Consolidated requirements and specifications for data access

Project Number:	FP6-2005-IST-026996
Deliverable id:	D5.1
Deliverable name:	Consolidated requirements and specifications for data access
Date:	1 February 2007

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	ACGT
Project Full Name:	Advancing Clinico-Genomic Clinical Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery
Document id:	D5.1
Document name:	Consolidated requirements and specifications for data access
Document type (PU, INT, RE)	RE
Version:	1.01
Date:	01/02/2007
Editors: Organisation: Address:	Erwin Bonsma and Anca Bucur Philips Research High Tech Campus 5656 AE Eindhoven the Netherlands

Document type PU = public, INT = internal, RE = restricted

ABSTRACT:

This report comprises the first deliverable of WP5 of the ACGT project. A main goal of this work package is to provide seamless and interoperable data access to the distributed data sources that are relevant to each of the ACGT clinical trials by developing a set of compatible software modules and services based on web services. The main objective of this deliverable is to consolidate the user requirements defined in WP2 with respect to distributed data access. For this purpose we use the CAFCR architecting framework. This deliverable contains a chapter for each of the five CAFCR views: the Customer View, the Application View, the Functional View, the Conceptual View, and the Realization View. Using this structure, we not only document the requirements for data access in more detail, but also make first steps towards a solution. This deliverable documents what has been done so far in WP5, and concludes by outlining what is next.

KEYWORD LIST: Data access, requirements, clinical trials, bio-medicine, CAFCR

MODIFICATION CONTROL			
Version	Date	Status	Editor
0.90	17/11/2006	Draft	Erwin Bonsma
0.91	22/11/2006	Draft	Erwin Bonsma
0.92	24/11/2006	Draft	Erwin Bonsma
0.93	29/11/2006	Draft	Erwin Bonsma
0.99	12/01/2007	Pre-final	Erwin Bonsma
1.00	15/01/2007	“Final”	Erwin Bonsma
1.01	01/02/2007	Final	Erwin Bonsma

List of Contributors

- Norbert Graf. University of Saarland
- Christine Desmedt, Jules Bordet Institute
- Gabriele Weiler, Fraunhofer IBMT
- Manolis Tsiknakis, FORTH
- Jasper van Leeuwen, Philips Research
- Jari Häkkinen, Lund University

List of Reviewers

- Norbert Graf. University of Saarland
- Stefan Rüping, Fraunhofer IAIS
- Stefan Kiefer, Fraunhofer IBMT
- Richard Vdovjak, Philips Research
- Fons de Lange, Philips Research

Contents

Executive Summary

ACGT is an Integrated Project (IP) funded in the 6th Framework Program of the European Commission under the Action Line “Integrated biomedical information for better health”. The high level objective of the Action Line is the development of methods and systems for improved medical knowledge discovery and understanding through integration of biomedical information (e.g. using modelling, visualization, data mining and grid technologies). Biomedical data and information to be considered include not only clinical information relating to tissues, organs or personal health-related information but also information at the level of molecules and cells, such as that acquired from genomics and proteomics research.

ACGT focuses on the domain of Cancer research, and its ultimate objective is the design, development and validation of an integrated Grid enabled technological platform in support of post-genomic, multi-centric Clinical Trials on Cancer. The driving motivation behind the project is our committed belief that the breadth and depth of information already available in the research community at large, present an enormous opportunity for improving our ability to reduce mortality from cancer, improve therapies and meet the demanding individualization of care needs.

The amount of information obtained from genetic and protein studies, clinical trials, and other research is growing rapidly. One of the main challenges in the biomedical field nowadays is to easily share and aggregate data, as there is not yet a unifying infrastructure or common technology standard that can be used. WP5 of the ACGT project addresses this challenge. The goal of this work package is to provide seamless and interoperable data access to the distributed data sources that are relevant to each of the ACGT clinical trials by developing a set of compatible software modules and services based on web services.

This document is the first deliverable of WP5. Its main objective is to consolidate the user requirements defined in WP2 with respect to distributed data access. For this purpose we use the CAFCR architecting framework. This framework consists of five views, which together are meant to bridge the gap between the needs, objectives and wishes of the user and their realization with available technology. The five CAFCR views are: the Customer View, the Application View, the Functional View, the Conceptual View, and the Realization View. In this deliverable, there is a chapter corresponding to each of the views.

The Customer View describes what the customer wants. Here we describe each of the clinical trials that are part of ACGT. Subsequently, the various stakeholders are listed, together with their drivers. From these, we derive several key drivers for data access, namely: distributed access to data, remote access and data entry, performance, flexibility, and interoperability.

In the Application View we consider how the users want to achieve their objectives, with a focus on data access. We describe for each of the clinical trials what data is collected, when it is collected, and how it is collected. The scenarios introduced in D2.1 that are most relevant to data access are also presented.

The Functional View describes what the system should achieve. For the data access wrapper services, this for a large part depends on how the data is stored and the queries that need to be supported. We therefore describe this for each of the clinical trials, and subsequently dive into more detail with respect to the data access requirements by using the scenarios. Various different types of data sources are identified in the Functional View, each with their own characteristics. The most important data sources are: relational databases, DICOM servers, public web databases, and files in various formats.

The Conceptual View is the first of the two views that describe the technical solution. It focuses on the concepts and sub-systems that are used, without yet specifying the

underlying implementation details. In providing access to medical data it is important that the privacy of patients is guaranteed. For this, anonymisation and pseudonymisation are important techniques. Furthermore, there may be a need to access data that resides inside a firewall. The architectural patterns that can be used here are introduced in this view, with the aim to facilitate subsequent discussion.

The focus of the Realization View is on the tools, standards and platforms that can be used to implement the solution outlined in the previous view. Web service technology is very suited for providing interoperable data access services. After a general introduction, we describe the most relevant web services standards and platforms, including OGSA-DAI and Soaplab. This is followed by a description of other relevant standards and technologies, such as DICOM. We.have also examined some of these technologies in more detail by way of hands-on technology explorations, which we have documented here.

We conclude by arguing that it is best to initially focus on data access to the trial-specific data as opposed to data in the public sequence databases. We furthermore intend to focus on data querying, as opposed to data entry. However, as the latter is important as well, we do propose to work on services for the creation and management of clinico genomic trials on cancer. In the short term, our effort will be focussed on finishing the OGSA-DAI wrapper services for access to DICOM data. Next to that, we identified two related research questions. Firstly, what query language should the data access services support? Secondly, how should the data access services describe the schema and content of the underlying databases? In addressing these questions, it is also important to take existing software and standards into account as much as possible.

1 Introduction

1.1 Project background

Recent and forthcoming developments in genomics and the increased importance of genetics in healthcare are already changing clinical care. Research on the molecular mechanisms of cell growth, apoptosis and differentiation has resulted in a better understanding of the nature of cancer cells. The genotypic knowledge of a cancer cell helps to identify the predisposition of the disease and can be used to develop therapies adapted to the genotype of a cancer patient. Medicine is becoming increasingly individualised.

The amount of information obtained from genetic and protein studies, clinical trials, and other research is growing rapidly. Despite that, there is not yet a unifying infrastructure or common standard for the technologies that cancer researchers use. There are for example no mechanisms for easily sharing and aggregating data. In response to these challenges, Biomedical Informatics is quickly evolving into a research field that encompasses the use of all kinds of biomedical information, from genetic and proteomic data to image and clinical data associated with various levels of the human body. This kind of integration and exploitation of the data and information requires a new synergetic approach that enables a bi-directional dialogue between these scientific disciplines and integration in terms of data, methods, technologies, tools and applications. While the goal is clear, the path is difficult to go, fraught with technical, scientific, clinical, legal, and ethical challenges. Many new tools for today's biomedical researcher have been developed to find the mechanism behind cancer, whereas legal and ethical issues are lagging behind.

The main goal of ACGT is to effectively fight against cancer. To achieve this goal ACGT has the following objectives:

- The ACGT project sees its mission to develop a Grid platform to support and stimulate further exchanges of both clinic and genetic information.
- ACGT intends to trigger the emergence of latent clinico-genomic synergies to ensure faster diagnosis and more efficient therapy.
- ACGT targets two major cancer diseases namely, breast cancer and paediatric Nephroblastoma presented by three (running) clinical trials.
- In addition, in-silico oncology trial scenarios will be run to assess the utility of tumour-growth simulation on both breast cancer and paediatric Nephroblastoma.

1.2 The ACGT environment

ACGT was set up to respond to the challenges arising from three global factors as mentioned above:

- Changing environment comprising a number of issues in all areas of life science.
- Changes in healthcare delivery comprising the move towards increasingly individualised medicine.
- Technology push in conjunction with Biomedical Informatics.

ACGT focuses on clinical trials on Cancer (Wilms tumour, Breast) and is based on the principles of open access (among trusted partners) and developing open source products. ACGT will provide a unified technological infrastructure to facilitate

- Integrated access to multi-level biomedical data;

- The development or re-use of open source analytical tools, accompanied with the appropriate meta-data allowing their discovery and orchestration into complex workflows.

ACGT brings together internationally recognised leaders in their respective fields, with the aim to deliver to the cancer research community an integrated clinico-genomic ICT environment enabled by a powerful Grid infrastructure. In achieving this objective ACGT has formulated a coherent, integrated work plan for the design, development, integration and validation of all technologically challenging areas of work. Namely:

- Grid: delivery of a European Biomedical Grid infrastructure offering seamless mediation services for sharing data and data-processing methods and tools, and advanced security;
- Integration: semantic, ontology based integration of clinical and genomic/proteomic data - taking into account standard clinical and genomic ontologies and metadata;
- Knowledge Discovery: delivery of data-mining Grid services in order to support and improve complex knowledge discovery processes;
- Legal and ethical issues: development and integration of technical solutions regarding data protection and secure personal data management in a European context.

The technological platform of ACGT will be validated in concrete setting of advanced clinical trials on Cancer. Pilot trials have been selected based on the presence of clear research objectives, raising the need to integrate data at all levels of the human being.

1.2.1 Position of WP5 within ACGT platform

In this document we consider the requirements from the point of view of WP5, which deals with distributed access to data, and make initial steps towards a technological solution. However, it is important to realise that the output of WP5 is not a complete system that can be demonstrated and used on its own. It needs to be integrated with the solutions emerging from the other ACGT work packages.

At the outset of the ACGT project, before the project officially started, part of architecture of the overall ACGT platform was already outlined. The main functional components that are required were identified, and were used to define the various technical work packages and their relations.

Figure 1 shows the ACGT integration architecture, as presented in the Description of Work. It shows how the wrapper services, to be provided by WP5, relate to the other functional components, to be provided by the other work packages. The purpose of the wrapper services is to provide uniform data access to heterogeneous data sources. It is one of the main outputs of WP5, upon which various other work packages directly or indirectly rely.

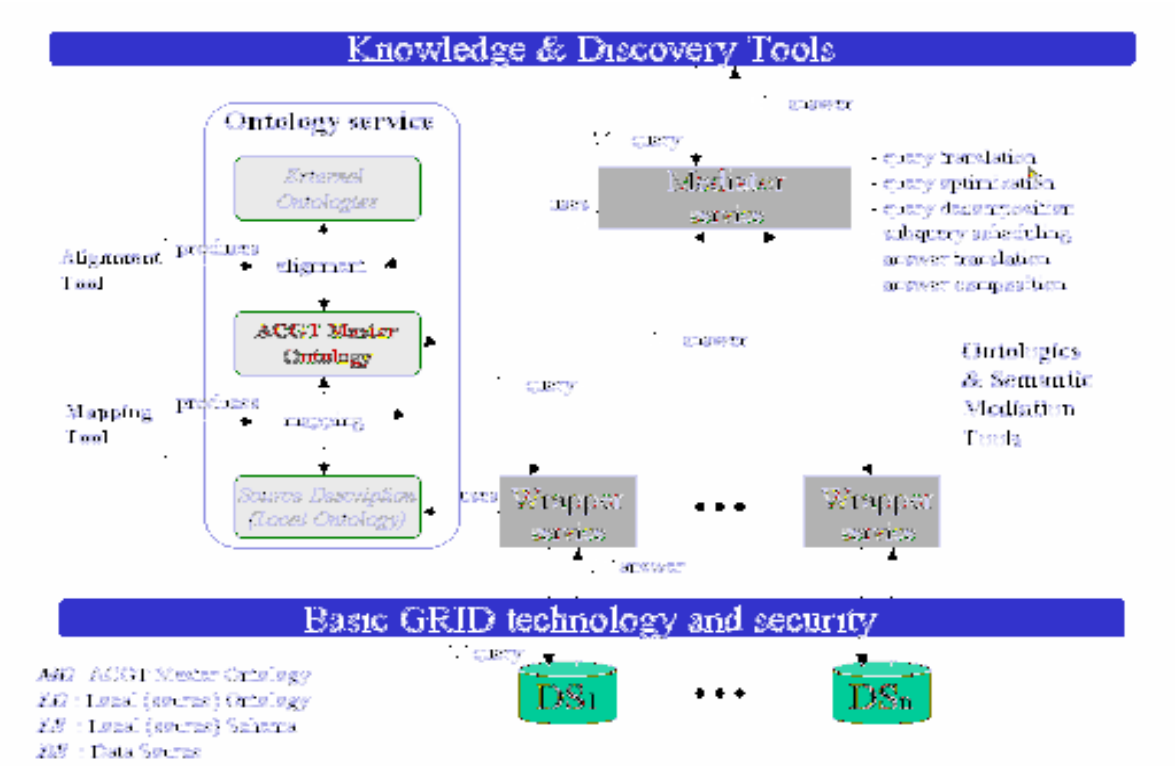


Figure 1 An outline of (part of) the ACGT platform as presented in the Description of Work

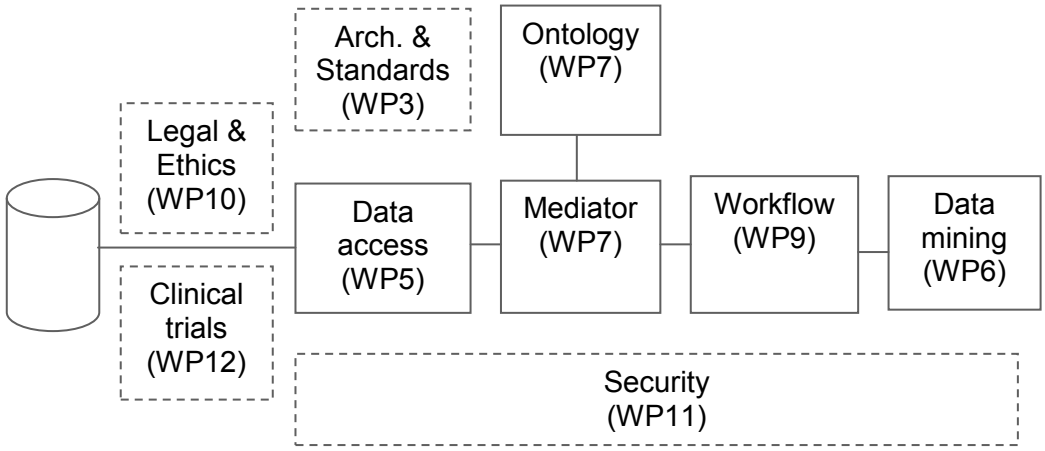


Figure 2 A simplified view of the main work package relations, from the point of view of WP5.

Figure 2 shows the main relationships between the work packages, from the point of view of WP5. WP5 depends on the clinical trials (WP12) to generate the trial-specific data and to make it available in some way. In accessing this data, the legal and ethical constraints need to be obeyed, and these are to be provided by WP10. In order to meet these constraints, the security and privacy services supplied by WP11 will be needed. The grid architecture and standards underlying the ACGT framework, the focus of WP3, are also of importance to WP5. WP5 will use part of the basic grid functions to access the various data sources and deliver the results, and all standards used by WP5 obviously need to be compatible with those of WP3.

The wrapper services provided by WP5 are used by the mediator. Close co-operation is therefore needed with WP7, to ensure that a suitable interface is defined. For development of

the mediator and the ontology, it is also important to know details about databases that are to be accessed, such as structure, content, and typical queries. Therefore, when relevant information has been obtained as part of the WP5 requirements consolidation, it is documented here extensively. As new information becomes available from the clinical trials, this document will be updated accordingly.

The mediator is used by the workflow engine, to be provided by WP9, so that workflows that require repeated access to one or more databases can be automated. This workflow engine may on its turn be used to gather the data required by the data mining tools developed by WP6. These work packages are relevant from a WP5 point of view, because they interface to the actual end users, who are obviously important when making architectural decisions.

1.3 Purpose and structure of this document

1.3.1 Goals

The goals of this document are as follows:

- Elaborate the data access requirements for WP5.
- Collect and organise the information that WP5 needs to subsequently develop an appropriate solution.
- Provide the work packages that have strong relations to WP5 with the information they need to progress their work.
- Gather and state the requirements of end-users to ensure that their needs are met and accurately understood.

1.3.2 CAFCR methodology

To carry out a well-structured analysis of the problem, we apply the CAFCR approach [1]. This framework consists of five views, which together are meant to bridge the gap between the needs, objectives and wishes of the user and their realization with available technology. These five views are customer, application, functional, conceptual, and realization. The first two are exclusively focusing on the problem that is to be solved, whereas the latter two focus on the (technical) solution. The third view sits in the middle and presents more details about the problem, as well as first steps towards a solution. Applying the CAFCR views to our requirements collection process enables us to proceed in a systematic manner, and also gives more insight and a more thorough understanding of the details and of the context of the problem we are addressing.

Figure 3 shows the five views of CAFCR framework. The Customer View refers to what the customers want to achieve. It presents the context in which the system is to operate. The Application View describes how the customers intend to realize their goals using the system. Together these two views capture the needs of the customer. The Functional View describes what the product should achieve. The non-functional requirements are also identified in this view. The Conceptual View and Realization View describe how the product will be built; they focus on the technical solution to the problem.

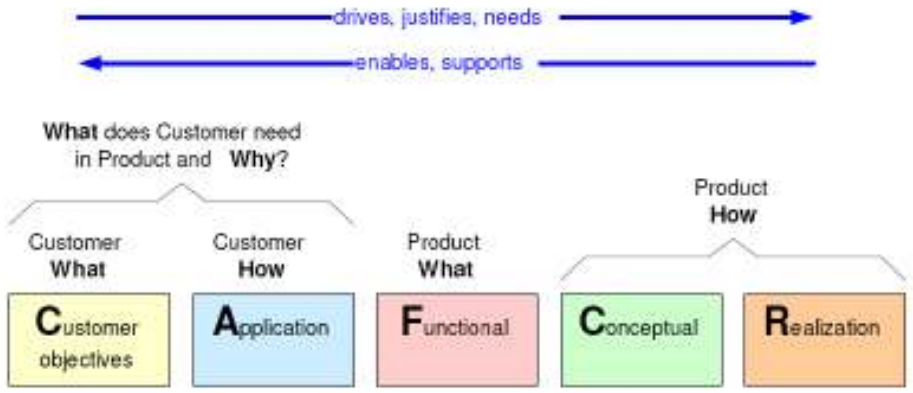


Figure 3 The five views of the CAFCR methodology

We will use the CAFCR views to structure the text. However, the use of the CAFCR framework is complicated by the fact that the document does not revolve around a single system. Multiple levels can be distinguished. Firstly, to understand the customer we need to consider the entire system used for running clinical trials and analysing the results. A part of this system will be the ACGT platform. It will hopefully address many of the customers' needs, but not all. For example, the clinical trials are already being run, and there are already databases for collecting trial-specific data. The ACGT project will not replace these. An additional complication is that the exact boundaries of the ACGT platform are not fixed yet, and may change over the duration of the project. Finally, within the ACGT platform there are various work packages. The focus of this document is WP5, the main aim of which is providing distributed access to data. Although the main goal of this document is outlining the requirements for data access, this cannot be done without considering the bigger picture, i.e. the system and users at the higher levels. This however, makes it harder to neatly organise the information across the various views. For example, in the context of the ACGT platform, the mediator is a functional component that is part of the solution, and belongs in the Conceptual view. However, from the point of view of WP5 the mediator would be a customer. In this document, we will mainly use the CAFCR views from the point of view of the overall system, while focussing on access to the data throughout.

