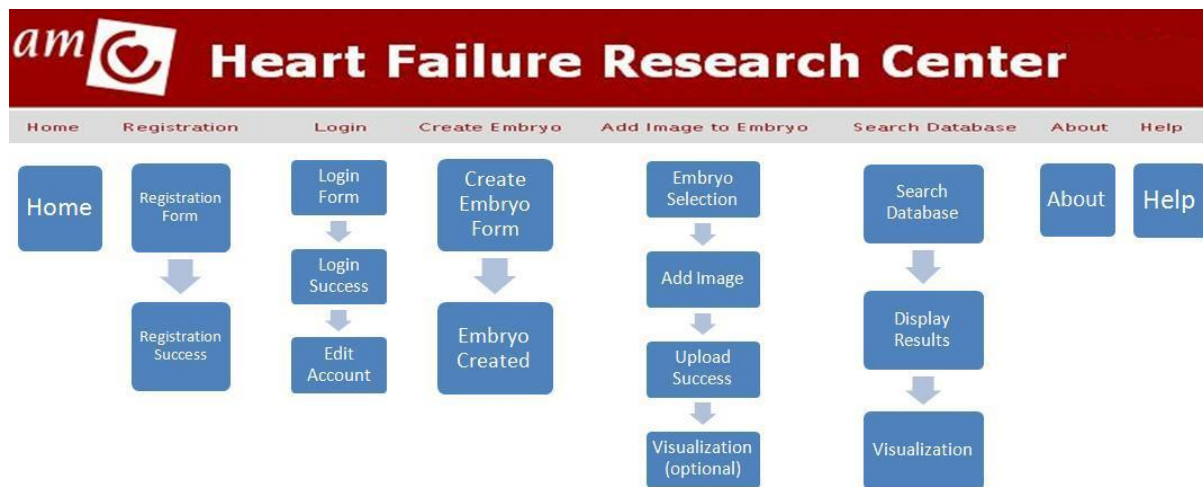


Fig. 27 Future navigation of the web interface

In the first version of the program there were several things that have to be changed, when the website is published. In figure 27 a sample of a future implementation of the website is shown. After an image is added, the visualization is obligatory. In a future version there should be the possibility to choose either to visualize the uploaded image, or to upload further images. By now, a query site is missing, where database queries can be made. Figure 27 illustrates such a routine where the database is first queried and the results are displayed afterwards. For example a gene is queried and the database returns all images with the required gene, which are displayed as thumbnails on the website. The thumbnails can moreover be enlarged and visualized via TRACTS. Finally additional information in form of a manual etc. should be present on the website to provide integrity.

Generalization of the project

All in all generalization was achieved by the architecture and implementation chosen for the web interface. The further development of an extended version of the website as shown in figure 27 should be possible for other students or employees of the AMC prospectively.

III. DATABASE

1. STATE OF TECHNOLOGY

This chapter describes the principles of a database and its integration in an object-based application.

Simply storing files in a file system leads to several problems, like redundancy, inconsistency, no synchronisation of data access, low protection from data loss and low data security concerning privacy, especially in a multi-user system. These problems can be avoided by storing data in a database system consistently.

Principles of a database

Definitions

A **database system (DBS)** consists of two components, a database and a database management system (DBMS) [33].

A **database** is a structured collection of records or data. This structure is also known as **database model** [34].

A **DBMS** is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database [35].

Architecture of a DBS

Three layers build up the architecture:

1. The **Physical Layer** is responsible for the storage of data, mostly on a hard drive.
2. The **Logical Layer** defines where data are stored in the database model.
3. The **Views** divide the logical layer into subsets to provide different user groups with data only necessary for their requirements.

These three layers enable to achieve two levels of **data independence**:

1. Physical data independence

The logical layer is independent from the physical layer. Thus changes of the data model do not affect the physical layer.

2. Logical data independence:

Changes of the physical layer cannot be recognized by the users [36].

Data models

Data models are the basis for the design of a database, as they formalize the structure of data. They define data objects and operations to manipulate them.

It is possible to distinguish between two data manipulation languages:

1. The **data definition language (DDL)** defines the structure of data objects. All data objects determine the database scheme.
2. The **data manipulation language (DML)** is used to store, modify, or delete data objects. Moreover it defines the query language.

The different data models [37]

1. Relational data model
2. Hierarchical data model
3. Network data model
4. Object model
5. Object-relational model

The **relational data model** is the most common one. It can be imagined as a set of tables. So a table row would be a data record consisting of attributes determined by the table columns. Data records are uniquely identified by a key attribute (primary key), which enables them to relate among each other.

Integrating a relational database into an object-based application

To guarantee that an application is maintainable and extendible, it should be kept independently from the data model. This means that the database layer of an application should be encapsulated.

Data Access Object (DAO) pattern

The DAO pattern is a common and proved approach to abstract the database from the business logic layer.

The data access objects are interfaces providing operations to access and manipulate the database. Hence, the business layer of an application is not dependent on the implementation of the database.

The following example is an illustration of this pattern:

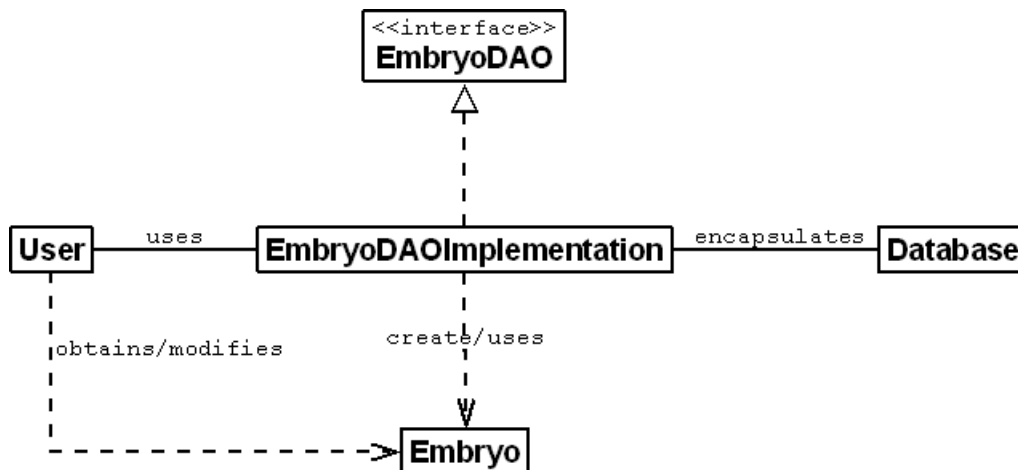


Fig. 28 Visualization of the DAO pattern

The DAO pattern is combined with a **factory pattern**:

The specific implementations for the DAO interfaces are created and managed by a factory class. This allows the change of the implementation for a DAO without affecting the rest of the application. [38] [39]

Example for the factory class

```
public class DAOFactory
{
    public ImageDAO getImageDAO()
    {
        return new ImageDAOHibernate()Impl;
    }

    public EmbryoDAO getEmbryoDAO()
    {
        return new EmbryoDAOMySQLImpl();
    }

    public ResearcherDAO getResearcherDAO()
    {
        return new ResearcherDAOOracleImpl;
    }

    public InstitutionDAO getInstitutionDAO()
    {
        return new InstitutionDAOPostrgeSQLImpl;
    }
}
```

Object-Relational Mapping (ORM)

Java is an object based programming language. As objects contain data, they must be stored in a database. Therefore the data fields of an object have to be mapped to the corresponding data fields of a table of a relational database.

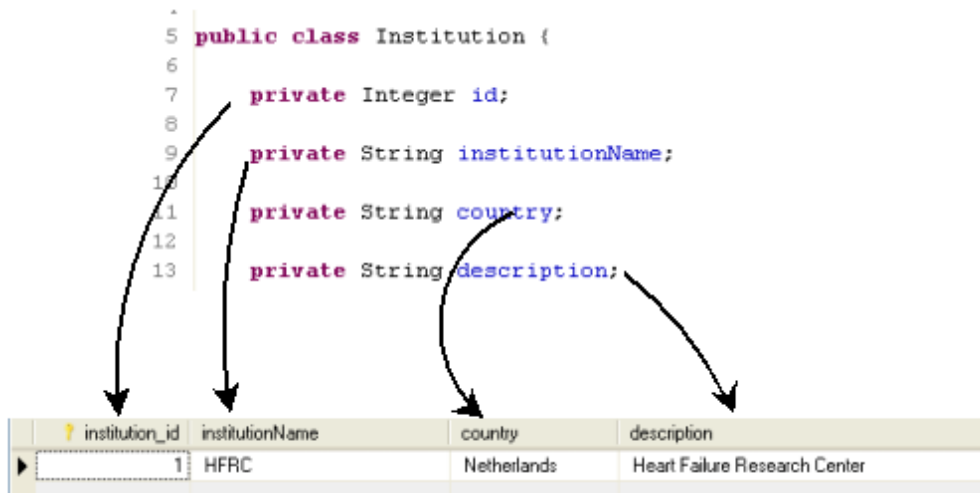


Fig. 29 Illustration of ORM

Persistence framework

An ORM is implemented in an object-based application to guarantee the persistence of objects. This makes a persistence framework necessary. This software manages the storage of objects including the conversion of JAVA data types to SQL data types, vice versa the data retrieval by generating stored data to objects.

2. METHODS

This chapter describes the methods used to achieve the specified requirements.

Entity-relationship-model

The entity-relationship (E/R)-model identifies objects of the real world, their attributes, and relationships between them. This model can be converted into a database scheme.

Definitions

Entities are discrete objects deduced from reality and are represented as rectangles in the E/R-diagram.

A **relationship** models the type of connection between identified entities. They are represented as diamonds, connected by lines to each of the entities in the relationship.

Three types of relationship can be distinguished:

One-to-one (1:1) relationship: every object of the left entity is connected to exactly one object of the right one.

One-to-m (1:n) relationship: one object of the left entity is connected to many objects of the right one.

N-to-m (n:m) relationship: many objects of the left entity are connected to many objects of the right one.

Attributes describe the characteristics and properties of the identified entities and relationships. They are represented by ellipses.

A **primary key** is a unique attribute of an entity and enables to identify a data record distinctively. It is visualized by underlining the attribute.

The **E/R-diagram** is the visualization of the E/R model. It provides a brief overview of the identified entities and describes the kind of relationship between them.

Generalization is used to structure entities by introducing a so-called “**is-a relationship**” according to inheritance. The subtype takes over all attributes of the supertype. So the subtype specifies the supertype and, vice versa the supertype generalizes the subtype.

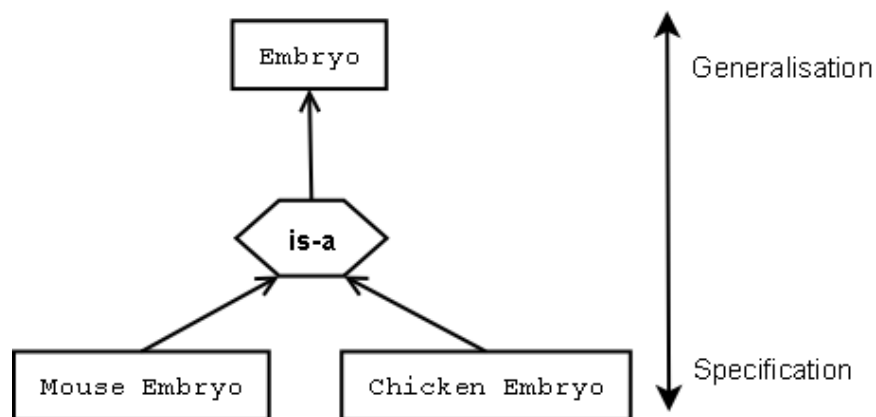


Fig. 30 Example for the generalization

Opposite to generalisation **aggregation** unites entities, which form together a supertype. Therefore a “**part-of relationship**” is used.

Transformation of the E/R-model into a relational database scheme

As it has already been mentioned in chapter 1.3 a relational database is built up of tables, which map relations. Thus the entities and relationships of the E/R-model have to be transformed into relations. The attributes are typified and transformed into data fields of a table.

3. RESULTS

This chapter sums up the results of the applied methods.

E/R-Model

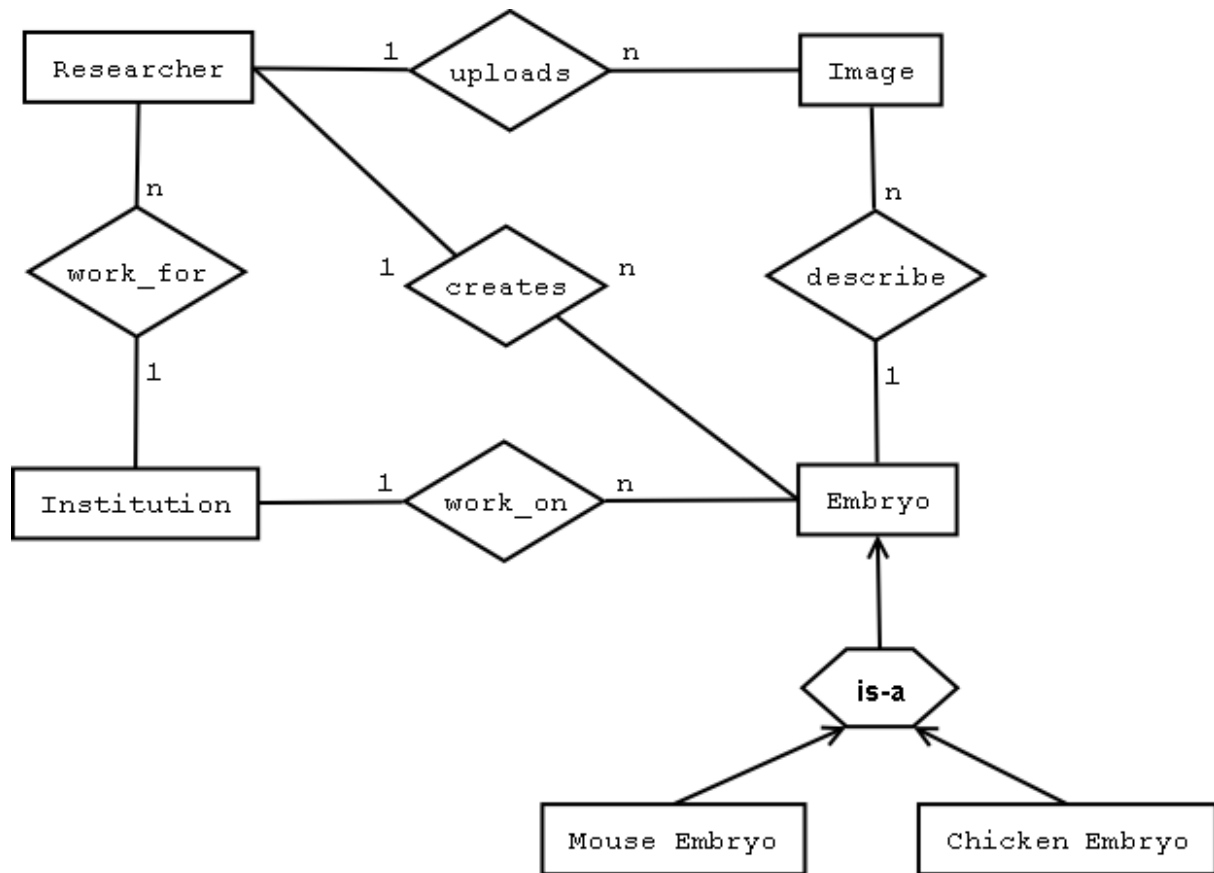


Fig. 31 The E/R-diagram

In this diagram the attributes are not included to allow a better overview. For the attributes please refer to the description of the entities in the following.

Entities

The **embryo** is the main entity and serves as a template for all the different species. It contains several images and so all common attributes of the images can also be stored in the embryo table.

Table 6 Attributes of the embryo

embryoid	Unique internal identifier.
researcherid	Unique identifier of the related researcher.
institutionid	Unique identifier of the related institution.
species	The species of the embryo.
ed	Age of the embryo in days.
treatment	Description of experimental treatment.
description	Optional description of the embryo.
slices	Number of slices of the embryo.
institutionembryoid	The internal id of an institution of an embryo.
thickness	The thickness of the section. (μm)

Both, the **mouse embryo** and **chicken embryo** are a specialization of the general embryo and inherit all its attributes.

Table 7 Attributes of the mouse embryo

embryoid	Unique internal identifier.
strain	Strain of the mouse.
theilerstage	Age according to Theiler.
transgeneid	0 = wildtype, 1..x = description of transgenes

Table 8 Attributes of the chicken embryo

embryoid	Unique internal identifier.
hamburgerhamilton	Age according to Hamburger Hamilton converted to an integer.

The **images** are uploaded by researchers and contain the gene expression information. In this table all the information specific for images is stored.

Table 9 Attributes of the image

imageid	Unique internal identifier.
embryoid	Unique identifier of the related embryo.
researcherid	Unique identifier of the related researcher.
stain	Name of the used staining method.
pubmedid	The internal identifier of the article at PubMed.
position	The position of the slice in an embryo.
magnification	$\mu\text{m}/\text{pixel}$
coordinates	Output of TRACTS consisting of 12 values, which are stored separately.
gene	Visualized gene.
qualityconfirmed	1: quality of the image is sufficient; 0: quality of the image is not sufficient
visibility	This field states, if an image should be visible for public or only for the submitting researcher. 0: private (Researcher); 1: private (Institution); 2: public
checked	An image needs to be checked, if it meets the quality requirements. 1: checked; 0: not checked
Path	Path to the image in the file system.
uploaddate	The date of the upload of an image.

The **researchers** work for an institution, create embryos, upload images and add these images to an embryo.

Table 10 Attributes of the researcher

reseacherid	Unique internal identifier.
institutionid	Unique identifier of the related institution.
firstname	First name of a researcher.
lastname	Last name of a researcher.
lastlogin	Date of the last login of a researcher.
password	Password of a researcher for the login. Stored "MD5" encrypted.
email	Email address of a researcher (also used for the login).
confirmed	A researcher needs to be confirmed by an employee of HFRC. 1: confirmed; 0: not confirmed

These **institutions** could be universities or laboratories researching gene expression patterns of embryos.

Table 11 Attributes of the institution

institutionid	Unique internal identifier.
institutionname	Name of the Institution.
country	Country where the institution is located.
description	Information about the institution.
confirmed	An institution needs to be confirmed by an employee of HFRC. 1: confirmed; 0: not confirmed

Is-a relationship of the embryo

The relationship between an embryo and its subtypes is used to minimize the amount of stored data and to guarantee the possibility of extendibility in a JAVA based application. The “is-a” relationship is implemented by two tables, one for the general embryo (supertype) and one for the specific subtype. These tables are connected by using the primary key of the supertype embryo as identifier of the subtype. In Hibernate this relationship is implemented as “joined-subclass”.