1.3.3 Document structure

The rest of this document is structured as follows:

- Chapter 2: Customer View ("Customer What")
- Chapter 3: Application View ("Customer How")
- Chapter 4: Functional View ("Product What")
- Chapter 5: Conceptual View ("Product How – Abstract")
- Chapter 6: Realization View ("Product How – Implementation")
- Chapter 7: Conclusion

This document also contains several appendices. The first one, Appendix 1, contains short descriptions for each of the acronyms and abbreviations that are used throughout the document. It may therefore be useful to occasionally refer to it when reading this document. The other appendices will be introduced in the text at the appropriate places.

2 Customer View

This chapter gives the Customer View of the system. It starts by describing each of the clinical trials that are part of ACGT. This is done in Section 2.1. Subsequently, Section 2.2 outlines the main customer drivers. It identifies different stakeholder, and for each lists their drivers. From these, several key drivers are derived.

2.1 Clinical trials

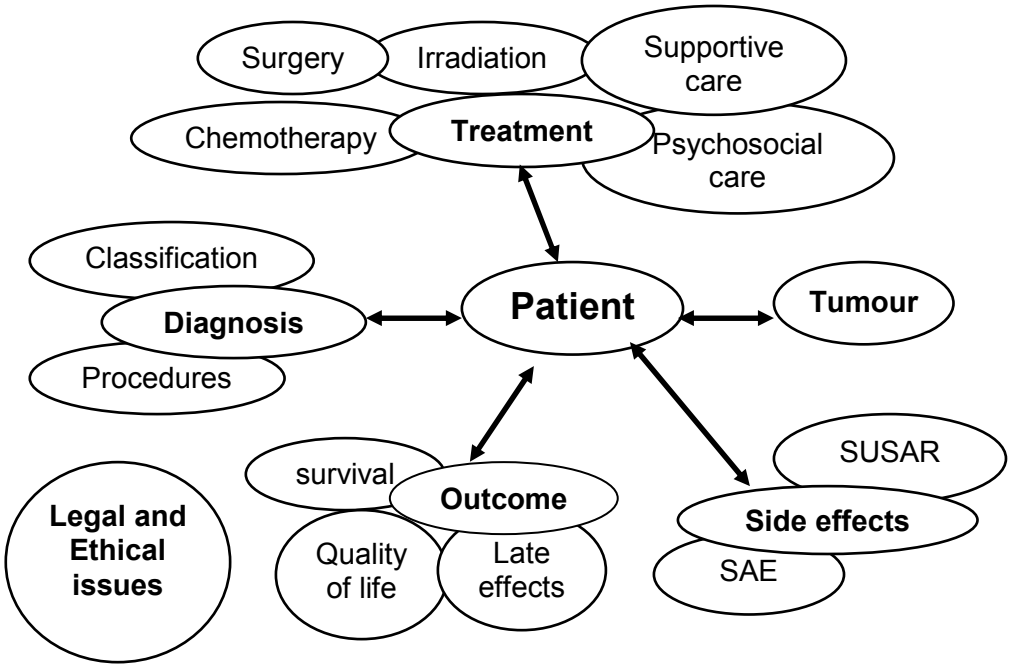


Figure 4 A patient-centric view of cancer treatment

Figure 4 shows a patient centric view of a patient with cancer who is being treated, possibly as part of a clinical trial. Firstly, there is the tumour, which is the reason that the patient ends up participating in a trial. However, from a patient's point of view, diagnosis occurs first. Only after diagnosis it is clear that there is a tumour. Subsequently treatment starts, which can have many facets, including chemotherapy, surgery, and irradiation therapy. During treatment there may be side effects. These can either be Severe Adverse Events (SAEs), or Suspected Unexpected Severe Adverse Reactions (SUSARs). The main distinction is that the latter had not been anticipated as possible reactions at the start of the trial. Finally, there is the outcome of the treatment.

Another way to look at clinical trials for cancer patients is to consider the data flows between the many entities that are involved in a trial. Figure 5 shows the relations that typically exist. A patient registers with a local hospital. After diagnosis, if the patient is eligible to take part in a trial and gives his consent, he is registered with the Trial Center specific to the trial. Next to that, the patient is also registered with a central cancer trial registry, whose main responsibility is to take care of the long-term follow up of the patient, which can be after five or ten years. This involves tracking down a patient's address if it turns out that the patient has moved. The trial center requests clinical data relevant to the trial from the local hospital. This is typically done by way of CRFs, which are sent to the Trial Center. At the moment, this rarely happens electronically; mostly the CRFs are faxed or sent by post. Image data and

tumour data that is relevant to the trial is sent to Reference Centers and/or Research Centers. Reference Centers perform second opinions on the data, and feed their results back to the Trial Center. Their function is especially important when classifying cases that occur infrequently. These are more likely to be misclassified by specialists in local hospitals, because of their rarity. Research Centers also analyse the data they receive, but their results are not fed back to the trial. When SAEs or SUSARs occur these are reported to the legal authorities as required, and also distributed to the other centers participating in the trial. This is for instance important to more quickly detect and react to SUSARs throughout the entire trial. A clinical trial that is being run nationally may also be part of a larger international trial. In this case, the trial center will forward the data that they collect onward to the center that is responsible for running the trial internationally.

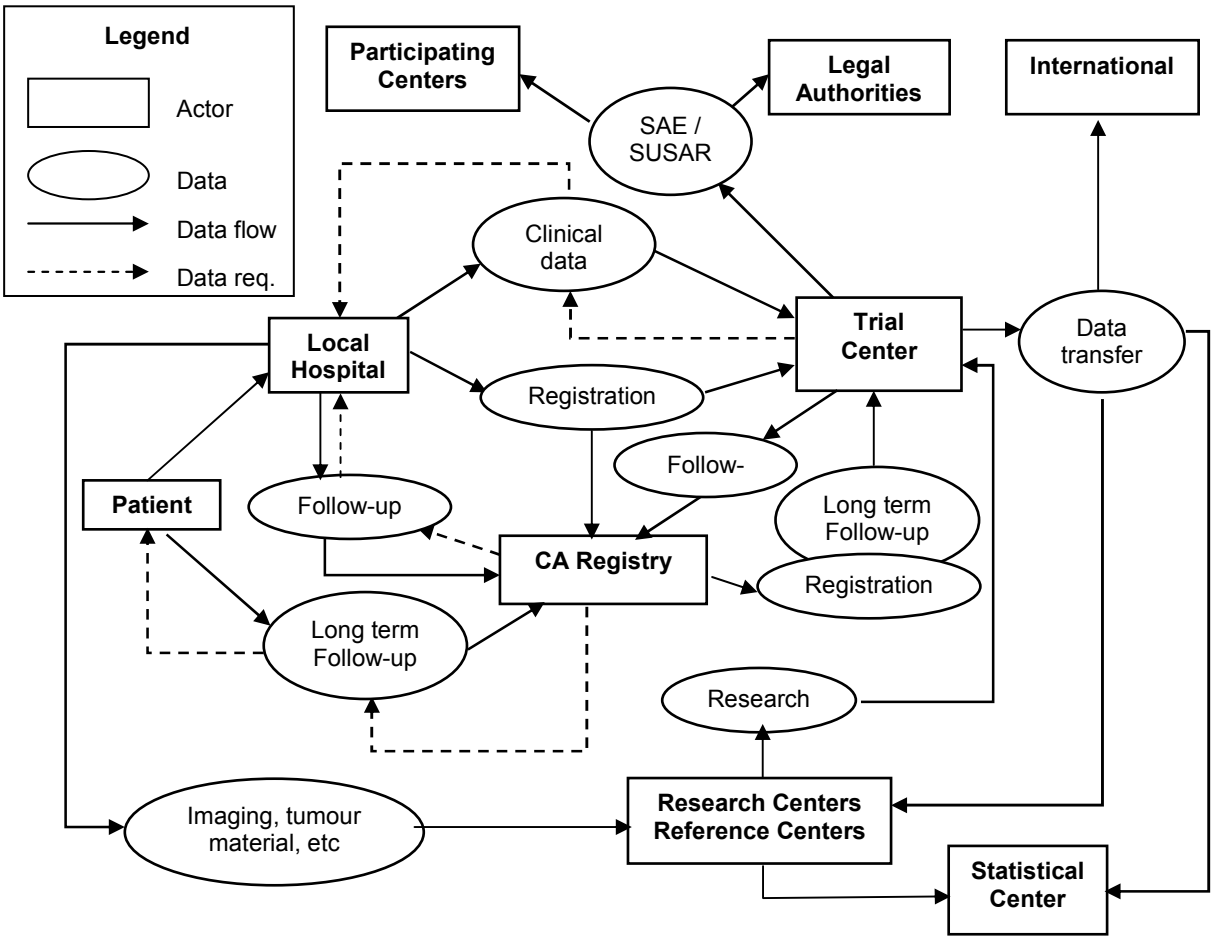


Figure 5 Data flow in a typical clinical trial against cancer.

The following sections describe the ACGT trials in more detail. Section 2.1.1 describes the TOP Trial, which addresses breast cancer. The SIOP Trial, which addresses paediatric kidney cancer, is described in Section 2.1.2. Finally, Section 2.1.3 describes the In Silico trial which uses data generated by both the other trials. Each of the trials is described in sufficient detail for the purposes of further working out the data access requirements. However, for a more detailed description of each trial, including clinical justifications for the particular experimental questions that each addresses, see D2.1 [22].

2.1.1 TOP Trial

The ACGT-TOP Trial focuses on a particular molecular subgroup of breast cancer patients and aims to determine the predictive factors of response to a single class of chemotherapeutic agents. Its sample size is based on statistical assumptions and it will include around 400 patients. Until now, approximately a hundred patients have been registered.

Breast cancer is the most common cancer in women in the world. Although mortality has declined in the last two decades, breast cancer continues to represent a major threat to the lives and productivity of women. The number of effective treatments for breast cancer rises; however, the benefit from specific treatments to individual patients and the adverse events they experience vary considerably. Efficacy and safety of anticancer therapies may depend on tumour, treatment, and host characteristics.

Conventional prognostic factors provide insufficient information to evaluate the heterogeneity of this disease and to make treatment more effective for individual patients. One problem faced by present cancer therapy is the over-treatment of patients with chemotherapy, which is associated with severe toxicity and increasing healthcare spending, without clear survival benefit over untreated controls. Because of the lack of adequate predictive markers, nearly all patients receive routinely standard treatment in spite of grim changes of deriving any benefit. Therefore, the identification of molecular markers predictive of patients' responsiveness to treatment is becoming a central focus of translational research.

The ACGT TOP study aims to identify biological markers associated with pathological complete response to anthracycline therapy (epirubicin), one of the most active drugs used in breast cancer treatment. Tumour samples drawn at the time of pre-treatment biopsy will be frozen and used to perform oligonucleotide based microarrays (Affymetrix). This technique allows the evaluation of thousands of genes and ultimately provides us with the tumour genetic profile. Homogeneous genetic profiles (genetic clusters) that might be identified, will be correlated with the efficacy of single-agent epirubicin. This correlation will be used to address the secondary end-point of this study, which is the identification of other genes or eventually a genetic profile playing a role in the determination of sensitivity to anthracyclines.

Several clinical trial centers are participating to the TOP Trial, but only two main centers will be considered in the ACGT context, Jules Bordet Institute and the University Hospital of Crete. Each of the two centers enrol patients from several other hospitals.

During this study multi-level data will be generated, managed and used: clinical data, genomic data, imaging data of several types (US, mammography, PET/CT, MRI), proteomic data, etc. Data from public biological databases may be used as well, and the volume and types of data and the number of data sources can change as the protocol may be amended during the trial. These heterogeneous data have to be integrated and shared within the clinical trial, while fulfilling the privacy and security demands.

The TOP study targets two patient groups: early breast cancer, and inflammatory and locally advanced breast cancer. These are discussed separately in the two subsections that follow.

2.1.1.1 Early breast cancer

The concept of delivering chemotherapy as primary treatment in early breast cancer patients is attractive because chemosensitivity of the tumour can be assessed "in vivo" allowing for a more "tailored" approach in systemic therapy.

The study is designed to test prospectively the value of topo II alpha gene amplification and protein overexpression in predicting the efficacy of anthracyclines. This study could have important practical implications in the daily clinical management of early breast cancer patients because, if the trial confirms that topo II α gene amplification and/or protein

overexpression are associated with high efficacy of anthracyclines, while topo II α normal/deleted gene and low protein content are associated with modest efficacy, an important step forward in the direction of anthracycline "tailoring" would be accomplished.

The practical advantage of this approach would be to use anthracyclines primarily in patients who are supposed to derive the largest benefit, thus avoiding the long-term anthracycline-related toxicity for those patients for whom no significant gain in anti-tumour activity is anticipated.

2.1.1.2 Inflammatory and locally advanced breast cancer

Inflammatory breast cancer, perhaps the most aggressive form of breast neoplasia, represents 1 to 3% of newly diagnosed breast malignancies. Often there is a diffuse infiltration of the breast without a well-defined tumour. Locally advanced breast cancer (LABC) encompasses a heterogeneous group of patients including those with neglected slow growing tumours as well as those with biologically aggressive disease.

Inflammatory and locally advanced breast carcinomas (carcinoma is a specifc type of malignant tumour) are both associated with poor prognosis. With surgery and/or radiotherapy alone, the prognosis in LABC is very poor.

In the present study, the use of a dose-dense administration of epirubicin is planned. The same drug as for early breast cancer is administered, but a slightly more aggressive regimen with a higher dose-density is used.

2.1.2 SIOP Trial

The SIOP Trial revolves around patients with Wilms' tumour. Section 2.1.2.1 provides a short background about this type of cancer. Subsequently, Section 2.1.2.2 outlines the goals and structure of the SIOP Trial.

2.1.2.1 Wilms' tumour

Although rare, Wilms' tumour is the most common kidney disease in children. Appropriate laboratory, radiologic and pathologic investigations are necessary for accurate diagnosis and subsequent treatment; information which is essential to generate a multidisciplinary treatment plan utilizing surgery, chemotherapy, and radiotherapy.

Outcomes of patients treated with either up-front surgical removal of a kidney or preoperative chemotherapy have been excellent. Multimodality treatment has resulted in a significant improvement in outcome from approximately 30% in the 1930s to more than 85% in the modern era. The results that have been achieved in children with Wilms' tumours support the strong value of the multidisciplinary team approach to cancer.

The goal of current clinical trials is to reduce therapy for children with low-risk tumours, thereby avoiding acute and long-term toxicities. Challenges remain in identifying novel molecular, histological and clinical risk factors in order to more effectively determine the proper treatment for each patient. This could allow a safe reduction in therapy for patients known to have an excellent chance of cure with the current therapy, while identifying, at diagnosis, the minority of children at risk of relapse, who will necessitate more aggressive treatments.

2.1.2.2 Goals and structure of the trial

The ACGT Nephroblastoma clinical study is based on SIOP 2001 study, a continuation of the philosophy of the former SIOP studies. The basic idea has always been: Collect a lot of reliable data by working together on an international base and answer questions which can

be of direct importance for the outcome of the patients. SIOP 2001 is based on the results of the previous SIOP trials and studies as well as on the results of the NWTS protocols.

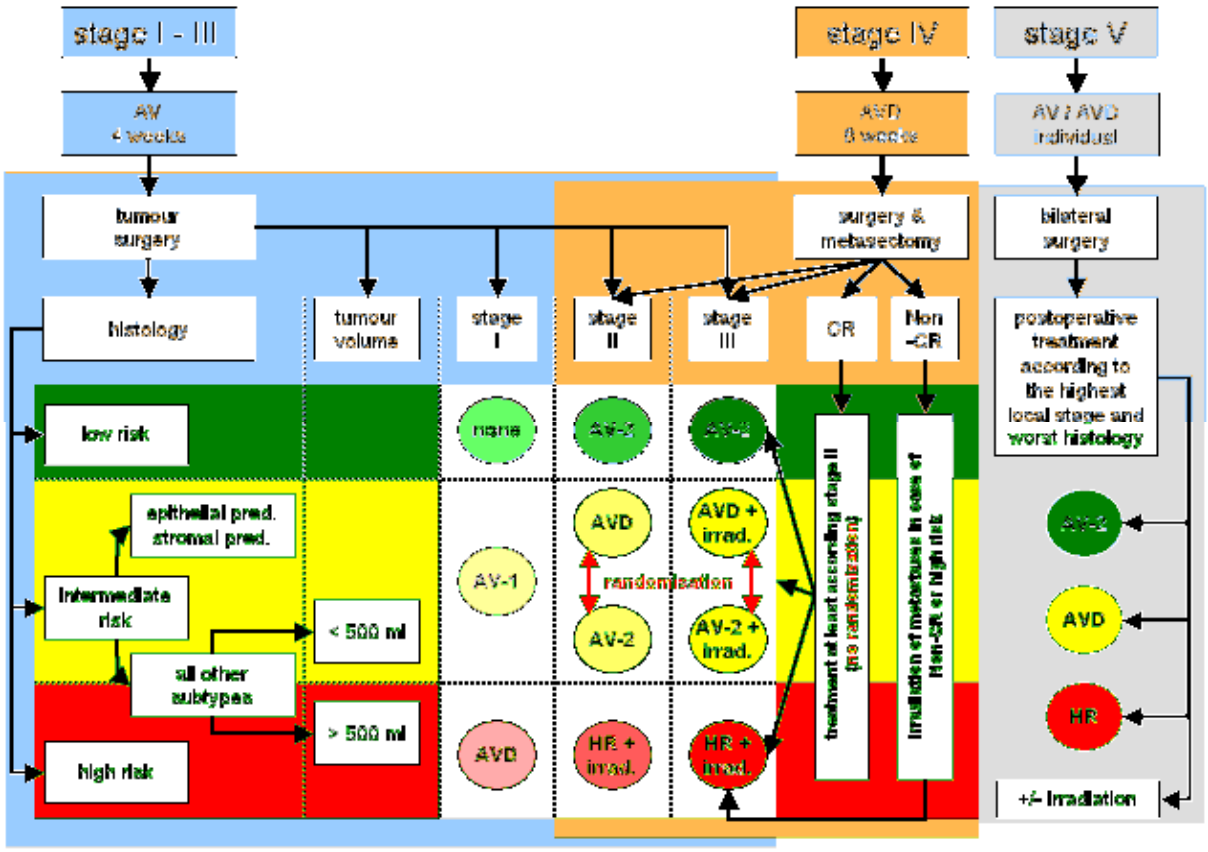


Figure 6 Outline of SIOP 2001/GPOH trial. It shows how treatment differs based on the tumour staging and risk derived from histology.

The structure of the trial is summarised in Figure 6. Treatment depends on the classification (staging) of the tumour. The initial diagnosis is by way of imaging; the precise modality depends on the hospital where the patient is enrolled. Stage I-III means that the tumour is localised. Here the patient receives four weeks of pre-operative chemotherapy. The tumour is classified as Stage IV when it is metastatic, i.e. it has spread to other places in the body. In this case, the neo-adjuvant chemotherapy lasts six weeks. Stage V is when the cancer is bilateral, i.e. there is a tumour in both kidneys. In this case the patient will receive individualised treatment, in order to save as much of the kidney function as possible. After operation, for Stage I-III, the tumour is classified further depending on the outcome of the operation. With Stage I, the tumour is taken out completely, entirely in capsule. At Stage II, the tumour is also taken out entirely, but is partly outside the capsule. With Stage III, the tumour has ruptured, or part of the tumour is remaining in the body. The tumour is also analysed by means of histology, and classified as low, intermediate, or high risk. In the case of intermediate risk, the type of the tumour (which cell type is pre-dominant) is also taken into account, and potentially its size as well. For instance, a tumour that is classified as medium risk based on the histology data can be classified as high risk if it is large. The type of adjuvant therapy that the patient subsequently receives is based on the tumour's stage, as well as its risk. In this trial, the randomisation involves patients with Stage II or III tumour of intermediate risk. Treatment for Stage IV patients depends on whether or not there is complete remission (no tumour left in the body). In case of complete remission, the patient is treated as Stage II or III always with the addition of anthracylines. Otherwise subsequent treatment always follows that of tumours classified as high risk. Treatment of Stage V

patients depends on highest stage and the worst histology data considering both tumours (of either kidney).

The distribution of patients into stages is approximately as follows:

- Stage I-III: 85%
- Stage IV: 10%
- Stage V: 5%

This distribution is affected by the quality of imaging studies. More accurate imaging can increase the number of patients in the metastatic stage (stage IV).

There are approximately a hundred hospitals involved in the trial. Of these, ten hospitals contribute half of the patients enrolled in the trial. Some smaller hospitals may contribute to the trial only one patient every five years.

The specific objectives of the SIOP Trial are:

- To adapt therapy to the known individual risk of the patient and increase survival for blastemal predominant tumours (*blastema is a specific type of tissue*) after preoperative chemotherapy by intensifying therapy and minimise acute and late toxicity without jeopardising event free survival and survival by reducing treatment for patients with focal anaplasia (*a certain histological characteristic of the tumour*), for stage I patients with intermediate risk tumours, and for stage II and III patients with intermediate risk tumours by randomising doxorubicin.
- To test the treatment hypothesis that doxorubicin is not necessary in patients with intermediate risk tumours and local Stage II or III by a multicentric prospective randomised trial.
- To prospectively analyse different histological components of Nephroblastoma with a special emphasis on a percentage of blastemal component which might be of prognostic significance.
- To reduce the number of drug administrations, hospital visits and thereby costs in the pre- and postoperative phase.
- To collect material for performing biological studies with specific aims and clinical research scenarios.

The specific research question that the randomisation of the SIOP 2001 trial is designed to address is the following: Is for medium risk patients a two drug post-operational treatment sufficient instead of the normal three drug treatment? The randomised question applies to approximately 15% of patients enrolled in the study.

Besides the excellent prognosis of children with Wilms tumour there is a well known risk of unnecessarily administered chemotherapy by treating children preoperatively without histologically proven diagnosis. This risk could be abolished by finding a specific marker for Wilms tumour in serum, which is lacking today. Immunogenic tumour-associated antigens have been reported for a variety of malignant tumours including brain tumours, prostate, lung and colon cancer.

The purpose of the ACGT Nephroblastoma study is to find such a marker by searching for a pathognomonic (*characteristic or diagnostic for a particular disease*) antigen (*a substance that stimulates an immune response*) pattern in patients with Wilms tumour. Serum from a specific patient will be tested against newly identified Wilms tumour antigens. As a result, for each patient there will be a specific pattern of antigens found. This pattern will be correlated to the histological subtype of the tumour, the gene expression profiling of the tumour, the

response to chemotherapy and the outcome of the patient. The study is described in further detail as Scenario S2 in D2.1 (Section 5.3).

2.1.3 In Silico Modelling of Tumour Growth

The aim of this third ACGT study is to provide clinicians with a decision support tool able to simulate within defined reliability limits the response of a solid tumour to therapeutic interventions based on the individual patient's data. This tool would enable clinicians to investigate the impact of specific treatment-induced perturbations over several orders of magnitude, which is currently impossible with conventional imaging methods alone. An intermediate goal of the study would be to provide researchers with a versatile platform for integrating experimental and clinical knowledge and performing exploratory experiments in silico.

The In Silico Oncology clinical research will be based on the two other clinical trials incorporated in ACGT (the expansions of the Nephroblastoma SIOP 2001/GPOH and breast cancer TOP Trial), and would aim at developing, optimising, validating and clinically adapting a computational system, denoted by the specially coined term "Oncosimulator" that would serve as simulation model of tumour response to chemotherapy. The most critical biological phenomena (e.g. metabolism, cell cycling, geometrical growth or shrinkage of the tumour, cell survival following irradiation or chemotherapeutic treatment, necrosis, apoptosis etc.) will be thus spatiotemporally simulated using a variety of clinical, radiobiological, pharmacodynamic, molecular and imaging data.

Furthermore, Virtual Environments designed to represent 3D (and to some degree also 4D) data and to provide intuitive interactive methods to explore this data will be applied for the virtual reality visualisation of both medical images and in silico oncology simulation results. The objective is to "involve" the researchers more, and bring them closer to their data in an effort to detect patterns and structures using the researchers' experience, expertise and cognitive abilities.

All types of data generated in the clinical trials are relevant for this study. The data provided to the Oncosimulator should be pseudonymized to allow the validation of the model and of the simulation results against the ground truth from the clinical trials.

2.2 Drivers

In this section customer drivers are identified. Section 2.2.1 starts of by listing various explicit data access wishes that have been expressed by users. Next, Section 2.2.2 takes a more generic and abstract approach by identifying all stakeholders of the system, and listing their drivers. From this, key drivers are extracted, which are presented in Section 2.2.3.

2.2.1 High level data access wishes expressed by users

In consolidating the requirements for data access, we visited the clinical partners at the University of Saarland and at the Jules Bordet Institute. During these meetings, they explicitly expressed various wishes with respect to data access, including data entry. Even though, strictly speaking, some may fall outside the scope of the ACGT project as defined in the description of work, they are obviously important and need to be taken into account. Furthermore, they also help to understand the customer better, and therefore it is worth listing them in the Customer View.

During the requirements gathering meetings for data access, the clinical users expressed the following wishes:

- Distributed entry of CRF data: Ability to enter CRF data at each clinical site, yet allow central validation of data. Currently the CRF data is typically sent by all the participating

hospitals to the clinical trial center in paper form, where it is manually entered in the system. It would be much more convenient if each participating hospital can locally enter their data. These data should then become available at the clinical trial centers where it can be further validated. The remote entry and/or the transfer of data should happen seamlessly for all the users involved.

- Distributed entry of bio-medical analysis results. The molecular biologists and clinical researchers involved in the trial would also benefit from collaborative approaches for molecular data analysis. Instead of sending and storing the biological samples and carrying out the analysis at the trial centers, the laboratory experiments on biological data should be performed locally, while the results should be available globally to all authorized users.
- Support for reporting SAEs and SUSARs.
- Enable safe cross-organisational data sharing.
- Homogenisation of procedures for future clinico-genomic trials, from CRF creation to analysis of molecular data. A standard set of procedures to carry out clinico-genomic trials, from the creation of the CRFs to the analysis of the molecular data, would be of great benefit.

2.2.2 Stakeholders and drivers

In this section, we present all stakeholders of the system, and their drivers. In doing so, we differentiate between the clinical trial centers (Jules Bordet Institute, University Hospital of Crete, University of Saarland), which are the sites carrying out the research in the trial and are responsible for both the process and the results, and the participating (treating) hospitals, which enrol patients in the clinical trials. Each clinical trial center enrols patients and integrates patient data from several participating hospitals. For example, the structure of the TOP Trial, as it will be integrated in the ACGT environment, is presented in Figure 7.

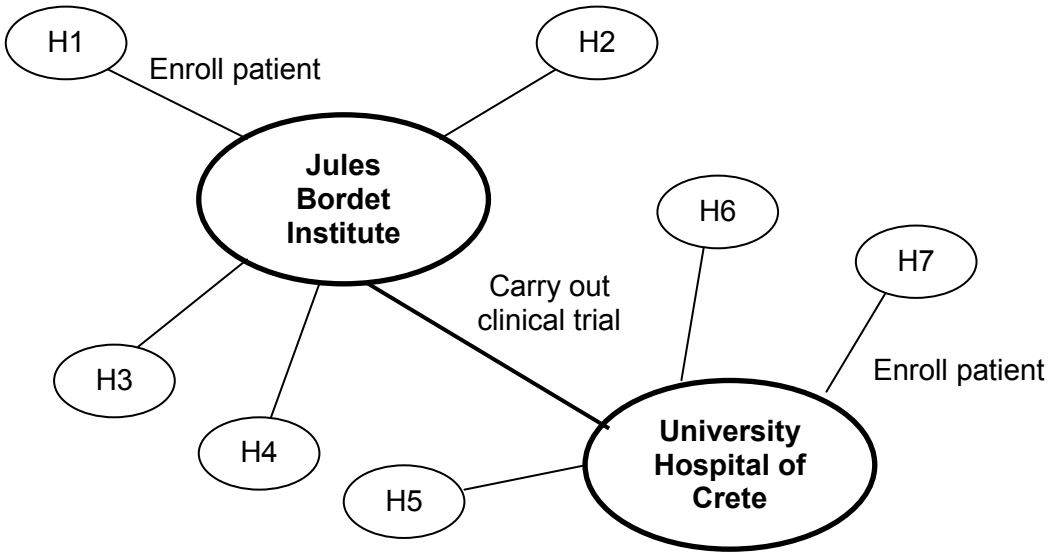


Figure 7 The structure of the TOP Trial.

The following stakeholders can be identified. For each, their drivers are also listed:

- **Principal investigator at clinical trial center**
  The principal investigator is the main person responsible for the clinical trial. The drivers

relevant for this stakeholder are access to distributed data, role-based access, compliance to privacy and security policies, compliance to legal regulations, correctness and completeness of the retrieved data.

- **Medical researcher at clinical trial center**
  The role of the medical researchers at the clinical trial site is to carry out the research in the clinical trial with data from all the participating sites and to treat the local patients. The relevant drivers are distributed access to data, role-based access, ease of use, reliability, performance, privacy, correctness and completeness of data retrieved, easy access to all relevant information, and expressiveness and flexibility of supported queries.
- **Molecular biologist at clinical trial center**
  The molecular biologist at the clinical trial site carries out research in the clinical trial to validate the hypothesis of the trial, with data from all participating hospitals. The drivers for this stakeholder are distributed access to data, role-based access, ease of use, reliability, performance, correctness and completeness of data retrieved, easy access to all relevant information, and expressiveness and flexibility of supported queries.
- **Pathologist at the clinical trial center**
  The pathologist at the clinical trial center needs access to the biological data of the patients enrolled in the trial and to the data input into the system at the participating sites. They update the system with the results of the lab tests on the biological samples. The drivers for this stakeholder are distributed access to data, role-based access, ease of use, reliability, performance, correctness and completeness of data retrieved, and easy access to all relevant information.
- **Data manager at the clinical trial center**
  This stakeholder is responsible for entering the data into the system, at the clinical trial center side. The drivers for this stakeholder are role-based access, ease of use, reliability.
- **Clinical IT staff at clinical trial center**
  They are responsible for all the IT-related aspects of the clinical trial. The main drivers for this stakeholder are maintainability, ease of use, reliability, flexibility, security, performance, compliance to standards, interoperability, scalability, and correctness and completeness of data retrieved.
- **Investigator at participating hospital**
  The investigator is responsible for the clinical trial. The drivers relevant for this stakeholder are role-based access, compliance to privacy and security policies, compliance to legal regulations, correctness and completeness of the retrieved data.
- **Medical researcher at participating hospitals**
  The role of the medical researcher is to treat the local patients and to provide the necessary research data for the clinical trial. The main drivers are remote access and data entry, role-based access, ease of use, reliability, privacy, correctness and completeness of data retrieved, and easy access to all relevant information.
- **Molecular biologist at participating hospital**
  The molecular biologist at the participating hospital analyses patient data at molecular level and provides data to the clinical trial center. The drivers for this stakeholder are remote access and data entry, role-based access, ease of use, reliability, performance, correctness and completeness of data retrieved, and easy access to all relevant information.
- **Pathologist at participating hospital**
  The pathologist collects and analyses the biological data of the local patients. They update the local system with the results of the lab tests on the biological samples. The

drivers for this stakeholder are remote access and data entry, role-based access, ease of use, reliability, performance, and easy access to all relevant information.

- **Data manager at participating hospital**
  This stakeholder is responsible for entering the data into the system, at the hospitals taking part in the clinical trial. The drivers for this stakeholder are remote access, role-based access, ease of use, reliability.
- **Clinical IT staff at participating hospital**
  The IT staff are responsible for all the IT-related aspects of the clinical trial. The main drivers for this stakeholder are maintainability, ease of use, reliability, flexibility, security, performance, compliance to standards, interoperability, and scalability.
- **Patient enrolled in a clinical trial**
  The main drivers relevant for a patient enrolled in a clinical trial are compliance to privacy and security policies, compliance to legal regulations, and correctness and completeness of the retrieved data.
- **Researcher involved in the development of the Oncosimulator**
  The main drivers for this stakeholder are ease of use, reliability, flexibility, performance, compliance to standards, access to distributed data, correctness and completeness of data retrieved, expressiveness of queries, easy access to all relevant information, and remote access.
- **Researcher carrying out data mining experiments**
  The main drivers for this stakeholder are ease of use, reliability, flexibility, performance, access to distributed data, compliance to standards, correctness and completeness of data retrieved, expressiveness of queries, easy access to all relevant information, and remote access.
- **Researcher involved in the development of data integration tools**
  The main drivers for this stakeholder are compliance to the ACGT interfaces, ease of use, reliability, flexibility, expressiveness of queries, security, performance, compliance to standards, interoperability, scalability, access to distributed data, correctness and completeness of data retrieved, and remote access.
- **Researcher responsible for ensuring the security of the ACGT platform**
  The main drivers relevant for this stakeholder are compliance to privacy and security policies, and role-based access.
- **Researcher responsible for ensuring the compliance to the legal regulations**
  The drivers relevant for these stakeholders are compliance to privacy and security policies, and compliance to law.
- **Pseudomization Trust Center**
  The drivers relevant for these stakeholders are compliance to privacy and security policies, compliance to the ACGT interfaces, compliance to standards.
- **Governmental bodies**
  The drivers relevant for these stakeholders are compliance to privacy and security policies, and compliance to law.

Summing up, the set of drivers relevant for this application are:

- Distributed access to data
- Role-based access
- Remote access and data entry

- Compliance to the ACGT interfaces
- Performance
- Reliability
- Maintainability
- Ease of use
- Flexibility
- Expressiveness of queries
- Correctness and completeness of data retrieved
- Interoperability
- Compliance with standards
- Privacy
- Security
- Scalability
- Compliance with laws

2.2.3 Key drivers for data access

With respect to data access, the following drivers are considered the most important:

1. Distributed access to data: A clinical researcher involved in the clinical trial can transparently access all the data of the patients in the trial, regardless of its physical location.
2. Remote access and data entry: Clinical researcher at participating hospitals can remotely access the data of their patients enrolled in the clinical trial. A data manager at a participating hospital can enter the patient data remotely, directly in the database(s) of the clinical trial. In both cases role-based access is necessary.
3. Performance: A clinical researcher or molecular biologist involved in the trial can query all the relevant data and obtain quick response regardless of the location where the data is stored.
4. Flexibility: A clinical researcher or molecular biologist involved in the trial should be able to receive answers to all relevant queries. The IT staff should be able to define any new (combinations of) queries against the data in the clinical databases when necessary. Researchers working on data mining or on the oncosimulator should be able to query any of the data wrapped by the data access services.
5. Interoperability: Researchers carrying out data mining experiments should be able to compose the data access services according to the needs of their experiments, and easily exchange information between the data access services and their own tools. The mediator should be able to retrieve data from the databases through the data access services.

Note, it is assumed here that ensuring that the privacy of patients is guaranteed is not the responsibility of the data access services, i.e. the data in the databases that are accessed by the wrapper services have already been anonymised. However, as the privacy requirements have not yet been finalised, the approach and precise mechanisms that ensure that these requirements are met have not been finalised either. The approach that is chosen here will influence our solution to data access.

3 Application View

In this chapter we address the Application View of the CAFCR framework by considering in more detail how the users want to achieve their objectives, with a focus on data access. Section 3.1 describes for each of the clinical trials what data is collected, and when it is collected. Subsequently, Section 3.2 examines the data access aspects of various scenarios.

3.1 Data collected

Section 3.1.1 describes the data that is being collected during the TOP Trial. Similarly, Section 3.1.2 does so for the SIOP Trial.

3.1.1 TOP Trial

Section 3.1.1.1 describes what data is collected and when it is collected, from a patient-centric view point. Next, Section 3.1.1.2 describes the Clinical Report Forms that are used to collect data during various stages of the trial. Finally, Section 3.1.1.3 describes the biological analyses that are being performed.

3.1.1.1 Timeline

The aim of the TOP Trial is to determine the predictive factors of response to a single class of therapeutic agents, the anthracyclines (Epirubicin), for a particular molecular subgroup of breast cancer patients with ER-negative tumours. To this end, at different steps during treatment different sets of data are being collected.