4. SOFTWARE SPECIFICATION

The DBMS and the persistence framework, which have been chosen, will be specified in this chapter:

DBMS: MySQL

MySQL is a relational DBMS (rDBMS) and was developed by “Sun Microsystems”. It is distributed under the “General Public License” (GPL), supporting widely the SQL-3 standard and is optimised to run on a web server with many read accesses. The communication with Java applications is guaranteed by a “JAVA database connectivity” (JDBC)-driver provided by MySQL.

Persistence Framework: Hibernate

The Hibernate framework is open source and at the moment one of the most common persistence frameworks. It guarantees the persistence of objects, maps data fields from JAVA-entity-classes to the fields of a table of a relational DBMS (ORM) and converts Java data types to SQL data types.

The Hibernate framework supports most DBMS and can be integrated in Java applications or servlets including association, inheritance, polymorphism, composition and collections.

ORM in Hibernate

In Hibernate the ORM is stated in an XML-based Hibernate mapping (HBM) file. This file contains all JAVA-entity classes ¹ with all their attributes ² and a definition ³ of the primary key. The relationships between entity-classes are also defined here.

In case that an object is stored, all the referenced objects are stored too. These relationships are represented as “JAVA-collections” in the application.

Mapping for the institution class

```
<class name="Institution" table="institution"> 1
  <id name="id" column="institution_id"> 3
    <generator class="increment"/>
  </id>
  <property name="institutionName" type="string" not-null="true"/> 2
  <property name="country" type="string" not-null="true"/>
  <property name="description" type="string" not-null="true"/>
  <property name="confirmed" type="boolean" not-null="true"/>
  <bag name="researchers" inverse="true" cascade="all-delete-orphan">
    <key column="institution_id"/>
    <one-to-many class="Researcher"/>
  </bag>
</class>
```

The Hibernate Query Language (HQL)

Another important advantage of Hibernate is the usage of a proprietary query language “HQL” instead of SQL. The HQL queries are translated into the specific SQL dialect, which allows keeping the implementation of the queries independent from the DBMS. Furthermore Hibernate generates the database scheme and provides data query and retrieval facilities.

“**Dirty Checking**” is another feature storing changed data fields of an object in the database.

Most important Hibernate classes

org.hibernate.SessionFactory

This class loads the mapping and the configuration. Per database one sessionFactory is required, which is implemented as singleton.

org.hibernate.Session

The session objects provide the JAVA-application with database with insert-, update-, delete- and query-operations to access the database.

Lifecycle of objects

The objects controlled by Hibernate can be in one of the three following states:

1. Transient

A transient object is not stored in the database. It is either newly instantiated and has not yet been stored or deleted from the database.

2. Persistent

A persistent object is in the database and associated with an active session.

3. Detached

A detached object is in the database, but not associated with an active session. It is not guaranteed that the database and the state of the object are synchronal. [39] [40]

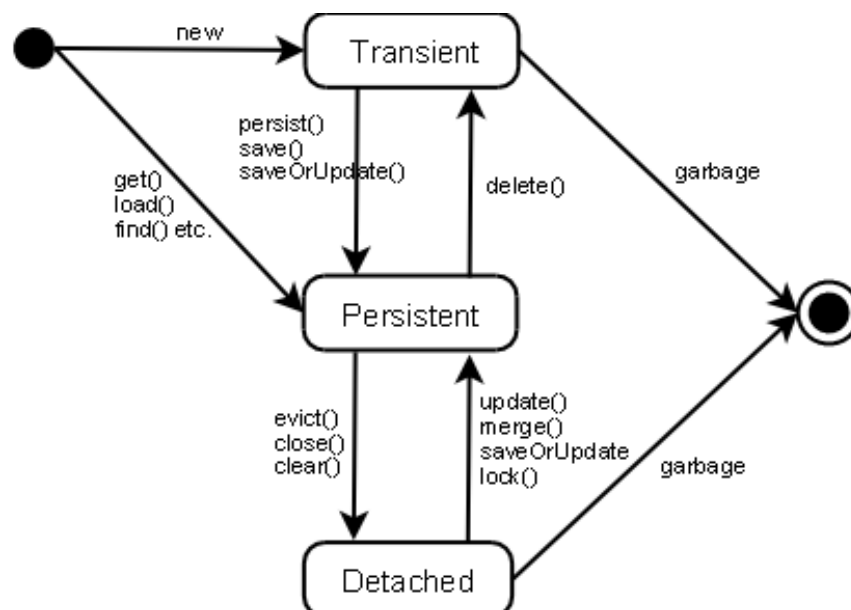


Fig. 32 Visualization of the object lifecycle

Description of the database layer

The database layer has the purpose that the business layer is independent from the DBMS. In this project two levels of independence have been achieved:

1. The business layer accesses the database over the DAO interfaces. The implementation of a DAO interface can easily be changed in a factory class without affecting the business layer.
2. Hibernate uses a proprietary query language, which is later translated in the specific SQL dialect of the DBMS. If the DBMS is replaced, the queries do not need to be changed.

Business objects represent the entities in an object-based application. As required by Hibernate the business objects only have “getter- and setter-methods” for their attributes and an empty constructor. These data objects, which are all mapped in the HBM file are equivalent to the tables of the database.

The relationships between these classes are implemented as bidirectional “Hibernate bags”, which means that the class of the supertype keeps its subtypes in a JAVA-collection and the subtype has a reference to the supertype object [41].

The HibernateUtil class

This helper class is implemented as singleton. It obtains the Hibernate-sessionFactory object, which initializes the configuration of Hibernate and manages Hibernate-sessions.

For its configuration Hibernate needs, as already described above, a HBM file and a specific configuration file, which configures the connection to the DBMS and other properties.

Implementation of the DAO pattern

For each separate entity an interface which contains the most basic methods to access the database: save, update and delete. For all of these interfaces an implementation for Hibernate has been programmed.

The factory pattern is realized by the **DAOFactory** class that creates and manages implementations of the DAOs.