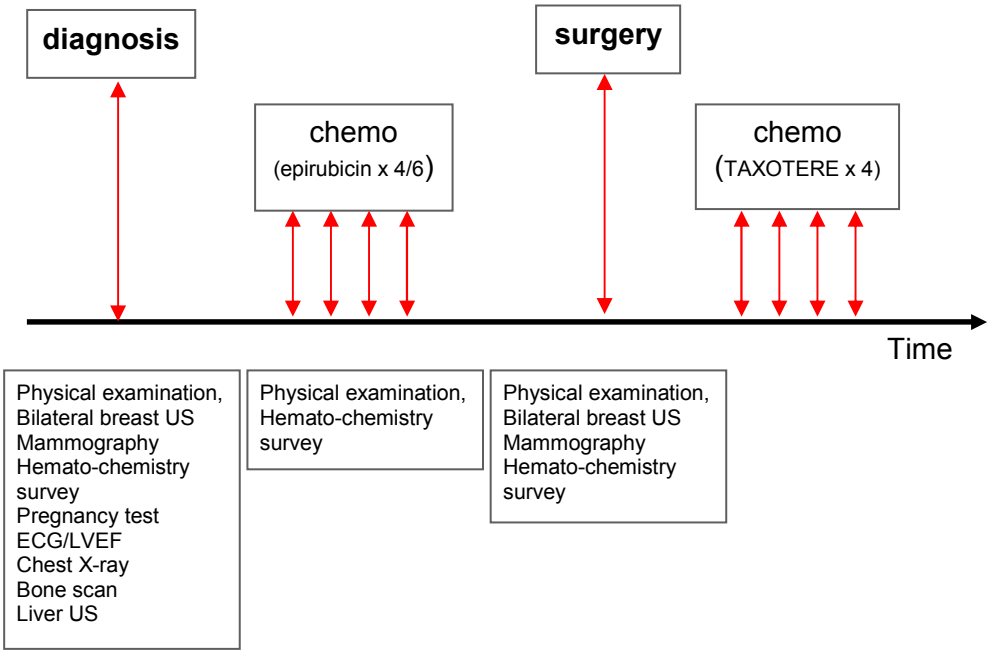


Figure 8 Timeline of patient treatment during TOP Trial

In the time-line of a patient encounter, four different phases can be distinguished: diagnosis, neo-adjuvant therapy (chemotherapy), surgery and adjuvant therapy. For the TOP Trial, the last phase is out of the scope of ACGT. Therefore, in this document only the data acquired during the steps between diagnosis and surgery are being considered. Figure 8 summarises the four phases of the TOP Trial, and the data that is collected in the first three phases.

The eligible patients are split into two groups, each group being treated with the same neo-adjuvant therapeutic agent and for the same period of 12 weeks. While the early breast cancer group will receive chemotherapy every three weeks (4 cycles) , the locally advanced and invasive breast cancer group will receive chemotherapy every two weeks (6 cycles).

3.1.1.2 Clinical report forms

For each relevant step, the necessary clinical data will be collected in paper-based CRFs by the treating hospitals participating in the clinical trial, according to the protocol. The CRFs are then centralized at the clinical trial center. A data manager has to collect all the required data, and to enter it into databases in the right format. Then the data needs to be verified and validated. The validation checks are specified based on the protocol requirements. At the end of the validation a discrepancy report is generated.

Table 1 provides an overview of the data included in each CRF.

ID	Title	Description
1	Patient registration form	Details about the investigator, information regarding the patient and the type of treatment planned, and the checklist of eligibility criteria.
2	Patient's characteristics	Height and weight of the patient, menopausal status, significant medical history, date and identification number of biopsy.
3	Pre-treatment tumour characteristics	Histology information before treatment: date of histological diagnosis, estrogen and progesterone receptor status, and histopathologic type and grade.
4	Tumour assessment baseline evaluation	Assessment of primary tumours and of lymph nodes involvement. Clinical examination and US.
5	Administration of Epirubicin	Details regarding the Epirubicin administration for each chemotherapy cycle.
6	Toxicity	Reports on the adverse events during the chemotherapy cycles.
7	Primary treatment completion	Specifies the number of chemotherapy cycles, the dose density and the cumulative dose, and specifies the reason for ending the treatment.
8	Tumour assessment: Post-Epirubicin Evaluation	Assessment of the primary tumours and of the lymph nodes involvement at the end of the chemotherapy cycles. Clinical examination and ultrasound.
9	Breast cancer surgery	Specifies the date and the type of surgery.
10	Residual tumour characteristics	Describes the type, grade and size of the residual tumour.
11	Regional lymph nodes	Assessment of the lymph nodes

12	Post-surgery treatment	Details of treatment after surgery: radiotherapy, hormonal treatment, adjuvant chemotherapy, etc.

Table 1 The clinical report forms used in the TOP Trial.

According to the CRFs the following data is being collected at each stage.

- **Baseline (at diagnosis and before starting the treatment with Epirubicin):**
  - Medical history
  - Patient characteristics
  - Physical examination + clinical tumour assessment
  - Breast biopsy + measurement of hormone receptors
  - Serum sample (at diagnosis)
  - Whole blood sample (before starting treatment)
  - Haematology and biochemistry
  - ECG
  - Chest X-Ray
  - Bone scan
  - Liver ultrasound
  - Bilateral Mammography
  - Breast ultrasound
  - Informed consent
- **Epirubicin treatment period (up to 6 cycles):**
  - Physical examination + clinical tumour assessment
  - Serum sample (after the 1st chemotherapy cycle and at surgery time)
  - Haematology and biochemistry (before each cycle of chemotherapy)
- **Post-Epirubicin treatment:**
  - Physical examination + clinical tumour assessment
  - Serum sample
  - Haematology and biochemistry (post-surgery?)
  - Bilateral mammography
  - Breast ultrasound

3.1.1.3 Biological analyses

The biological specimens stored and analyzed are described in both the protocol and the patient consent form. They are collected at different stages during the study. The following biological analyses are planned:

1. Immunohistochemistry:

- ER and PgR
- Topoisomerase II alpha (topoII)
- HER2
- Ki-67
- HER1/EGFR
- CK5/6
- BRCA1

These experiments are performed on both pre-treatment (core biopsies at diagnosis) and surgical samples. The results are stored in an Excel spreadsheats or Access databases and expressed as percentages of stained cells (with or without intensity information), or as a score.

2. Fluorescent In Situ Hybridisation (FISH): HER2 and topoII amplification status assessed with a triple probe by Vysis (Illinois, USA). The results are stored in Excell/Access databases as HER-2 gene copy number per 60 cells, topo II gene copy number per 60 cells, centromere 17 copy number per 60 cells and the ratios.
3. Gene expression profiling: U133 Plus 2.0 GeneChip® arrays from Affymetrix®, which contain approximately 47,000 genes, will be used. For all the samples an H&E (hematoxylin-eosin) stained section will be prepared prior to cutting slides for RNA isolation to assess tumour cell percentage; only samples with >70% tumour cells will be considered. Raw results are stored as CEL files of about 30MB, as text files.
4. Proteomics: The plasma and serum will be used for SELDI, LC-MS/MS or other proteomics studies.
5. Genotyping: DNA will be extracted from blood sample drawn at the time of treatment initiation to perform genotyping analyses. Genotyping will be realized with the 500K arrays from Affymetrix, according to protocols recommended by Affymetrix.
6. Comparative Genomic Hybridisation (CGH): Array-based CGH will be done using the Affymetrix 500K arrays, according to protocols recommended by Affymetrix.
7. microRNA's analysis: MicroRNAs are an abundant class of small non-protein-coding RNA's that function as negative gene regulators. The exact technique that will be used is still to be determined.
8. Sequencing: The intention is to sequence the genes p53 and BRCA1, whose mutations seem to occur particularly in ER-negative tumours. They may have important prognostic and predictive relevance.
9. Detection of Circulating Tumour Cells (CTC's): This has not yet been started but patients will be asked to provide blood samples to determine the number of CTC at inclusion, after the first cycle, and on the same day that the patient will have surgery. (The "CellSearch" technology from Veridex is available at J. Bordet.)

3.1.2 SIOP Trial

This section describes the data that is being collected as part of the SIOP Trial. It is structured similarly to the previous section. Section 3.1.2.1 describes what data is collected and when it is collected, from a patient-centric view point. Next, Section 3.1.2.2 describes the Clinical Report Forms that are used to collect data during various stages of the trial.

3.1.2.1 Timeline

Figure 9 shows the timeline of a patients treatment during the trial. At time of diagnosis imaging studies are obtained. Also, the patient is registered with the Children Cancer Registry. After diagnosis, the patient receives neo-adjuvant chemotherapy. Subsequently surgery takes place. Here, biological samples are obtained, and histopathological analysis of the tumour takes place. This is followed by adjuvant chemotherapy, during which the patient may also receive radiation therapy (in particular when after operation there may still be tumour cells remaining in the body). At any time during treatment SAEs or SUSARs could occur. After successful treatment there are regular follow-ups, to check that there are no relapses.

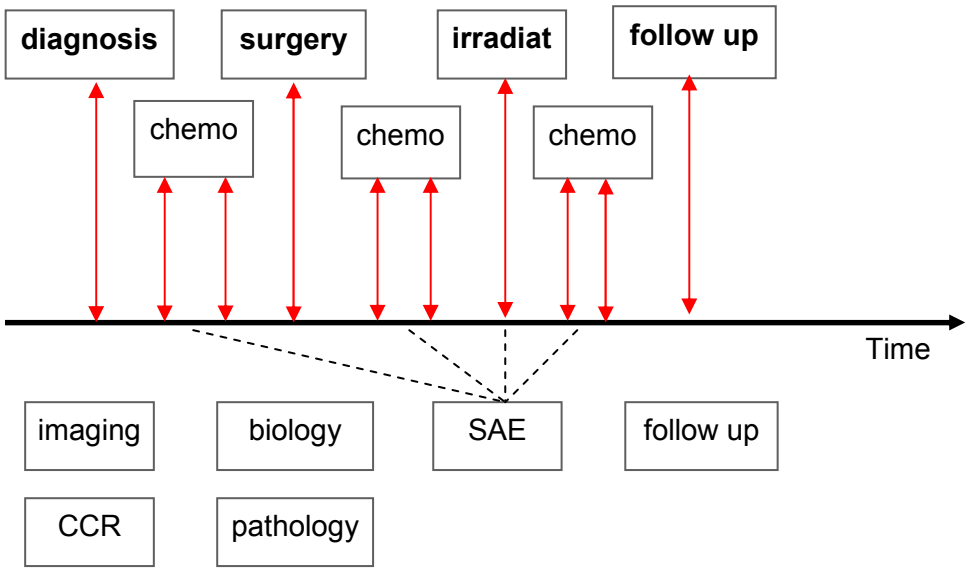


Figure 9 Timeline of patient treatment during SIOP Trial

3.1.2.2 Clinical report forms

The CRFs that are used as part of the SIOP Trial as summarised in Table 2. Some forms need to be filled in for each patient, such as F1 and F3a. For other forms, it depends on the treatment stratification for the patient whether or not it needs to be filled in. For example, F6 only needs to be completed for patients who receive radiation therapy, which is not the case for all patients (refer to Figure 6). Furthermore, there are many F7 forms, but only one needs to be filled in, based on the post-operative chemotherapy that is received. Finally, some forms, such as F3a_K and F8b, only need to be filled in if complications occur during treatment.

ID	Title	Description
F1	Registry form.	Details about any therapy so far, medical family background, size of tumour at diagnosis, details about metastases.
F2a	Preoperative chemotherapy - Stage I, II, III, V and IV-CT	Size of tumour after four weeks of chemotherapy, and details about metastases
F2b	Preoperative chemotherapy - Stage IV	Details of treatment, size of tumour after six weeks of chemotherapy, and details about metastases

F3a	Operative findings – Primary tumour	Details of operation, operative findings, complications, etc). Filled in by attending surgeon, directly after surgery.
F3a_K	Surgery form – Postoperative complications	Details about the complication.
F3b	Surgery form – Metastases	Details about the metastases. Filled in by attending surgeon, directly after surgery.
F4	Pathology form	Filled in by pathologists. Pathology findings of surgically removed tumour. Includes staging of tumour.
F5	Randomisation form	Result of randomisation.
F6	Post-operative radiotherapy	Details of radiation therapy.
F7a	Post-operative chemotherapy – AV 1	Details of treatment and result.
F7b	Post-operative chemotherapy – Randomisation	Records if randomisation actually took place.
F7c	Post-operative chemotherapy – AVD, AV-2	Details of treatment and result.
F7d	Post-operative chemotherapy – VP16/CARBO/CYCLO/DOX	Details of treatment and result.
F7c4	Post-operative chemotherapy – AVD - Stadium IV	Details of treatment and result.
F7d4	Post-operative chemotherapy – VP16/CARBO/CYCLO/DOX - Stadium IV	Details of treatment and result.
F7e	Post-operative chemotherapy – Regime 1 [VCR]	Details of treatment and result.
F7f	Post-operative chemotherapy – Regime 2 [AV]	Details of treatment and result.
F7g	Post-operative chemotherapy – Regime 3 [AVD]	Details of treatment and result.
F7h	Post-operative chemotherapy – Regime 4 [VCCD]	Details of treatment and result.
F7x	Post-operative chemotherapy – Individualised	Details of treatment, treatment result, and reason for deviating from therapy specified by protocol.
F8a	Cardiotoxicity	Details of cardiac symptoms.

F8b	Serious Adverse Events	Details of SAE.
F9	Follow-up form	Details of patient status at time of follow-up.

Table 2 The clinical report forms used in the SIOP Trial.

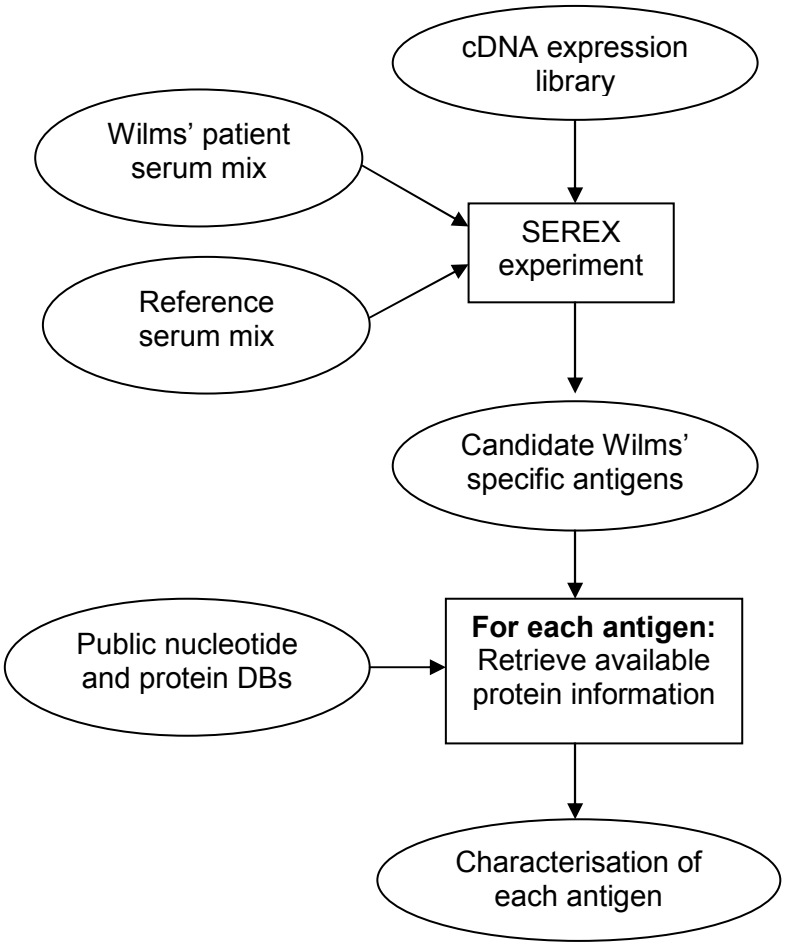


Figure 10 Overview of Step 1 of SC2

3.2 Scenarios

3.2.1 SC2

SC2 is described in detail in D2.1 in Section 5.3. It consists of two steps. The aim of the first step is to identify and characterise candidate Wilms' specific antigens. This step only has to be performed once, but must be carried out before the second step can commence. In the second step of the scenario, for each patient and at different time points during the treatment, it is checked which of the candidate antigens are relevant in the patient's tumour. This data, together with the data that is being collected by way of the CRFs, is then further analysed.

Figure 10 summarises Step 1 of the scenario. It starts with a SEREX experiment, which is carried out in a bio-molecular lab. Two batches of serum are used as input to the experiment.

The first batch is a mix of serum from patients with different types of Wilms' tumour. The second batch is a mix of serum from healthy persons as well as persons with different but related diseases. The aim of the experiment is to find a collection of antigens that are specific to patients with a Wilms' tumour. A cDNA library is used to provide a large collection of proteins that could be potential antigens. The output of the experiment is, hopefully, a small collection of antigens that triggered an anti-body response with the first batch of serum, yet not with the second. These antigens are thus potentially specific to Wilms' tumour. For each of these antigens, further information is obtained from various relevant public databases. This includes information about their structure, function, pathways they occur in, etc. In fact, the public sequence databases are also used to confirm that each of the nucleotide sequences that result from the SEREX experiment, is "in frame" with its corresponding protein sequence.

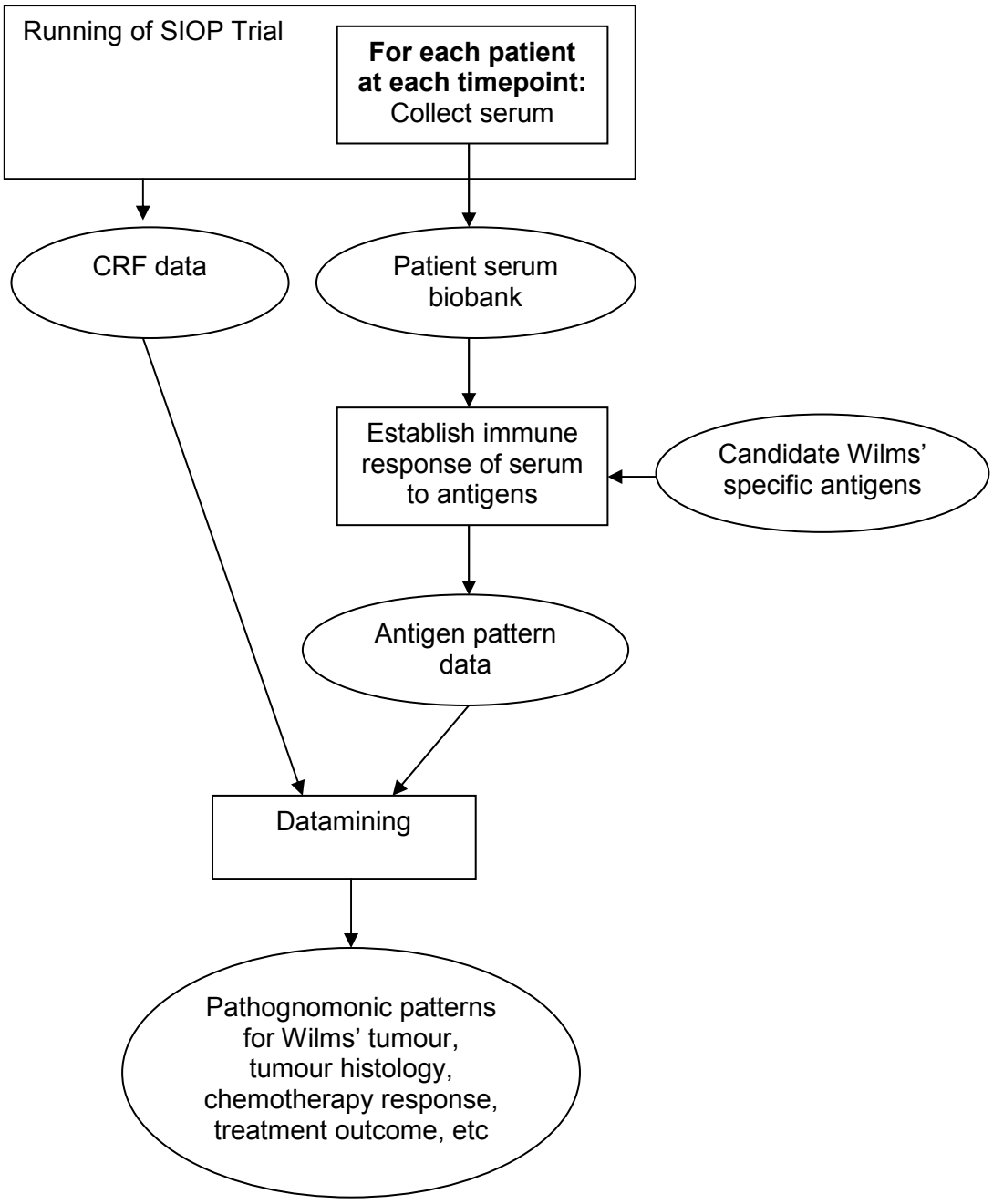


Figure 11 Overview of Step 2 of SC2

Figure 11 summarises Step 2 of the scenario. As part of the trial, for each patient serum is collected at different time points. Then, for each of the candidate Wilms' specific antigens identified in Step 1, the immune response is observed; either there is a response, or there isn't. Once this data has been obtained for a sufficiently large number of patients, it can be analysed together with the data that is being collected by way of CRFs as part of the normal running of the trial. It will be checked if the antigens are specific to Wilms' tumour. Furthermore, it is checked if there are meaningful correlations of the antigen pattern data with the data in the CRF database. It would be very valuable, for instance, if the antigen pattern at time of diagnosis can be used to identify the histology of the tumour.