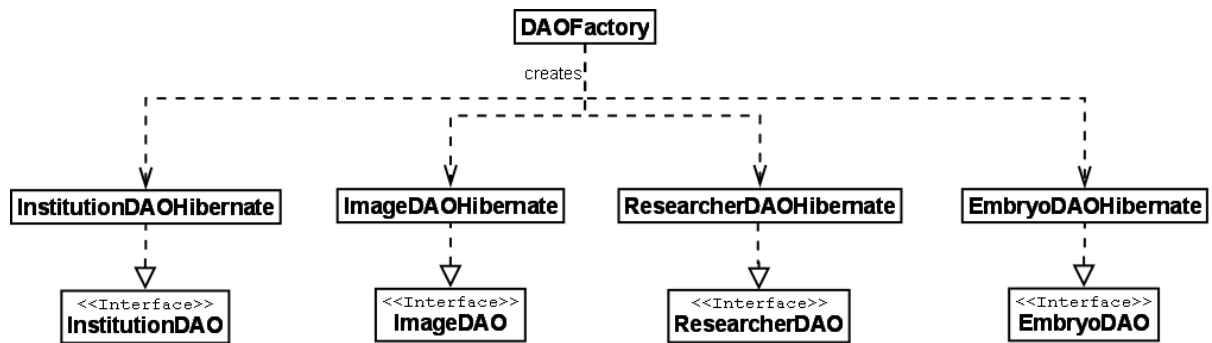


Fig. 33 Visualization of the DAO pattern

Storage of images

There are two possibilities of storing images:

- Storing an image in the file system
- Storing an image as binary large object (BLOB) in the database

Superficially considered, storing the images in the database seems to be the better solution. Nevertheless, the file system offers more advantages:

1. If the database breaks down, the images, stored in the file system will not be affected (but it is arguable whether the images are useful without the according information in the database).
2. It is possible to access the images from applications without a database connection, like TRACTS.
3. Large binary files stored in the database could make the search slower.

The only problem that has to be considered is the synchronisation of the database and the file system.

To avoid this issue the uploaded images are at first renamed, converted to the Portable Network Graphic (PNG)² file format and then stored to the hard drive. Only the location of an image in the file system is stored in the database. The location is determined by the name of the institution and by the primary ID of the embryo. The primary identifier of an image in the database serves as its file name.

Example for the storage of an image

The Heart Failure Research Center (HFRC) uploads for embryo 47 an image, which has the primary ID 1049 in the database.

Directory structure in the file system:

\HFRC\47\1049.png

² PNG is a lossless image compression file format. It is acknowledged by the World Wide Web Consortium and supported by most web browsers.

5. DISCUSSION

The prototype

For the prototype the following database scheme was implemented:

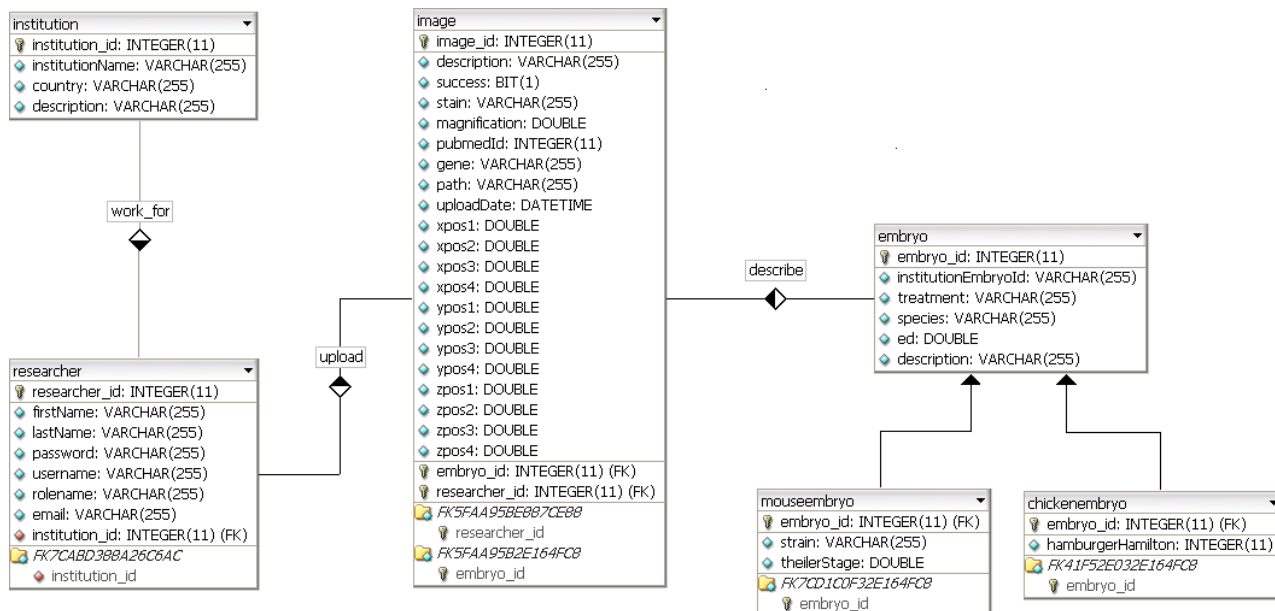


Fig. 34 Implementational database diagram of the prototype

As it can be seen in figure 34, not all the attributes described in the E/R-model are included.

Embryo	MouseEmbryo	Image	Researcher
slices	transgeneid	position	lastlogin
		qualityconfimed	confirmed
		visibility	
		checked	

The relationship between the institution and the image as well as the relationship between the researcher and the embryo are also missing.

Future perspectives and discussion

Interviewing experts and afterwards deducing the database scheme using an E/R-model proved to be a good approach. Specific queries could not be found during the interviews. Most researchers were not sure what they wanted to search in the database since TRACTS is still in development. Their uncertainty was supported by the unknown number of users, further the quantity and quality of the submitted data. Hence, it was not possible to configure Hibernate in detail or to create more complicated queries. So only the standard configuration and basic queries were implemented. The storage of images is working fine but during practical application unpredictable issues could occur.

Using the DAO pattern for the architecture was a good decision. This design pattern was easy to implement and already guaranteed in an early state of development the communication with the logical layer.

The usage of Hibernate as persistence framework was at the beginning difficult and took a longer time than estimated. Once working, the advantages of Hibernate compensated the efforts of its implementation: it was easy to extend and maintain. The combination of Hibernate and MySQL proved to be a good solution and delivered a good performance in practice.

Summarizing the development of the database layer was more work than planned. Nevertheless the database is in principle fully working and almost the complete database scheme could be implemented.

IV. COMMON PART

1. COMMON DISCUSSION

Significance of the project

The visualization of the results of TRACTS shows the significance of this project. The main goal was to prove that it is possible to visualize a 3D image of the Matlab program in a web browser. The researchers of the AMC are now able to think about further development of a leading gene expression database in the field of embryonic heart research. It is moreover written in the conclusions of the interviews that there is no online application available that provides such a detailed illustration of an embryonic heart. Now the first step was made to change this fact in the field of embryonic research.