Figure 12 shows at what time points patient serum is being collected for the purpose of the antigen scenario. This is done at four time points:

1. Diagnosis
2. Just before surgery (after pre-operative chemotherapy)
3. Just after surgery (before post-operative chemotherapy)
4. End of treatment

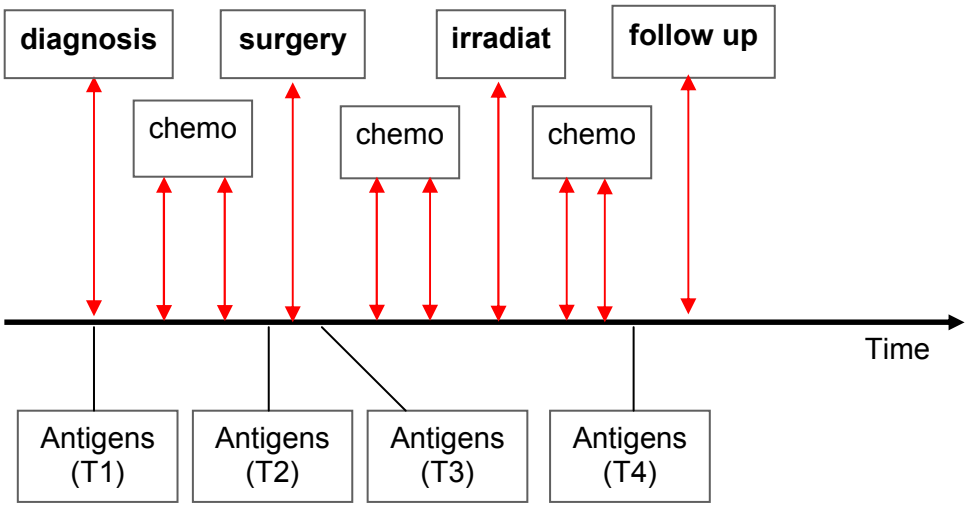


Figure 12 Timeline of antigen data collection as part of SIOP Trial

4 Functional View

The Functional View describes what the system should achieve. For the data access wrapper services, this for a large part depends on the way that the data is stored and the queries that need to be supported. These are therefore the subject of Section 4.1. Section 4.2 subsequently considers what the system should achieve, based on the scenarios that have been defined.

4.1 Data storage and queries

This section describes how data is stored, and typical queries that are being run on that data, for the TOP Trial and the SIOP Trial. This is done in respectively Section 4.1.1 and 4.1.2.

4.1.1 TOP Trial

4.1.1.1 Clinical data storage

For Jules Bordet Institute the clinical data is stored in an Oracle Clinical database, version 4.5.1.

At Jules Bordet Institute the electronic patient records are stored in the Hospital Information System (HIS). The patient records are not yet completely electronic, some information is still stored in analog format.

The available imaging devices (MRI, US, X-ray and PET/CT) produce DICOM-compliant digital imaging studies that are stored on a PACS. Also in this case, analog studies still exist. However, a digitiser is available. Finally, microarray data is produced and stored in flat files along with a protocol description.

The HIS is a relational database. A GUI called ORIBASE is introduced to facilitate interacting with the database and supports full-text search and also lab data search. ORIBASE can also retrieve corresponding DICOM studies from the PACS, and is integrated with the Bordet Cancer Registry, which stores extensive structured information about tumours diagnosed after 1/1/2000. ORIBASE structures the data as “virtual pages”.

In the current implementation it is difficult and laborious to integrate new document types when needed. In addition, the database complexity is growing, with larger data sets of different types being stored. Foreseen is the transition to a document-centric repository (based on HL7v3/CDA).

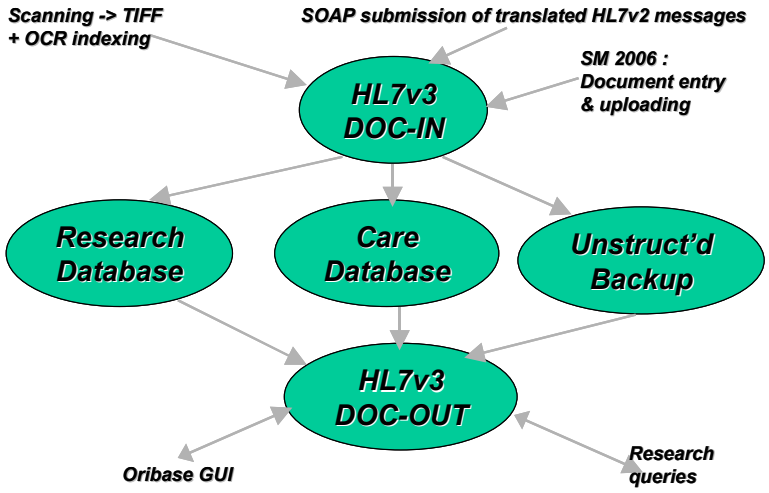


Figure 13 Proposed document-centric architecture

Figure 13 describes the envisioned architecture for the hospital information system at Jules Bordet Institute. Information will be entered according to the HL7 CDA specification. The care database is the database for clinical practice. Unstructured backup is the backup database. Research records will be stored in the research database. The system will be queried using HL7.

4.1.1.2 Imaging data

For each patient at least two breast ultrasound and two mammography examinations are being performed, before and at the end of the neo-adjuvant therapy. For some patients also MRI examinations have been done, and in the future PET-CT data will be acquired for consenting patients.

For the ultrasound examination, digital images, DICOM compliant, are available since 2005. The size of a total examination is of 5-10MB. Recently a Somo-vu scanner supporting three-dimensional imaging has been acquired and will be used to perform US examinations for the patients in the trial.

A complete mammography examination is composed of 4 images of either 28MB or 32MB each. Digital DICOM images are available since 2004.

MRI studies of 512x512x200 are acquired, with a total size of 100MB.

PET-CT studies are acquired in two ways, with attenuation or without. An examination consists of 200 slices of 128x128 or 512x512, resulting in a total size of 1MB or 16 MB.

4.1.2 SIOP Trial

4.1.2.1 Clinical data storage

The trial-specific clinical data is stored in a home-grown CDMS (Clinical Data Management System), hereafter referred to as Wilms CDMS. The data is stored in a Postgres database running on Linux. The front-end is using a Microsoft Access database running on Windows 2003. This is illustrated in Figure 14.

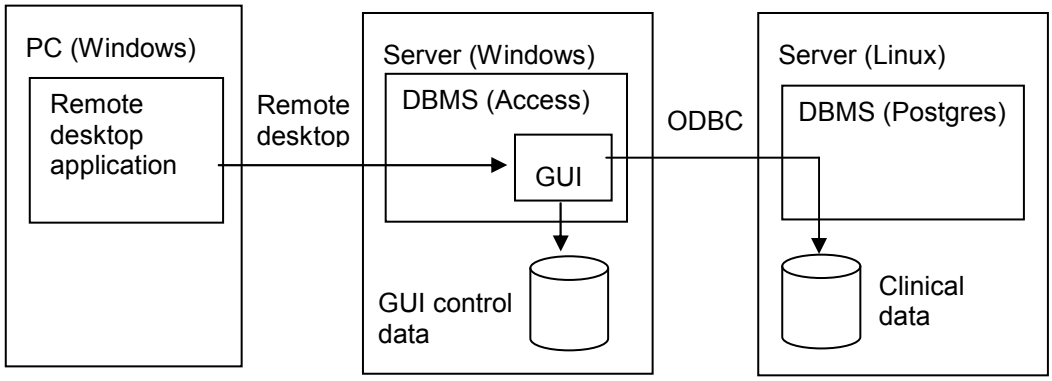


Figure 14 The clinical data management system for the SIOP Trial.

The database consists of 224 tables. These can be separated in the following groups:

- Code tables (165): Each table provides an enumeration of possible values that may appear in specific columns of other tables. For example, the table code_histotyp contains the names of the various tumour histology types that are distinguished.
- Control tables (15): These are used to control the way data is entered in other tables.

- Administration tables (4): These describe the users that have access to the database, including their roles and the clinics that they are associated with.
- Other tables: The remaining tables mostly store clinical data. These includes information about the patients, the data entered on the CRFs, information about the available imaging data, etc.

The code tables only have two columns each, but many tables are much larger. The largest table is f1 with 134 columns, and there are three other tables with more than a hundred columns. In total there are 1821 columns.

The main relationships between the various tables are summarised in Appendix 2. See Appendix 4 for full details of the database schema.

4.1.2.2 Clinical data queries

The Wilms CDMS has the ability to store and execute user-defined SQL queries. A few representative ones are shown in this section, to illustrate the type of queries that a clinician may want to execute. They also serve to give some further insight into the structure of the current Wilms CDMS.

Patients who are treated in the Homburg hospital

```
SELECT DISTINCT patient.siopnr, patient.klinik, patient.diag_klinik FROM
     patient WHERE (((patient.klinik)=17));
```

More specifically, the query retrieves all patients who are treated in the Homburg hospital who are enrolled in the Nephroblastoma trial. The latter is implicit, as only these patients are listed in the database that is queried. This holds for the other queries that follow as well.

The value of 17 represents the Homburg hospital, because it is the primary key for the entry of the Homburg hospital in the adm_klinik table. Note that the column names in the database are in German.

Pre-operative chemotherapy treatment details for patients with bilateral Wilms tumour

```
SELECT DISTINCT patient.siopnr, Min(chemo2.dat) AS von, Max(chemo2.dat)
     AS bis, Count(chemo2.act) AS [Anzahl von act], Sum(chemo2.act) AS
     [Summe von act], Count(chemo2.vcr) AS [Anzahl von vcr],
     Sum(chemo2.vcr) AS [Summe von vcr], Count(chemo2.dox) AS [Anzahl von
     dox], Sum(chemo2.dox) AS [Summe von dox], f2.gewicht AS naf2gewicht,
     chemo2.dosred, chemo2.grund_red, chemo2.dosred_com, f2.comment FROM
     f1 RIGHT JOIN (chemo2 RIGHT JOIN (f2 RIGHT JOIN patient ON f2.pnr =
     patient.pnr) ON chemo2.pidx = f2.idx) ON f1.pnr = patient.pnr WHERE
     (((f1.lokal)=3)) GROUP BY patient.siopnr, f2.gewicht, chemo2.dosred,
     chemo2.grund_red, chemo2.dosred_com, f2.comment;
```

The tables f1 and f2 are named after the CRFs whose data they store. Note that the column names of the result table are in German. This could of course be easily changed by modifying the SQL query.

Patients with Clear Cell Sarcoma of the Kidney that died

```
SELECT meldung.studname, patient.siopnr, f9.totdat, f9.status,
     f4_sub.histotyp FROM meldung RIGHT JOIN (f9 RIGHT JOIN ((f4 INNER
     JOIN f4_sub ON f4.idx = f4_sub.f4idx) INNER JOIN patient ON f4.pnr =
     patient.pnr) ON f9.pnr = f4.pnr) ON meldung.pnr = f4.pnr GROUP BY
     meldung.studname, patient.siopnr, f9.totdat, f9.status,
     f4_sub.histotyp HAVING (((f9.totdat) Is Not Null) AND
     ((f4_sub.histotyp)=12)) OR (((f9.status)=5) AND
     ((f4_sub.histotyp)=12));
```

For the column histotyp in the f4_sub table, the values are the primary keys in the code_histotype table. The primary key 12 corresponds to the value “hohe Malignität_Klarzellensarkom der Niere(CCSK)” in the histotype column of the code table.

Similarly, for the column status in the f9 table, the values are the primary keys in the code_status table. The primary key 5 corresponds to the value “Tod” in the status column of this code table.

Overall survival and other characteristics like age, histology, tumour volume of patients with unilateral Wilms tumour

```
SELECT patient.siopnr, ([f1].[diagdat]-[patient].[gebdat]) AS Alter_d,
     f1.diagdat, f1.lokal, f1.thermit, f1.metast, f1.us_v_r, f1.ct_v_r,
     f1.mrt_v_r, f1.us_v_l, f1.ct_v_l, f1.mrt_v_l, f1.protpat,
     f4_sub.pathtyp, f4_sub.histotyp, f4_sub.stadlok, Max([f9].[status])
     as verstorben, Max([f9].[l_unt]-[f1].[diagdat]) AS FUZeit_d,
     f9.todurs FROM f9 RIGHT JOIN (f4_sub RIGHT JOIN (f4 RIGHT JOIN (f1
     RIGHT JOIN patient ON f1.pnr = patient.pnr) ON f4.pnr = patient.pnr)
     ON f4_sub.f4idx = f4.idx) ON f9.pnr = patient.pnr GROUP BY
     patient.siopnr, ([f1].[diagdat]-[patient].[gebdat]), f1.diagdat,
     f1.lokal, f1.thermit, f1.metast, f1.us_v_r, f1.ct_v_r, f1.mrt_v_r,
     f1.us_v_l, f1.ct_v_l, f1.mrt_v_l, f1.protpat, f4_sub.pathtyp,
     f4_sub.histotyp, f4_sub.stadlok, f9.status, f4_sub.rel_his,
     f9.todurs HAVING (((f1.lokal)<>3) AND ((f1.metast)=1) AND
     ((f4_sub.histotyp)>1 And (f4_sub.histotyp)<12) AND ((f9.status)=1 Or
     (f9.status)=5) AND ((f4_sub.pathtyp)=1));
```

Characteristics of patients that received radiation treatment

```
SELECT patient.siopnr, patient.vorname, patient.name, F6.best_beg,
     F6.best_end, F6.best_organ, F6.best_organ_sonst FROM F6 INNER JOIN
     patient ON F6.pnr = patient.pnr;
```

Development of the tumour size during pre-operative chemotherapy for patients with a Rhabdoid Tumor

```
SELECT distinct patient.siopnr, [f2].[mrt_v_l]-[f1].[mrt_v_l] AS [Volumen
     Regression MRT links], [f2].[ct_v_l]-[f1].[ct_v_l] AS [Volumen
     Regression CT links], [f2].[us_v_l]-[f1].[us_v_l] AS [Volumen
     Regression US links], [f2].[mrt_v_r]-[f1].[mrt_v_r] AS [Volumen
     Regression MRT rechts], [f2].[ct_v_r]-[f1].[ct_v_r] AS [Volumen
     Regression cT rechts], [f2].[us_v_r]-[f1].[us_v_r] AS [Volumen
     Regression US rechts], f2.formid, f4_sub.histotyp FROM f4_sub RIGHT
     JOIN (f4 RIGHT JOIN (f2 RIGHT JOIN (f1 RIGHT JOIN patient ON f1.pnr
     = patient.pnr) ON f2.pnr = patient.pnr) ON f4.pnr = patient.pnr) ON
     f4_sub.f4idx = f4.idx WHERE (((f4_sub.histotyp)=13) AND
     ((f4_sub.pathtyp)=2 or (f4_sub.pathtyp)=3));
```

More specifically, this query shows for patients with a Rhabdoid Tumor the difference in tumour size (right and left kidney separately) before and after pre-operative chemotherapy according to the various imaging studies.

Characteristics of patients with unilateral Wilms tumour

```
SELECT patient.siopnr, (f1.diagdat-patient.gebdat) AS Alter_d,
     f1.diagdat, f1.lokal, f1.thermit, f1.metast, f1.us_v_r, f1.ct_v_r,
     f1.mrt_v_r, f1.us_v_l, f1.ct_v_l, f1.mrt_v_l, f1.protpat,
     f4_sub.pathtyp, f4_sub.histotyp, f4_sub.stadlok FROM f4_sub RIGHT
     JOIN (f4 RIGHT JOIN (f1 RIGHT JOIN patient ON f1.pnr = patient.pnr)
     ON f4.pnr = patient.pnr) ON f4_sub.f4idx = f4.idx GROUP BY
     patient.siopnr, (f1.diagdat-patient.gebdat), f1.diagdat, f1.lokal,
     f1.thermit, f1.metast, f1.us_v_r, f1.ct_v_r, f1.mrt_v_r, f1.us_v_l,
```

```
f1.ct_v_l, f1.mrt_v_l, f1.protpat, f4_sub.pathtyp, f4_sub.histotyp,
f4_sub.stadlok, f4_sub.rel_his HAVING (((f1.lokal)<>3) AND
((f1.metast)=1) AND ((f4_sub.histotyp)>1 And (f4_sub.histotyp)<12)
AND ((f4_sub.rel_his)=2));
```

More specifically, this query returns characteristics (age at time of diagnosis, date of diagnosis, right or left kidney involved, preoperative chemotherapy is given (yes or no), metastatic disease (yes or no), tumour volume at diagnosis and after preoperative chemotherapy, histology and local stage) of patients with unilateral Wilms tumour.

4.1.2.3 Imaging data

The way that image data is stored depends on the local hospital. Images could be available as hardcopy, on CD-ROM or online (PACS). The Wilms CDMS only stores which data is available, but does not include further information how to obtain/access the image data. Currently, hardcopies of images are sent to the Reference Center instead of using DICOM. Work is underway to use a centralised GPOH-wide repository server for managing the images digitally, although some practical problems still need to be solved. Firstly, there is the question to what extent the data needs to be pseudonymised. Secondly, there is aversion by some radiologists to give up “their” data.

4.2 Scenarios

4.2.1 SC2

This section considers the Nephroblastoma antigen identification scenario, which was already discussed in Section 3.2.1, in more detail. First, Section 4.2.1.1 describes the desired output of Step 1. Section 4.2.1.2 provides an overview of the public databases that may all need to be accessed in Step 1 of the scenario. Finally, Section 4.2.1.3 gives details about the way that the antigen data that is collected during Step 2 will be stored.

4.2.1.1 Step 1 – Desired output

Prof. Meese, molecular biologist at Saarland University, provided us with the following requirements on the antigen information to be retrieved in Step 1 of the scenario:

The overall goal is to find factors of the found antigens against Nephroblastoma, that are common to all antigens and that may be responsible for the specific antigen pattern.

Here is a list of criteria, that have to be in the output report of the scenario:

- Homology to known proteins listed ordered according to the grade of homology
- Comparison of the amino acid sequence of the different antigens, to classify the identified antigens according to their amino acid composition. The following listings should be provided in order:
    1. Contingent of positive and negative rests
    2. The isoelectric point (pI). The isoelectric point, pI, is the pH of an aqueous solution of an amino acid (or peptide) at which the molecules on average have no net charge. In other words, the positively charged groups are exactly balanced by the negatively charged groups.
    3. Percentage of aminoacid contingent
- Analysis of the antigens and listening according to their protein domains:
    - Globular domains

    - Zinc finger [Zinc finger DNA binding domain (ZnF_GATA). ZnF_GATA domain-containing proteins are typically transcription factors that usually bind to the DNA sequence [AT]GATA[AG] of promoters.]
    - GPI-anchor [A GPI anchor or glycosylphosphatidylinositol is a glycolipid that can be attached to the C-terminus of a protein during posttranslational modification. It is composed of a hydrophobic phosphatidyl inositol group linked through a carbohydrate containing linker (glucosamine and mannose linked to phosphoryl ethanolamine residue) to the C-terminal amino acid of a mature protein. The two fatty acids within the hydrophobic phosphatidyl-inositol group anchor the protein to the membrane.]
- Cellular localisation. Is the cellular localisation unknown, the above found results of the protein domains will help to define the localisation of the antigen with the highest probability:
    - Localisation signal of the nucleus
    - Transmembrane domain
    - etc.
- Comparison of all antigens in respect to repeated occurring structures.
- Localisation of the proteins in pathways with the help of Kegg, Biocarta. How often is an antigen found in a specific pathway.

The display of these results can be done as a structured list and table and should be stored. A graph is also useful and should be provided. The x-axis should show the identified antigens and the y-axis the analytical parameters. For each analytical parameter one graph. Or one graph should be provided, where the y-axis can be chosen. For the specific antigen all the parameters will be displayed as a point (marker) in the graph. The markers can be done in different shapes or colours.

According to these data the identified antigens can be clustered hierarchically in different groups. It should be possible to list criteria according to which the clusters can be defined or chosen. Such criteria are structural criteria as the domains, localisation, pathways, etc.

4.2.1.2 Step 1 – Public databases

Table 3 shows the public databases that have been identified by the clinical and bio-medical experts in the SIOP Trial as being relevant to Step 1 of the scenario. It includes resources that are provide very generic information about proteins (e.g. Swiss Prot, NCBI, GeneCards). These are commonly used by many bio-molecular scientists. Some databases on the other hand are more specific, such as for example the CAP database.

Desired Information	**Resource**	**Location**
Chromosomal localization, protein function, and sub cellular localization	National Center for Biotechnology Information	http://www.ncbi.nlm.nih.gov
	GeneCards	http://www.genecards.org/index.shtml
	EBI	http://www.ebi.ac.uk/
	Swiss Prot	http://www.ebi.ac.uk/swissprot/access.html

Pathways	Kegg pathway database	http://www.genome.jp/kegg/pathway.html
	Biocarta database	http://www.biocarta.com/search/index.asp
Domains	SMART database	http://smart.embl-heidelberg.de
Antigens in other tumours	Cancer Immunome database	http://www2.licr.org/CancerImmunomeDB/
	Cancer Associated Proteins (CAP) database	http://www.bioinf.uni-sb.de/CAP/
DNA to protein translation	Swiss Institute of Bioinformatics	http://www.expasy.org/tools/dna.html
Autoimmunity of antigens	Autoimmune database	http://www.wileyvch.de/contents/jc_2040/2005/25481_s.pdf

Table 3 Databases with potentially useful information about antigens identified in Step 1.

Most of these public databases only offer a web interface, designed for direct human interaction. For of few of these databases, for example those hosted by EBI, web service interfaces are available as well. This makes it much more straightforward for other software components, such as wrapper services, to query these databases. See for example, Section 6.3.1, where Step 1 of this scenario is carried out using only web service interfaces, so without using any of the (human) web interfaces.

4.2.1.3 Step 2 – Antigen data storage

The antigen data that is to be collected for each patient at each of the four time-points is small, and has a simple structure. For each antigen that is identified in Step 1, there is a boolean value that indicates if the antigen trigged an immune response with the serum. The number of antigens that will be tested for in Step 2 of the scenario is not finalised. This depends on the outcome of the SEREX experiment carried out in Step 1. The results of this experiment became available only recently (early November 2006). Thirty in-frame antigens were identified. As this number is relatively low, it is currently being considered to carry out another SEREX experiment with different pools of sera, to hopefully identify additional potentially interesting antigens.

As the set of antigens to test for not been fixed yet, no antigen serum analysis has yet been carried out in Step 2; so no data has been recorded yet. However, collection of serum has been taking place since August 2006, and at this moment (November 2006) serum has been collected for about fifteen patients. In fact, for none of these patients the treatment has yet been completed, so there is no serum yet for time point 4 (end of treatment).

There is no specific database to store the antigen patterns obtained in Step 2. However, as the data that is collected is simple, the idea is to initially collect the data in an Excel file.

4.3 Discussion

This chapter gave an overview of a variety of ways in which data is stored that needs to be accessed to carry out the various scenarios that have been identified. To summarise, the most important data sources are:

- *Relational databases*. Data collected on the CRFs in both the TOP Trial and SIOP Trial is stored in relational databases. It is important to note here that it is necessary but not

sufficient to know the schema of the database to phrase meaningful queries. The databases also use “meta tables”. These tables do not store data obtained during the trial, by way of CRFs, but instead have content that was created at design time of the database. An example are the code tables in the Wilms CDMS. Each row in the meta-tables represent specific questions on the CRFs, or specific answers to these questions. This means that the content and meaning of rows in these meta-tables needs to be known in order to express meaningful queries.

- *DICOM servers*. Medical image data is stored on DICOM servers. DICOM is the universally accepted standard for storing, transferring, accessing and otherwise using medical image data. It has its own information model and query support, which is will be introduced in Section 6.2.1.
- *Public web databases*. Although some offer a web-service interface, most only offer web interfaces geared towards human interaction. The lack of support for machine-to-machine interaction makes it more complicated to provide wrapper services for querying these. An interesting aspect of most of the sequence databases is that they support a very domain-specific querying. They support non-exact look up of sequences where an underlying biological model determines the amount of mismatch that is allowed.
- *Files in various formats*. Not all data is currently stored in databases. Some data is also stored in a more ad-hoc fashion, such as Excel files, CEL files, or plain text files. For some data, such as for example some of the biological analyses that are planned, it is not even certain yet how it will be stored and maintained.

Access to the data will have to be in accordance with the legal and ethical rules that apply. Due to the international nature of the ACGT project, legal and ethical experts are required to determine what exactly the legal and ethical constraints are. This is therefore a task of WP10. They will produce the initial ethical and legal requirements in month 12 of the ACGT project. Subsequently, we can analyse how this impacts the data access requirements for WP5. The next chapter, however, already outlines several patterns that could be used to meet these requirements.

5 Conceptual View

The Conceptual View is the first of the two views that describe the technical solution. It focuses on the concepts and sub-systems that are used, without yet specifying the underlying implementation details. This chapter currently consists of a single section, Section 5.1, which lists the main architectural patterns that are relevant.

5.1 Architectural patterns for data access

In providing access to medical data it is important that the privacy of patients is guaranteed. There are a number of techniques, or more generally, architectural patterns, that are useful in this context. The most important ones are introduced in the sections that follow. The aim is to provide a common terminology and build a common understanding in order to facilitate subsequent discussions about the requirements and potential solutions with the various stakeholders. Section 5.1.1 discusses how data can be anonymised and pseudonymised. Section 5.1.2 describes the basic mechanisms for accessing data residing inside a firewall.

5.1.1 Anonymisation and pseudonymisation

This section describes how data can be anonymised and pseudonymised. Section 5.1.1.1 begins by describing anonymisation. Subsequently, Section 5.1.1.2 describes the process of pseudonymisation, which is related. Section 5.1.1.3 then briefly describes how a trusted third party can be used for pseudonymisation. Finally, Section 5.1.1.4 compares the basic approach that can be taken when data from multiple sources has to be anonymised.

5.1.1.1 Anonymisation

Anonymisation is a means of ensuring anonymity. Anonymity of a person means that he or she is not identifiable. Anonymity needs to be defined from a given viewpoint, i.e. from the perspective of someone, taking into account the knowledge this person has and the information that is available to him [6].

For example, consider a DICOM medical image. It includes amongst others:

- the patient's name and date of birth, and
- a unique identifier (UID). The UID has been assigned by the DICOM modality that generated the image. It is a globally unique identifier for the image. IDs are constructed from a prefix registered to the supplier, vendor, or site, and a unique suffix that may be generated locally, e.g. from a date and time stamp. It has no other significance other than to ensure that the identifier is globally unique.

As is, the image obviously does not provide patient anonymity because it includes the patient's personal details. If the image is anonymised by removing the personal details, patient anonymity is already greatly increased. Although the personal details of the patient can still be obtained by querying the DICOM server, this server is typically only accessible to a few persons in the hospital with the appropriate credentials. Anonymity can be further increased by removing the unique identifier from the image. However, even then the patient could still be identified. For example, the diagnosing or treating clinician may recognise the patient, in particular if it was an interesting or unique medical case. So there are degrees of anonymisation. A minimum requirement is typically that anonymity must be ensured taking into account all information that is (readily) publicly available.

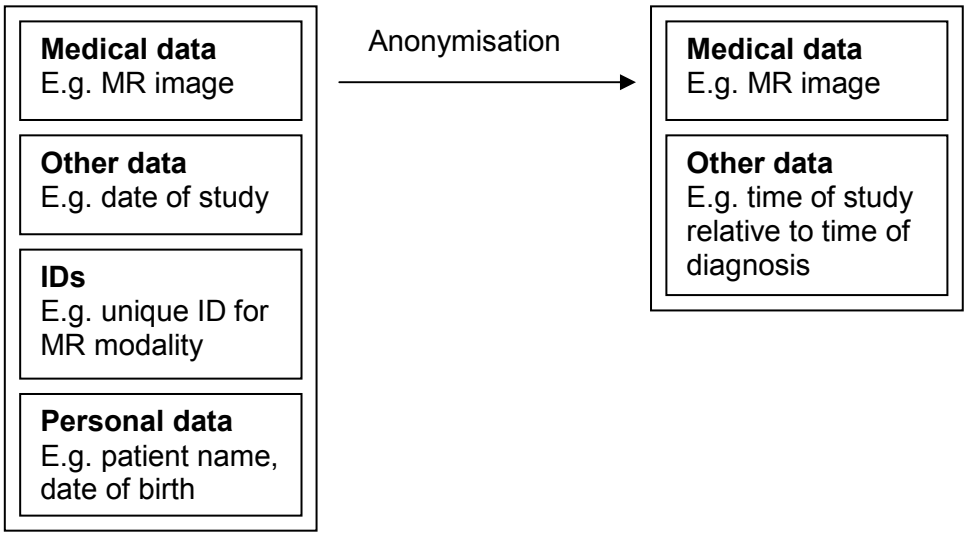


Figure 15 Anonymisation of data: Removing personal data and other identifiers from the data.

Figure 15 illustrates how anonymisation of medical data can be achieved. Here, not only personal information is stripped from the data, but also unique identifiers associated with the data. Although these identifiers do not directly reveal the identity of the person, they can be a means to obtain the identity as illustrated in the example above. Therefore, stronger anonymisation is achieved when all such identifiers are also removed from the data.

The figure also illustrates that anonymisation may involve transformation of data. For example, for analysis the date that a particular image is made is rarely needed. Instead, the time of the study relative to the time of diagnosis is typically sufficient. By omitting the date of the study, which reveals when the patient was being treated, anonymisation becomes stronger. Similarly, instead of a date of birth, an anonymiser could include the age of the patient at the time of the study. Especially when the time of study itself is omitted from the anonimised data, anonymisation is once again stronger while providing sufficient information to enable medical analysis.

5.1.1.2 Pseudonymisation

Strictly speaking, pseudonymisation is the process of assigning a pseudonym. A pseudonym is simply an identifier, such as a name or another combination of signs and symbols, including numbers. In the most simple case, a pseudonym refers to exactly one holder and it is not transferred to others over its lifetime [6]. This is what we assume here for simplicity.

Using a pseudonym does not imply anonymisation. For example, a pseudonym can be public from the outset. On the other hand, use of a pseudonym does not preclude anonymisation either. For example, if a pseudonym is chosen by a person himself, when used carefully, his anonymity can be guaranteed to the extent that nobody else can identify the person behind the pseudonym.

In this document, we will use pseudonymisation to refer to the process of assigning a pseudonym to anonymised data, as illustrated in Figure 16. In contrast, the anonymisation shown in Figure 15, where the resulting data does not carry any pseudonyms or other personal identifiers, will be referred to as strict anonymisation[1].

[1] It should be noted that our use of the terms “anonymisation” and “pseudonymisation” differs slightly from the use in D11.1 [24]. Our use of the term anonymisation describes the process of making data

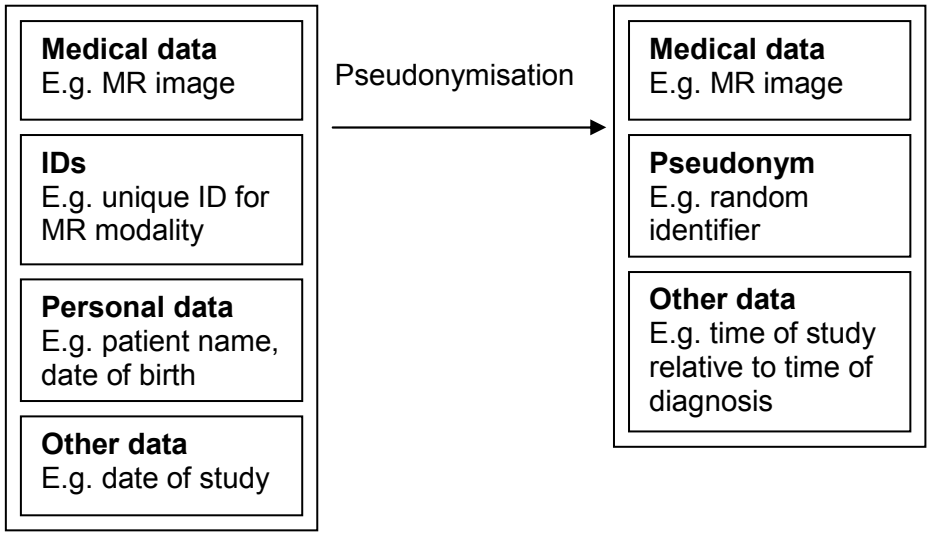


Figure 16 Pseudonymisation of data: Anonymising data while assigning a pseudonym.

5.1.1.3 Trusted third party

For all practical purposes, pseudonymisation involves recording for each pseudonym that is assigned the person to whom the pseudonym corresponds. If this relation is not stored anywhere and assuming the pseudonym does not include clues about the place and time where it has been generated, then the pseudonym does not (partly) identify a person and therefore fails to be a pseudonym.

Obviously, when a pseudonym is assigned to otherwise completely anonymised data, the records that link pseudonyms to persons need to be carefully protected, to protect the anonymity of the persons involved.

Records that link pseudonyms to persons can be used in various ways. Firstly, they can be used to ensure that the same pseudonym is used when multiple data associated with the same person is pseudonymised at different times. Secondly, when multiple pseudonyms are used for the same person, these records can be used to bring together all data associated with the same person. Thirdly, these records can also be used to break anonymity if need be. For example, when analysis of pseudonymised data reveals a few cases of patients who are at high risk and who would accordingly benefit greatly from a change of treatment, it can be justified to use these records to feed the results back to the clinicians treating these patients.

To enable these three uses while ensuring the anonymity of the persons involved, a trusted third party is typically used. It can perform the pseudonymisation, and will closely guard the records that map pseudonyms to persons. The trusted third party can receive requests from other parties that can only be carried out by the trusted third party, because it requires access to the pseudonymisation records. For each request, the trusted third party decides if it can be carried out, because either anonymity is preserved, or when this is not the case, the person requiring the data is authorised.

more anonymous, whereas in D11.1 the legal status of the resulting data determines whether data is anonymised. According to the definitions in D11.1, pseudonymised data is never considered anonymised. Anonymisation in D11. 1 more closely resembles what we refer to as “strict anonymisation” here.

5.1.1.4 Anonymising data from multiple sources

When there are multiple sets of data associated with the same person, there are various ways in which these can be anonymised. Firstly, there is a choice between strict anonymisation and pseudonymisation. Secondly, there is the option to anonymise all data at once, or to anonymise subsets of the data at different times. The latter may be convenient when data is collected at different times or at different places. Finally, when pseudonymisation is used, there is the possibility to assign the same pseudonym to all data associated with the same person, or to assign different pseudonyms to each of the subsets of anonymised data.

These different ways of anonymisation obviously have different implications, which are summarised in Table 4. When strict anonymisation is used to anonymise all data associated with a given patient, assuming the resulting data remains batched together, it can be analysed. For example, it could be used for In Silico experiments (see SC5). Here all patient data at the time of diagnosis together with details of the treatment could be used to configure the simulation, and later data (e.g. change of size of tumour after treatment) can subsequently be used to validate the simulation results. Although the data can be analysed, if the data was anonymised appropriately then there is no link to the patient so results can not be fed back. On the other hand, if pseudonymisation was used, the trusted third party can then be used to feed results back to the patient, as it maintains the link from the pseudonym back to the patient.

When data associated with a patient is anonymised at different times, it needs to be pseudonymised for analysis to be possible. If there is no pseudonym associated with the data, it is impossible to group all data together that corresponds to the same patient. When a single pseudonym is used, data can be grouped without intervention by the trusted third party. This is not the case when different pseudonyms are used, in which case the trusted third party is required before analysis can start.

<table>
<tr><th></th><th>Strict anonymisation</th><th colspan="2">Pseudonymisation</th></tr>
<tr><td>Performed at once</td><td>Analysis of data is possible. Results cannot be fed back to patient.</td><td colspan="2">Analysis of data is directly possible.
Feedback of results to patient requires TTP</td></tr>
<tr><td rowspan="2">Performed at different times</td><td rowspan="2">Analysis of data is not possible.</td><td>Single pseudonym</td><td>Feedback of results to patient requires TTP, but data can be analysed without using TTP.</td></tr>
<tr><td>Multiple pseudonyms per person</td><td>A TTP is needed before data can be analysed. Feedback to patients also requires TTP.</td></tr>
</table>

Table 4 The implications of different anonymisation schemes. (TTP = Trusted Third Party).

5.1.2 Accessing data inside a firewall

Trial-specific data is typically maintained inside a hospital, within the boundary of the hospital's firewall. An important question is how to make this data available to users of the ACGT platform who are outside this firewall, while maintaining patient privacy. Two contrasting approaches exist here. Data can be pushed outside the firewall from inside, or it

can be pulled from the outside by the client that needs it. These will be described in the next two sections.