Generalization of the project

The used architecture of the database and the web interface discussed in the particular chapters offer the ability to develop a secure web application, which is efficient and extendable and can be adapted to fit further needs.

Requirement analysis interviews

The interviews held with several researchers helped the authors to get a good insight in the field of embryonic heart research and databases, which are present at the moment. On the other side it cost a lot of time interviewing these persons and analyzing the results, which could be used to spend on implementation. Most of the information gathered has already been present before the interviews were held, and maybe it was not the best idea for the prototype to interview later users of the system. A better solution would have been to achieve the communication with TRACTS and a rough structure of the program to hold the interviews afterwards.

Future work of the AMC

In the future the AMC is now able to take this project one step further to develop a competitive online gene expression database with a focus on the embryonic heart. On account of the successful embedding of TRACTS into a web application is that the HFRC can certainly improve TRACTS and be sure that the application is still runnable.

V.APPENDIX

1. DATA DICTIONARY

A data dictionary is an index with descriptions of all data used in a system, which is stored in the database. It is defined as “centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format” [42].

The data dictionary has been evolved and complemented continuously during the interviews.

Table 12 Entity: Embryo

Embryo		The embryo is the main entity and serves as a template for all the different species. It contains several images and so all common attributes of the images can also be stored in the embryo table.		
Data Name	Description	Type	Additional Type Information	Default Value
Embryoid	Unique internal identifier.	INTEGER	PK, AI, NN	
researcherid *	Unique identifier of the related researcher.	INTEGER	FK, NN	
Institutionid *	Unique identifier of the related institution.	INTEGER	FK, NN	
Species	The species of the embryo.	STRING	NN	
Ed	Age of the embryo in days.	DOUBLE	NN	
Treatment	Description of experimental treatment.	VARCHAR		null
description	Optional description of the embryo.	VARCHAR		null
slices *	Number of slices of the embryo.	INTEGER		null
institutionembryoid	The internal id of an institution of an embryo.	VARCHAR	NN	

Thickness *	The thickness of the section. (µm)	DOUBLE		null
-------------	---------------------------------------	--------	--	------

Table 13 Entity: Mouse Embryo

MouseEmbryo	The mouse embryo is a specialization of the general embryo and inherits all its attributes.			
Data Name	Description	Type	Additional Type Information	Default Value
Embryoid	Unique internal identifier.	INTEGER	FK, NN	
Strain	Strain of the mouse.	VARCHAR	NN	
Theilerstage	Age according to Theiler.	Double	1-25	null
transgeneid *	0 = wildtype, 1..x = description of transgenes	INTEGER	NN	0

Table 14 Entity: Chicken Embryo

ChickenEmbryo	The chicken embryo is a specialization of the general embryo and inherits all its attributes.			
Data Name	Description	Type	Additional Type Information	Default Value
Embryoid	Unique internal identifier.	INTEGER	FK, NN	
hamburgerhamilton	Age according to Hamburger Hamilton converted to an integer. **	INTEGER	1-135	null

Table 15 Entity: Image

Image	The images are uploaded by researchers and contain the gene expression information. In this table all the information specific for images is stored.			
Data Name	Description	Type	Additional Type Information	Default Value
Imageid	Unique internal identifier.	INTEGER	PK, AI, NN	
Embryoid	Unique identifier of the	INTEGER	FK, NN	

	related embryo.			
researcherid	Unique identifier of the related researcher.	INTEGER	FK, NN	
Stain	Name of the used staining method.	VARCHAR	NN	
Pubmedid	The internal identifier of the article at PubMed.	LONG		null
position *	The position of the slice in an embryo.	INTEGER		null
Magnification	µm/pixel	DOUBLE	NN	null
Coordinates	Output of TRACTS consisting of 12 values which are seperately stored	DOUBLE		null
Gene	Visualized gene.	VARCHAR	NN	
qualityconfirmed *	1: quality of the image is sufficient; 0: quality of the image is not sufficient	BIT	NN	0
visibility *	This field states states, if an image should be visible for public or only for the submitting researcher. 0: private (Researcher); 1: private (Institution); 2: public	INTEGER	NN, 0-2	0
checked *	An image needs to be checked, if it meets the quality requirments. 1: checked; 0: not checked	BIT	NN	0
Path	Path to the image in the file system.	String		Null
Uploaddate	The date of the upload of an image.	DATE	NN	

Table 16 Entity: Researcher

Researcher		The researchers work for an institution to create embryos, upload images and assign these images to an embryo.		
Data Name	Description	Type	Additional Type Information	Default Value
Reseacherid	Unique internal identifier.	INTEGER	PK, AI, NN	
institutionid	Unique identifier of the related institution.	INTEGER	FK, NN	
Firstname	First name of a researcher.	VARCHAR	NN	
Lastname	Family name of a researcher.	VARCHAR	NN	
lastlogin *	Date of the last login of a researcher.	DATE	NN	
Password	Password of a reasearcher for the login. Stored "MD-5" encrypted.	VARCHAR	NN	
Email	Email adress of a researcher, which is also used for the login.	VARCHAR	NN	
confirmed *	A researcher needs to be confirmed by an employee of HFRC. 1: confirmed; 0: not confirmed	BIT	NN	0

AI = auto incremental

NN = not null

PK = primary key

FK = foreign key

* Future work and beyond of the scope of this project

** Algorithm for the conversion of a Hamburger Hamilton stage to an integer:

```
public void setHamburgerHamilton (String sign, Integer hh)
{
    if (hh > 0 && hh < 46 && (sign == null || sign.contentEquals("+") || sign.contentEquals("-")))
    {
        if (sign == null) hamburgerHamilton = hh * 3 - 1;
        else if (sign.contentEquals("+")) hamburgerHamilton = hh * 3;
        else if (sign.contentEquals("-")) hamburgerHamilton = hh * 3 - 2;
    }
}
```

** Algorithm for the conversion of an integer back to a Hamburger Hamilton stage:

```
public String getHHasString()
{
    int hh = hamburgerHamilton.intValue();
    String hhString = null;

    if (hh % 3 == 1) hhString = "-" + String.valueOf(hh/3+1);
    else if (hh % 3 == 2) hhString = String.valueOf(hh/3+1);
    else if (hh % 3 == 0) hhString = "+" + String.valueOf(hh/3);

    return hhString;
}
```

2. QUESTIONNAIRE

General questions

1. What is your profession at the AMC? What is your current research project?
2. Why and for what do you use genePaint or EMAP?
3. How often do you access these sites?
4. What kind of data are you submitting?
5. Do you also use other databases?

Questions concerning genePaint and EMAP

6. What is the main difference between genePaint and EMAP? Why are they not put together?
7. It seems that there are many existing projects similar to the Amsterdam mouse heart atlas project. Why do you think this project is really needed? What will be the advantage of the Amsterdam project?
8. What kind of information do you retrieve from genePaint and EMAP and what is the benefit for your research? Could you work without these databases?
9. If you could, what would you change on EMAP or genePaint?

Questions concerning the web interface

10. Which functionalities do you have in mind for the web interface?
11. Who should be able to access the database and submit data? Should everybody have the possibility to derive information from the database?
(Login)

12. GenePaint offers the possibility to request gene expression pattern data, if they are provided with cDNA. Maybe the HFRC is going to request data. Is there any possibility to acquire gene expression patterns at the AMC or does gene expression data, needed for research, have to be ordered from somewhere else?
13. Is it planned that there is an anatomical annotation or even ontology in Amsterdam?
14. Are genes going to be annotated by experts in Amsterdam like by the editors on EMAP?
15. What kind of data is needed for the research at the HFRC and how is the gene expression data produced? Should an explanation of how data is produced be available on the web interface like on genePaint?
16. What is the benefit of the Amsterdam database for the research on the embryonic heart?
17. Which gene symbols are the most common? Which gene annotation should we use for our project?
18. Which database queries should be supported?
19. Is there any specific data to be stored for the chicken?
20. On EMAP several possibilities of displaying are available. Which one of these is most suitable for our project?

A list of product data has been evolved, corrected and complemented continuously by the interviewed scientists.

VI. LIST OF FIGURES

Fig. 1	Simplified overview of the project.....	16
Fig. 2	Schematic illustration of the development of an embryonic heart of a higher vertebrate. a = atrium, v = ventricle, l = left, r = right. The other labels are not relevant for this paper.	17
Fig. 3	From left to right: An image of a histological section, a resized thresholded binary image, its contour, and its distance transformed image. a = atrium, v = ventricle, l = left, r = right. Note that at this stage the systematic and pulmonary circulation are not yet separated.	19
Fig. 4	Result of a text query for the gene “Pax6”	22
Fig. 5	Result of a sequence homology search for the protein “Fbox 2”	22
Fig. 6	Example of an embryo map: Mouse E 14.5 with distance from midline: 0.2 mm, left slide. Annotated regions are marked with a red pointer. Yellow lines are borderlines between the different brain regions.....	24
Fig. 7	3D Model with text anatomy descriptions	27
Fig. 8	Example of a text search.....	28
Fig. 9	Result of a text search.....	28
Fig. 10	Example for spatial search	29
Fig. 11	Result of a spatial search	29
Fig. 12	Use case diagram	42
Fig. 13	Client-server architecture	48
Fig. 14	Three Tier architecture: The software used in the thesis is shown in parentheses next to the different layers.....	48
Fig. 15	Form Based Authentication	56
Fig. 16	MVC2 pattern	58
Fig. 17	Sequence diagram of the web application architecture	59
Fig. 18	Architecture of the web interface.....	60
Fig. 19	Navigation diagram of the current web interface	61
Fig. 20	Registration process.....	62
Fig. 21	Registration procedure via MD5 encryption	63

Fig. 22 Login interface.....	63
Fig. 23 Creation of a mouse and a chicken embryo	64
Fig. 24 Embryo selection.....	65
Fig. 25 Upload of an image	65
Fig. 26 Visualization of the result	67
Fig. 27 Future navigation of the web interface	69
Fig. 28 Visualization of the DAO pattern	73
Fig. 29 Illustration of ORM.....	75
Fig. 30 Example for the generalization	77
Fig. 31 The E/R-diagram	78
Fig. 32 Visualization of the object lifecycle	85
Fig. 33 Visualization of the DAO pattern	87
Fig. 34 Implementational database diagram of the prototype.....	89

VII. LIST OF TABLES

Table 1	Interviewed researchers	34
Table 2	PD1: Information describing submitted images	40
Table 3	PD2: The image and its attributes	40
Table 4	PD3: Information describing submitting persons	41
Table 5	The accomplished goals of the specified requirements.....	68
Table 6	Attributes of the embryo	79
Table 7	Attributes of the mouse embryo.....	79
Table 8	Attributes of the chicken embryo	80
Table 9	Attributes of the image	80
Table 10	Attributes of the researcher	81
Table 11	Attributes of the institution	81
Table 12	Entity: Embryo	92
Table 13	Entity: Mouse Embryo	93
Table 14	Entity: Chicken Embryo	93
Table 15	Entity: Image	93
Table 16	Entity: Researcher.....	95

VIII. REFERENCES

- [1] Jan M. Ruijter, Alexandre T. Soufan, Jaco Hagoort, and Antoon F.M. Moorman; Molecular imaging of the embryonic heart: Fables and facts on 3D imaging of gene expression patterns. *Birth Defects Research (Part C)* 72:224–240 (2004)
- [2] American Heart Association. Heart Disease and Stroke Statistics — 2008 Update. Dallas, Texas. *Am Heart J.* 2004;147: 425–439.
http://www.americanheart.org/downloadable/heart/1200078608862HS_Stats%202008.final.pdf
- [3] Weninger W.J., Mohun T.J. High-resolution episcopic microscopy: a rapid technique for high detailed 3D analysis of gene activity in the context of tissue architecture and morphology. *Anat. Embryol* 2006; 211: 213–221
- [4] Alexandre T. Soufan, Jan M. Ruijter, Maurice J. B. van den Hoff, Piet A. J. de Boer, Jaco Hagoort, and Antoon F. M. Moorman. Three-dimensional reconstruction of gene expression patterns during cardiac development. *Physiol Genomics* 2003; 13(3): 187–195
- [5] Bouke A. de Boer, Jan M. Ruijter, Frans P.J.M. Voorbraak. Towards the automatic registration of histological sections into a 3D reference model. BNAIC 07 - Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence 2007, Utrecht: 5-6 Nov. 2007; 41-48
- [6] Indirect immunohistochemistry (IHC) on tissue sections Standard Operating Procedures, “AELW_CE06(05).doc”, Lab Info - AMC Amsterdam C. de Gier-de Vries, S. van der Velden 2008;
- [7] In situ hybridisation using non radioactive labeled (DIG) RNA probes (NR-ISH). Standard Operating Procedures, “AELW_DE03.doc(05)”, Lab Info - AMC Amsterdam. C. de Gier-de Vries 2008;
- [8] Whole mount insitu hybridisation using non radioactive labeled (DIG) RNA probes (NR-ISH). Standard Operating Procedures, “AELW_DE02(04).doc”, Lab Info - AMC Amsterdam. C. de Gier-de Vries 2005;
- [9] Sharpe, J., et al. Optical projection tomography as a tool for 3D microscopy and gene expression studies 2002; *Science* 296.5567: 541-45.
- [10] Axel Visel, Christina Thaller, and Gregor Eichele. GenePaint.org: an atlas of gene expression patterns in the mouse embryo 2003;
- [11] Jeffrey H. Christiansen, Yiya Yang, Shanmugasundaram Venkataraman, Lorna Richardson, Peter Stevenson, Nicholas Burton, Richard A. Baldock, and Duncan R.