5.1.2.1 Data push

Figure 17 illustrates how data can be pushed outside a firewall. Inside the hospital there is a master database, which stores all clinical data and is used within the hospital as part of the normal workflow. Alongside this database there is another one which is located outside the firewall. This external database stores an anonymised version of the data stored by the master database. Data is pushed from the master database outside the firewall through an anonymiser. How and when the external database is updated depends. Changes to the master database could be propagated as soon as they happen, but this may incur too much overhead. So typically, updates are batched. They could for instance take place once a day, which would typically be sufficient when the external database is only used for research purposes. As the mirror database is located outside the firewall it can be queried directly by clients with permission to do so.

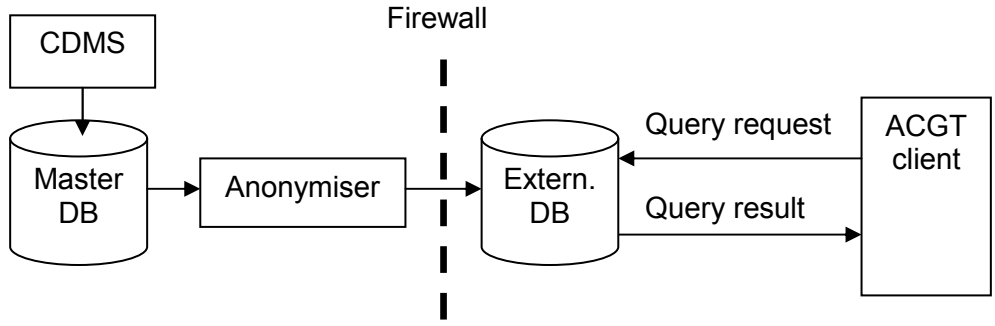


Figure 17 Using "data push" to make data available outside the hospital firewall.

5.1.2.2 Data pull

Figure 18 illustrates the data pull pattern for accessing data inside a firewall. It requires that the firewall can accept connections from the outside. Requests are handled by a handler dedicated to external requests. This is needed because the results that are returned by the hospital database need to be anonymised before they can be forwarded to the client that submitted the query.

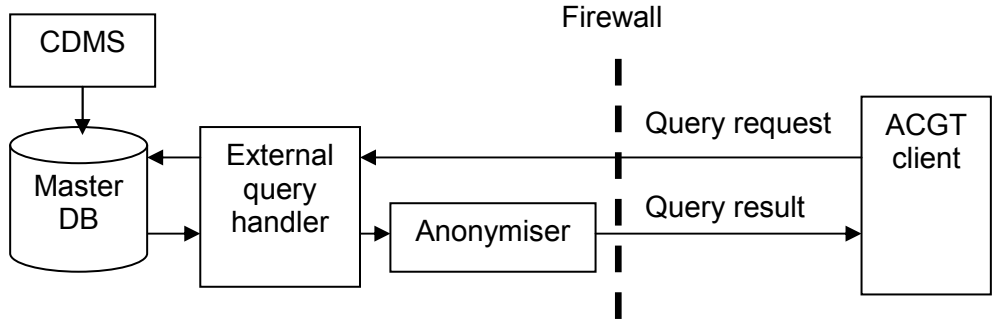


Figure 18 Using "data pull" to make data available outside the hospital firewall.

5.1.2.3 Pseudonymisation at the source

In both the data push and data pull patterns, the Master DB stores all clinical trial data (personal patient data, medical data, etc). In order to make the medical data available to the

ACGT client, while maintaining the privacy of patients, both patterns therefore require an anonymiser. The use of an anonymiser can be avoided by storing the data differently. Instead of a single trial database, two separate databases can be used. One database would store all personal data, whereas all other data would be stored in the second database. Data in the second database would only be linked to patient data in the first database by way of pseudonyms. This set-up makes the implementation of the CDMS software more complex. However, it has the advantage that sharing data with other clinical researchers is much simpler. By giving them access to the second database, they have all they data they need, while the privacy of patients is protected. This approach is illustrated in Figure 19.

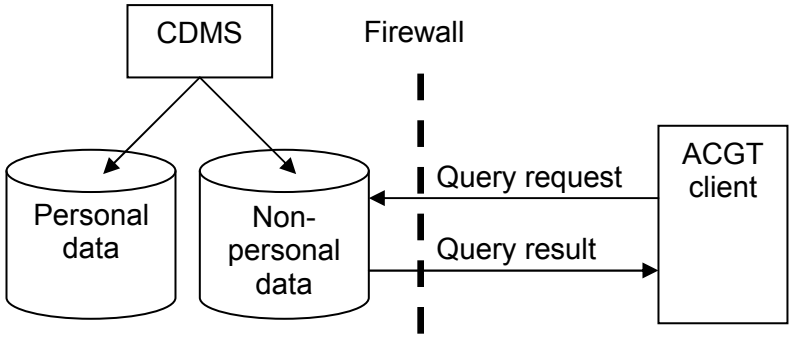


Figure 19 When data is separated into personal and non-personal from the outset, subsequent sharing does not require an anonymiser.

In the case of both ACGT trials, personal data is stored in the same database as non-personal data, which unfortunately means that it is more difficult to provide ACGT researchers access to the data. However, changing the existing set-up to that shown in Figure 19 would be even more complicated, as it amongst others requires large changes to the CDMS software. Therefore, this is not a realistic option and not considered further here. We only consider data access solutions that leave the current database and CDMS software unchanged, such as the pull and push patterns that have been introduced.

5.1.2.4 Comparison

Whether a data push or data pull solution is more appropriate depends on the situation. Table 5 lists the main differences that need to be taken into consideration.

An important difference is with respect to the configuration of the firewall that is required. With the data push pattern, the firewall needs to allow connections from inside the firewall to the outside, which is typically not a problem. In the basic data pull approach, however, the firewall needs to accept connections from the outside, which network administrators are typically very hesitant to allow as it reduces the security of the firewall. The data pull pattern can also be used without requiring the firewall to accept outside connections, but this requires a proxy outside the firewall. The proxy would accept requests from clients, and forwards them to the data handler inside the hospital, along a permanent connection initiated by the data handler.

Another difference is with respect to how up-to-date the results that the external client receives are. For the data push pattern the data is not necessarily up to date. The data is as recent as the time the last update to the mirror database occurred. For data pull, in contrast, the data is always up to date.

When data is pushed outside the firewall, it is anonymised when updates occur. In the case of data pull, however, it is anonymised once for each query. Which method incurs more overhead depends on the amount of querying compared to the amount of updates.

Finally, in the data push approach, the external queries do not put extra load on the main database, as they are entirely handled by the mirror database. This is not the case for the data pull approach, where each query is handled by the main database, whose load is increased as a result.

The choice between data push or data pull cannot yet be made. Both approaches may be used within ACGT, as which is best may vary per data source. For example, it depends on the local hospital security policies, as well as the approach taken to anonymise and pseudomise the data. The latter needs to be decided by WP11, taking into account the ethical and legal requirements to be produced by WP10.

	Data push	**Data pull**
Firewall	Connection to external database initiated from inside	Needs to accept outside connections (unless a proxy is used)
Data freshness	Data is not necessarily up to date	Received data is up to data
Anonymisation	Once for each update	Once for each query
Impact of external queries on load of master database	None	Direct impact. Load depends on number and type of external queries

Table 5 Summary of differences between data push and data pull approach for accessing firewalled data.

6 Realization View

The subject of this chapter is the Realization View, the last of the CAFCR views. Its focus is on the tools, standards and platforms that can be used to implement the solution outlined in the previous view. Section 6.1 describes web services in general, and discusses a few platforms and standards that are particularly relevant. Section 6.2 describes other relevant software and standards. Finally, Section 6.3 documents technology explorations that have been undertaken to better understand the capabilities and limitations of these technologies.

6.1 Web services

As described in Section 1.2.1, one of the main roles of WP5 is to provide data access interfaces to middleware and applications to be developed in other work packages. Therefore, the interfaces that need to be provided are machine-to-machine interfaces, as opposed to visual interfaces to end users. Web service standards are being developed to enable interoperable machine-to-machine interaction, and as they are very relevant to WP5, they are the subject of this section. First, a quick introduction is given to explain what web services are. Subsequently, a few web service standards and web service based technologies that are key to data access to bio-molecular tools and data are presented.

The W3C provides the following definition for a web service:

> "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."

There exist many standards related to web services, but the above definition includes the most important ones:

- XML: Extensible Mark-up Language. The syntax that is used by all web service standards.
- SOAP: An XML-based, extensible message envelope format. It has bindings to various underlying protocols, of which HTTP is the most commonly used.
- WSDL: An XML format that allows service interfaces to be described. It includes a definition of data types, messages, port-types (operations), binding (transmission details), and service (location).

Web service standards have been around since 2000 [13, 14, 15] and are still evolving. For instance, there has been a change of emphasis from RPC (Remote Procedure Call) based interaction towards document-based interactions. RPC-based interaction was initially popular because it is easily understood by developers and maps well to the implementation that the web service wraps. The latter, however, is also its drawback. When using the RPC-based paradigm, there is a tight coupling between the web service interface and the underlying implementation, which hinders interoperability. For that reason, there has been a push towards document-based interaction. This affected the standards, and implementations thereof, which also had some practical implications, as will be described in Section 6.3.2

6.1.1 Soaplab

Soaplab is a set of web services providing programmatic access to (supported) applications on remote computers [18]. Soaplab was developed in the European Bioinformatics Institute

(EBI), within the eScience initiative, as a component of the myGrid project. Many applications that are wrapped by Soaplab come from the EMBOSS package.

The Soaplab API is also used by the Gowlab sub-project, which can be used for providing access to ordinary web pages as web services. The ACD language is used for describing the command-line applications that are wrapped.

How Soaplab can be used is demonstrated in Section 6.3.1, where web services are used to carry out Step 1 of SC2.

6.1.2 EMBOSS

EMBOSS is the European Molecular Biology Open Software Suite [19]. EMBOSS is a free Open Source software analysis package specially developed for the needs of the molecular biology (e.g. EMBnet) user community. The software automatically copes with data in a variety of formats and even allows transparent retrieval of sequence data from the web. Also, as extensive libraries are provided with the package, it is a platform to allow other scientists to develop and release software in true open source spirit. EMBOSS also integrates a range of currently available packages and tools for sequence analysis into a seamless whole.

Section 6.3.1 demonstrates the use of various EMBOSS tools, which have been made available by EBI using Soaplab.

6.1.3 OGSA-DAI

OGSA-DAI [10, 11] is a specification for web service data access and integration. The aim of the OGSAI-DAI project is to develop middleware to assist with access and integration of data from disparate sources via the grid. More specifically, as stated on their website [10], OGSA-DAI is motivated by the need to:

- Allow different types of data resources - including relational, XML and files - to be exposed onto Grids.
- Provide a way of querying, updating, transforming and delivering data via web services.
- Provide access to data in a consistent, data resource-independent way.
- Allow metadata about data, and the data resources in which this data is stored, to be accessed.
- Support the integration of data from various data resources.
- Provide services that can be combined to provide higher-level web services that support data federation and distributed query processing.

Contribute to a future in which scientists can move away from technical issues such as handling data location, data structure, data transfer and integration and instead focus on application-specific data analysis and processing.

6.1.3.1 Concepts

The OGSA-DAI specification revolves around a number of concepts. These are introduced here, as it helps to explain OGSA-DAI's functionality and how it can be used.

A *data service* is a point of contact for clients who wish to access, query or update data resources. A data service offers a document-oriented interface which accepts OGSA-DAI perform documents from clients and returns response documents to clients.

A data service exposes zero or more *data service resources*. Each data service resource represents a specific data resource, for example a specific database. A data service resource accepts perform documents from data services, parses and validates them,

executes the data-related activities specified within them, and constructs response documents.

A *perform document* describes the actions that a data service resource should to take on behalf of the client. A *response document* describes the status of execution of a perform document and may contain result data, such as the results from a database query.

Each action in a perform document is known as an *activity*. In the perform document activities may be linked together so that data from one activity flows into another activity. This mechanism allows multiple potential interactions with a service to be encapsulated in a single client-server interaction. OGSA-DAI includes a large number of activities for performing common operations such as database queries, data transformations and data delivery. For example, there are activities for SQL queries, XSL-T transformations, and GridFTP data delivery.

Activities interact with a data resource by way of their *data resource accessor*. Data resource accessors control access to an underlying data resource. More generally data resource accessors hold state and provide methods that can be accessed by multiple activities and multiple instances of those activities.

If a perform document contains any end-points then the data service resource will process it synchronously. This means that result data will be returned in the response document to the client when the activity processing is complete. If however a perform document contains no end-points then there will be no result data to embed within the response document. In this case the data service resource is able to return the response document to the client immediately and then begin processing the perform document asynchronously.

Sessions provide the ability to store state across multiple requests to a data service resource. When a data service resource processes an activity described in one perform document, that activity may interact with a session and store some state. Then an activity specified in a subsequent perform document may interact with the same session to read, update or remove that state.

6.1.3.2 An example

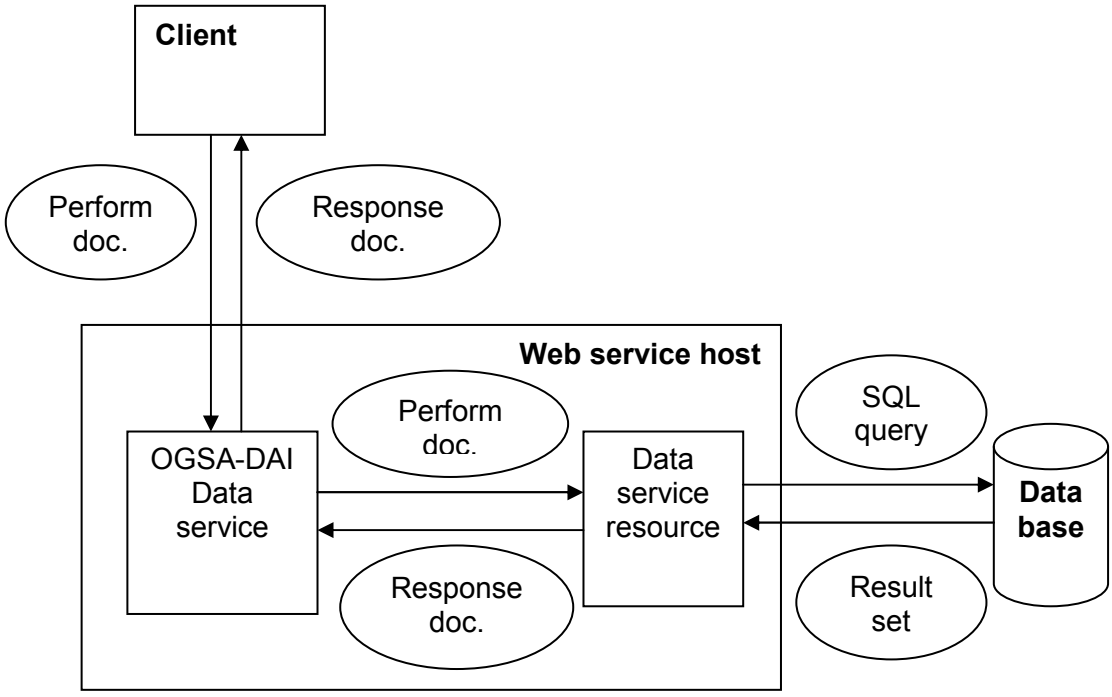


Figure 20 Example use of OGSA-DAI to query a relational database.

The following example, taken from the documentation at the OGSA-DAI website [10], shows how a simple SQL query can be carried out on a relational database using OGSA-DAI, as shown in Figure 20. The database could be hosted on the server that also hosts the OGSA-DAI web service, but this does not need to be the case.

Figure 21 shows a perform document that may be used. It describes a query whose results will be transformed into WebRowSet XML. There is one end-point so this perform document will be processed synchronously and the query results will be delivered within the response document. No session requirements are specified so the request will be joined to an implicit session.

```
<?xml version="1.0" encoding="UTF-8"?>
<perform xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">
  <documentation>
    Perform a simple SELECT statement and transform the
    results into WebRowSet XML.
  </documentation>
  <sqlQueryStatement name="myQuery"/>
    <expression>select * from littleblackbook where id=10</expression>
    <resultStream name="statementOutput"/>
  </sqlQueryStatement>
  <sqlResultsToXML name="webRowSet">
    <resultSet from="statementOutput"/>
    <webRowSet name="webRowSetOutput"/>
  </sqlResultsToXML>
</perform>
```

Figure 21 Example perform document for querying a relational database.