Davidson. EMAGE: a spatial database of gene expression patterns during mouse embryo development 2005;

[12] Wikipedia. Pub Med. cited 06.06.2008. URL: <http://en.wikipedia.org/wiki/Pubmed>

[13] Johannes Aschaber; Software Engineering; university of medical informatics and techniques Hall in Tirol. institute of biomedical engineering; winter term 2006/07;

[14] Wikipedia. Use case diagram. cited 30.03.2008. URL:

http://de.wikipedia.org/wiki/Use_case

[15] Wikipedia. Hyper Text Markup Language. cited 05.06.2008. URL:

<http://en.wikipedia.org/wiki/Html>.

[16] Wikipedia. World Wide Web. cited 05.06.2008. URL:

<http://en.wikipedia.org/wiki/Www>.

[17] Wikipedia. Cascading Style Sheets. cited 05.06.2008. URL:

<http://en.wikipedia.org/wiki/CSS>.

[18] Wikipedia. Web Server. cited 21.07.2008. URL:

http://en.wikipedia.org/wiki/Web_Server.

[19] Wikipedia. Apache Tomcat. cited 21.07.2008. URL:

http://en.wikipedia.org/wiki/Apache_Tomcat.

[20] Wikipedia. Client-Server-Architecture. cited 21.07.2008. URL:

<http://de.wikipedia.org/wiki/Client-Server-Architektur>.

[21] Wikipedia. Three Tier Architecture. cited 21.07.2008. URL:

http://en.wikipedia.org/wiki/Three_tier.

[22] Wikipedia. Dynamic Web Sites. cited 21.07.2008. URL:

http://en.wikipedia.org/wiki/Dynamic_web_page.

[23] Wikipedia. Java Script. cited 21.07.2008. URL:

<http://en.wikipedia.org/wiki/JavaScript>.

[24] Official Website of PHP. cited 21.07.2008. URL: <http://www.php.net>

[25] Java Servlet Programmierung, Jason Hunter/ William Crawford, 1.Auflage 2002, p. 2 - 21]

[26] Official Website of DIA. cited 08.08.2008. URL: <http://live.gnome.org/Dia>

[27] Official Website of ArgoUML. cited 08.08.2008. URL: <http://argouml.tigris.org/>

[28] Official Website of Sun Microsystems. cited 18.12.2007. URL:

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/Security5.html>

[29] Official Website of Sun Microsystems. cited 18.12.2007. URL:

<http://java.sun.com/javaee/5/docs/tutorial/doc/bncbe.html>

- [30] Wikipedia MD5 Encryption. cited 25.07.2008. URL:
<http://de.wikipedia.org/wiki/Md5>
- [31] Official IBM Website. cited 08.08.2008. URL:
<http://www.ibm.com/developerworks/library/j-struts/>
- [32] Wikipedia MVC. cited 07.08.2008. URL:
http://de.wikipedia.org/wiki/Model_View_Controller
- [33] Skript zur Vorlesung Informationssysteme und Datenbanken 2006/2007, Claudia Plant
- [34] Wikipedia Database. cited 25.5.2008. URL: <http://en.wikipedia.org/wiki/Database>
- [35] Wikipedia Database Management System. cited 25.05.2008. URL:
<http://en.wikipedia.org/wiki/Dbms>
- [36] Datenbanksysteme (Database Systems), A. Kemper/ A. Eickler, 6. Auflage 2006,p. 20,21
- [37] Wikipedia Data model. cited 25.05.2008. URL:
http://en.wikipedia.org/wiki/Data_model
- [38] Dao pattern combined with Hibernate. cited 24.05.2008 URL:
<http://powerdream5.wordpress.com/2007/10/13/dao-pattern-for-hibernate/>
- [39] Data Access Objects. cited 28.05.2008. URL:
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
- [40] Official Hibernate Website. cited 21.05.2008. URL: <http://www.hibernate.org/>
- [41] Wikipedia Hibernate. cited 21.05.2008. URL:
[http://en.wikipedia.org/wiki/Hibernate_%28Java%](http://en.wikipedia.org/wiki/Hibernate_%28Java%28)
- [42] Wikipedia Data dictionary. cited 22.05.2008. URL:
http://en.wikipedia.org/wiki/Data_dictionary

Danksagung/ Acknowledgement

An erster Stelle bedanken wir uns bei unseren Eltern, die uns das Studium finanziert und uns bislang in jeglicher Hinsicht tatkräftigt unterstützt haben. Ein besonderer Dank gilt Christine Weidenholzer, für die zeitaufwändigen Korrekturarbeiten. Unseren Freundinnen danken wir für ihre Geduld und, dass sie uns jederzeit mit Rat und Tat beiseite standen.

Nicht zuletzt danken wir noch Herrn Klemens Woertz, der am gesamten Projekt maßgeblich beteiligt war. Ein herzliches Dankeschön auch an Frau Professor Elske Ammenwerth, welche es uns ermöglicht hat, die Arbeit an der UMIT fertigzustellen, ferner einen reibungslosen Ablauf ermöglicht hat.

Special thanks to Frans, Bouke, Jan, and all the other researchers of the HFRC, who helped us to write this thesis. They never hesitated to correct and improve our work. Furthermore they helped us to get a good insight in the field of embryonic heart research and also showed us the cultural diversity of Utrecht.