An activity pipeline is formed connecting the sqlQueryStatement and sqlResultsToXML activities. The query results are then transformed into WebRowSet XML and delivered back to the client in the response document. A possible response is shown in Figure 22.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:response xmlns:ns1="http://ogsadai.org.uk/namespaces/2005/10/types">
  <ns1:session id="session-ogsadai-106efa15ca3"/>
  <ns1:request status="PROCESSING"/>
  <ns1:result name="myQuery" status="COMPLETED"/>
  <ns1:result name="webRowSet" status="COMPLETED"/>
  <ns1:result name="webRowSetOutput" status="COMPLETED"/>
    <![CDATA[<?xml version="1.0" encoding="UTF-8"?>
      <webRowSet schemaLocation="http://java.sun.com/xml/ns/jdbc
        http://java.sun.com/xml/ns/jdbc/webrowset.xsd">
      ...
      <currentRow>
         <columnValue>10</columnValue>
         <columnValue>John Smith</columnValue>
         <columnValue>123 Some Lane, AnyTown</columnValue>
         <columnValue>0131-555-1234</columnValue>
      </currentRow>
      ...
    ]]>
  </ns1:result>
</ns1:response>
```

Figure 22 An example response document containing the results of an SQL query.

The <result> element with name “webRowSetOutput” corresponds with the end-point (or unconnected output stream) in the perform document. The query results are inserted into a

CDATA child node of the <result> element. Also note that there is a <result> element for each activity, which simply reports the status of the activity.

6.1.3.3 Implementations

OGSA-DAI provides web services compliant with two popular web services specifications: Web Services Inter-operability (WS-I) and Web Services Resource Framework (WSRF). A specific OGSA-DAI distribution will conform to only one of the above standards and use a specific implementation of it. There are, for example, differences in the way that data resources are addressed. In OGSA-DAI WS-I it is specified in postfix to the service URL. In OGSA-DAI WSRF the resource is specified in WS-Addressing information added to the SOAP header of the operation request message. However, these differences are hidden when the client toolkit is used to access OGSA-DAI resources.

6.1.4 OGSA-WebDB

OGSA-WebDB is a system that provides grid service interfaces to existing web databases [9]. It provides applications with an OGSA-DAI web interface, which the application can use to submit queries expressed in SQL. These are then forwarded to the appropriate public database. Because these databases typically only have a web interface, the query needs to be converted from SQL to Boolean conjunctions of keyword and field search conditions. Furthermore, a single query may involve more than one public database. So handling queries may involve joining datasets from multiple databases, which also involves query scheduling and optimisation. Finally, OGSA-WebDB also performs semantic mediation in order to hide semantic differences that may exist between the various public databases.

OGSA-WebDB uses a proxy database to store the intermediate results from the public databases and to aggregate these results before returning them to the application. When an SQL query is received, it is first analysed and decomposed into one or more sub-queries to the public databases. These sub-queries, which are simple Boolean conjunctions of keywords and field search conditions, are forwarded to the public databases, using a dedicated wrapper for each database. These wrappers are responsible for submitting the queries to the web interface and for parsing the resulting HTML output. These results are then stored in the proxy database, where a relational table represents each web database. Once all sub-queries have been executed, the original query is executed on the proxy database, which may then join the data from multiple tables. These results are returned to the user by way of OGSA-DAI, which means that the user has a choice of delivery methods.

The OGSA-WebDB technology is certainly relevant to the ACGT project. OGSA-WebDB has for instance already been used for querying bio-medical databases specifically in the area of cancer prevention. There are also similarities in approach. The OGSA-WebDB system uses a mediator and wrappers, which are also key components in the ACGT integration architecture. On the other hand, OGSA-WebDB does not use an ontology, so here the ACGT project is more ambitious. Furthermore, the focus of OGSA-WebDB is on querying public (web) databases, whereas access to trial-specific data is a large part of ACGT, which results in many additional challenges. Finally, within ACGT, there is also the need to access other types of data, such as image data and microarray data, which means that the proxy architecture used by OGSA-WebDB may not be appropriate here.

6.2 Other software and standards

Section 6.2.1 gives a brief introduction to the DICOM standard. It aims to provide the basic background necessary to understand Section 6.3.3, which documents our experiences of using OGSA-DAI to query DICOM servers. Section 6.2.2 gives a short overview of the BASE database, which can be used for storing all data related to microarray experiments.

6.2.1 DICOM

The Digital Imaging and Communications in Medicine (DICOM) standard was created by the National Electrical Manufacturers Association (NEMA) to aid the distribution and viewing of medical images, such as CT scans, MRIs, and ultrasound [16, 17]. Amongst others, the standard defines how images can be stored, transferred, viewed and printed. The standard is specifically designed for the medical domain and closely reflects how images are used within hospitals. For example, for diagnostic purposes it is important that images look the same when viewed on different devices, which the standard therefore tries to guarantee. DICOM images not only include the image itself, but also other information relevant to diagnosis. A DICOM image can for example include details of the patient (name, sex, date of birth, etc), details of the study that the image is part of, and details of the acquisition (the modality that was used, acquisition settings, etc). Different types of modality generate different types of images. For each modality the DICOM standard specifies how the image data is stored, and what additional, potentially modality specific, data can or must be included.

6.2.1.1 Concepts and terminology

DICOM uses *Service Classes* to describe interactions between a pair of devices. Two roles are distinguished: *Service Class User (SCU)* and *Service Class Provider (SCP)*. A large part of the Service Class is the description of information and related operations. These are defined by *Service Object Pair (SOP) Classes*. In each SOP Class definition a single Information Object Definition is combined with a Service Group, containing one or more services.

Information Object Descriptions standardise the format and content of information that is being exchanged. An *Information Object Descriptions (IOD)* is an abstract description of a class of entities. It can for example represent an MR image, a CT image, or a structured report. For each IOD, the standard defines required and optional *attributes* that can be used to describe information objects. A distinction is made between normalised IODs and composite IODs. A *normalised IOD* represents a single entity in the DICOM model of the real world. It contains only attributes inherent in the entity it represents. A *composite IOD*, on the other hand, represents parts of several entities in the DICOM application model. For example, the Computed Tomography IOD contains both attributes which are inherent in the image (e.g. image date) and attributes which are related to but not inherent in the image (e.g. patient name). It is therefore a composite IOD.

A *Service Group* combines related services. *DICOM Message Service Element (DIMSE) Services* are the actions that can act upon information objects. DIMSE services are classified as either DIMSE-N or DIMSE-C, depending on whether they are applicable to, respectively, normalised or composite IODs. For example, C-STORE is a DIMSE-C service that can be used to store images.

Figure 23 summarises how these main concepts relate.

6.2.1.2 DICOM Identifiers

Each DICOM image is a assigned a *unique identifier (UID)* that is globally unique. A UID is a string of numeric characters and dots, with a maximum length of 64. The dots are used to hierarchically structure the UID space. For example, each DICOM vendor is assigned its own UID. From this, each vendor can generate derived UIDs, according to its own scheme, by appending additional numbers and dots to its vendor-specific root identifier. UIDs are used for various purposes. For example, each modality has its own UID, which it can subsequently use to generate a UID for each image that it produces. Furthermore, UIDs are also used to identify parts of the DICOM standard, such as SOP classes. This is used to unambiguously state which part of the DICOM standard specific Application Entities support.

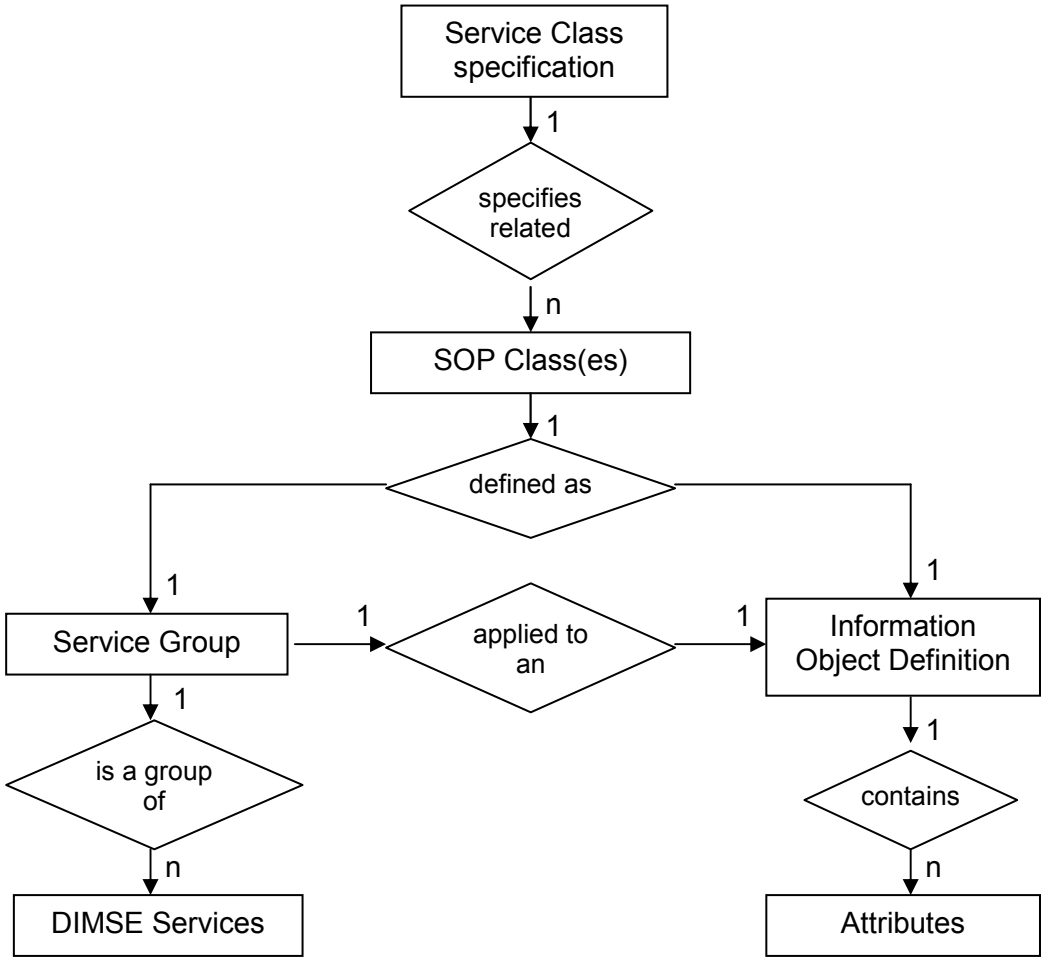


Figure 23 Major concepts used in the specification of the DICOM standard.

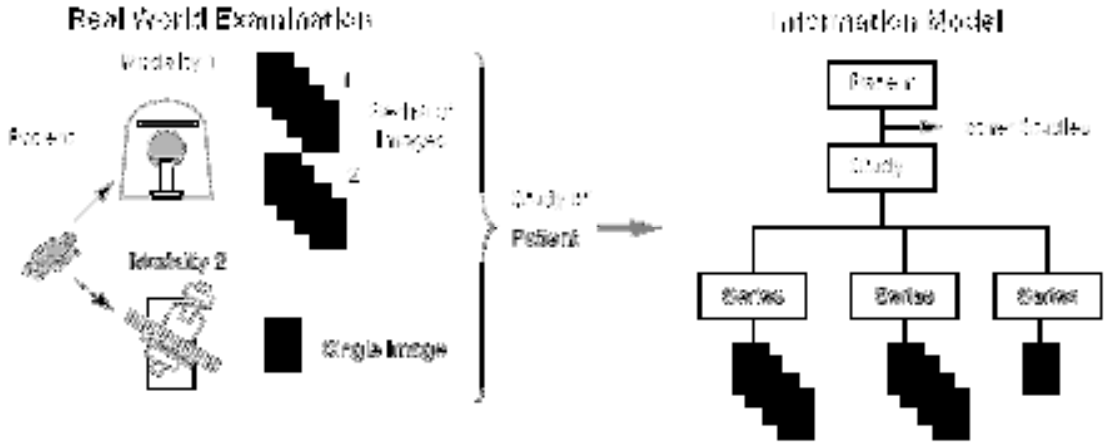


Figure 24 The DICOM information model for examinations and its mapping to the real world.

6.2.1.3 Querying and retrieval of images

Querying and retrieval of images is defined in the DICOM *Query/Retrieve (Q/R) Service Class*. The types of queries that are allowed are not complex. It is focused towards basic composite object instance information queries using a small set of common key attributes.

Figure 24 shows the information model that underlies the DICOM querying functionality. It has the Patient IOD at the root. It contains one or more studies, which correspond to medical examinations that the patient has undergone. It may involve studies at different modalities. A study contains one or more series. A series is a collection of related images coming from a single modality. The way the images are grouped depends on the clinical usage. A series can consists of a single image, or a series of images. For example, CT scans produce series of images where each image represents a "slice" of the body. In other cases, images in a series can show the same part of the body at different times. For example, an multiframe ultrasound image may show an animation of a beating heart.

When querying, there are three information models that can be used: Patient Root, Study Root, Patient/Study Only. The *Patient Root Q/R information model* is based upon the four level hierarchy shown in Figure 24. The *Study Root Q/R information model* is similar, except that the top level is the study level. Attributes of patients are considered to be attributes of studies. The *Patient/Study Only Q/R information model* is also similar to the Patient Root model, except that it only supports the two levels at the top. The models determine the type of queries that can be issued, but do not directly restrict what can be returned. For example, even though the Patient/Study Only model does not include images, images can still be retrieved by retrieving all images for a specific patient.

For each of the three models the standard defines, for each level in the model, the attributes that can be searched for in the query. The SCP of the Q/R Service Class does not have to support all attributes. For each attribute it is stated whether they are required or optional. Most attributes are optional. For example, in the Patient Root model, the only two attributes that are required are Patient's Name and Patient ID. All other attributes at the patient level, such as Patient's Birth Date and Patient's Sex, are optional. Different types of matching are supported, partly depending on the type of each attribute. The following matching types are supported:

- *Single Value Matching*: Exact matching of the values. For most attributes the matching is case-sensitive, but for those with a PN value representation, such as Patient Name, the application may perform case-sensitive matching, in an application specific way to be specified in the conformance statement.
- *List of UID Matching*: A list of UIDs is provided, each of which may generate a match.
- *Universal Matching*: All entities match the given attribute.
- *Wild Card Matching*: For plain-text string values, the "*" and "?" wildcard characters can be specified in the query, where "*" matches any number of characters, and "?" matches any single character.
- *Range Matching*: For date, time, and datetime values, a range can be specified.
- *Sequence Matching*: When the attribute type is a sequence of items, one or more items may be specified in the query, each of which should match.

The Query/Retrieve Service Class uses the C-FIND, C-MOVE and C-GET services. The *C-FIND service* allows a client to query a server for matches on one or more attributes. The server returns object instance identifies of matching records to the client. Subsequently, the client can use the C-GET and C-MOVE services allow the transfer of images. The *C-GET service* can be used to retrieve a set images, where the issuer of the request is the destination. The *C-MOVE service*, on the other hand, allows third parties to initiate the

transfer of images between two locations. For example, an imaging station may use it to initiate the transfer of images from a scanner to an archive. For both the C-GET and C-MOVE service, the receiver of the image needs to support the DICOM standard (it respectively needs to be a SCU for the Query/Retrieve Service Class, or a SCP for the Store Service Class). To make it possible for applications or services that do not support the DICOM standard to retrieve images, the WADO service was added to the DICOM standard.

6.2.1.4 WADO

The *Web Access to DICOM Objects (WADO) Service* enables the web clients to retrieve DICOM Persistent Objects managed by a web-enabled DICOM server using the HTTP protocol. Data can be retrieved either in a presentation-ready form as specified by the requestor (e.g., JPEG or GIF) or in a native DICOM format. The WADO server may also perform additional operations before returning an image, such as scaling, anonymisation or annotation. The WADO service does not support querying; the client needs to provide the UID of the image. Furthermore, only one image can be retrieved per request.

6.2.1.5 MEDICUS

MEDICUS (Medical Imaging and Computing for Unified Information Sharing) is a Globus Alliance Incubator project to federate Medical Imaging and Computing Resources for clinical and research applications [20, 21]. It provides the ability to federate and to share large medical images using the DICOM standard within data grids using the Globus Toolkit. It adheres to the WSRF and OGSA standards. For instance, along side the standard DICOM Q/R interface, it also offers an OGSA-DAI inferface for querying images that are stored on the grid. MyProxy, Shibboleth, and SAML are used for management of credentials and user role assertions. MEDICUS also includes some support for anonymisation, as it offers some privacy-preserving DICOM query/retrieve and image storage functionality.

The DICOM Grid Interface Service (DGIS) is the first output of the MEDICUS project. DGIS consists of three service entities. A gateway for accessing the DICOM images on the grid, a storage service provider (SSP) for image storage and indexing, and meta catalog service (MCS) for image referencing. Each entity requires a specific setup. A typical deployment of DGIS uses one or many gateways and a single SSP and a single MCS server. Each hospital that uses MEDICUS accesses the DGIS functionality through a gateway.

The MEDICUS project is still relatively young. The first version of the software was released in August 2006, and only very recently (31 December 2006) a second version was released. There is documentation about how to install the various MEDICUS servers, but not yet any developer documentation or documentation about how to use it. Furthermore, there is not yet a visible developer community (there are no public forums, accessible mailing list archives, bug trackers, etc). Despite this immaturity, it is recommended to keep an eye on the project.

6.2.2 BASE

BASE (BioArray Software Environment) is a comprehensive free web-based database solution for the massive amounts of data generated by microarray analysis [7, 8]. BASE is developed and maintained by Lund University, one of the ACGT partners. BASE is available from http://base.thep.lu.se.

The BASE database is not merely storing results from microarray experiments; it can also track the entire workflow of microarray experiments. This includes the ability to track the array production process for laboratories that produce their own microarrays (LIMS). However, it should be noted that there is no requirement on using the LIMS features.

The BASE software has been completely rewritten. BASE 2 is the new version that is being actively developed and extended, whereas BASE v1.2 is the previous version. Table 6

describes the main concepts used by BASE. This should give an idea of the workflow that is supported, and should also help to understand the database schema. If there is a name given within brackets, then the data is stored in a different table in BASE 2. The new name is then given within brackets.

Concept	**Description**
Reporter	A probe on a microarray (a “probeset” in Affymetrix terms).
ArraySlide	A particular instance of a microarray slide.
ArrayType (ArrayDesign)	A particular microarray design.
ArrayBatch	A set of microarrays of the same design that are created at the same time.
Plate	A plate represents a physical microtiter plate, possibly only at some stage in its existence. Microtiter plates are used to create array slides. They are often produced through multi-step procedures, so that a chain of plates describes a single microarray design.
Sample (BioMaterial)	Biological material, for example tumour tissue obtained during operation.
Extract (BioMaterial)	An extract is derived from a sample. A sample can be used to create multiple extracts. An extract can have a quantity associated with it.
LabeledExtract (BioMaterial)	A (part of an) extract used for a specific hybridisation.
Hybridization	The result after hybridising a microarray with DNA from one or more labelled extracts.
Image	The scan of a hybridisation. One hybridisation can be scanned in more than one scanner and/or under different settings.
RawBioAssay	The raw data that results from hybridisation. It is obtained from the scanned software by third-party software. BASE only imports the resulting files, *i.e.*, scanning is performed outside of BASE.
BioAssay	When analysing data, you work with "BioAssays" rather than raw data. BioAssays consist of intensity values only (e.g. Int1 and int2 for two-channel raw data). BioAssays are created from Raw data sets and will continue to be associated with its corresponding Raw data set throughout analysis.
BioAssaySet	A compilation of BioAssays to a “matrix” of gene rows and sample columns.

Experiment	An Experiment is a collection of raw data sets and any associated analysis steps that has been performed on these raw data sets. Experiments may for instance involve transformations and filtering of data.

Table 6 The main concepts used in the BASE database. Each corresponds to a table in the database. Names within parenthesis indicates name change between BASE 1.2 and 2.

The main relationships between the various tables in the BASE database are summarised in Appendix 3. See Appendix 5 for full details of the database schema.

6.3 Technology explorations

6.3.1 Web service exploration of SC2

Step 1 of SC2 involves the use of public databases to characterise antigens found using the initial SEREX experiment. In the description of the scenario in D2.1 (see Section 5.3.6.1) web interfaces are used to carry out the various steps of the workflow. This is a menial process, especially when you take into account that it has to be executed for each of the antigens identified using the SEREX experiment. Therefore, we would like to be able to fully automate the process in the ACGT environment. This requires machine-machine interaction, and therefore it is natural to use web service interfaces when possible. To investigate the feasibility of using only web services to carry out Step 1 of the scenario, we attempted to do just that. The results are documented below.

6.3.1.1 Step by step description

To translate the SEREX nucleotide sequence into the six different possible reading frames, the transeq tool that is part of the EMBOSS package can be used. It is available as a service through Soaplab, and hosted by the EBI institute. To invoke the web service the AnalysisClient, which is implemented in Java and also provided by the EBI, is used. The client is executed by way of a script file. The script file was modified to hardcode the location of the Java executable, as well as the proxy settings needed to go through our local firewall. The resulting invocation is then:

```
run/run-analysis -e http://www.ebi.ac.uk/soaplab/services -name
     nucleic_translation.transeq frame_6 yes sequence_direct_data
     :MPMGp800A01579Q231.txt -w -r outseq
```

It returns, as expected, a text file with the protein sequences corresponding to each of the six reading frames.

To blast the nucleotide sequence of positive SEREX clone, the WU-Blast2 web service hosted by EBI was used. To invoke it, one of the clients also provided by EBI was used. The chosen client is implemented in Java and using the Apache Axis web services toolkit. The invocation is as follows:

```
java -Djava.ext.dirs=lib -jar WSWUBlast.jar --email
     erwin.bonsma@philips.com --program blastn --database embl --sequence
     MPMGp800A01579Q231.txt --sensitivity normal --numal 10 --scores 100
     --exp 10 --stats sump
```

Note, the default parameter settings for the web service client are not sensible. They need to be explicitly set as shown above (or something similar), or the matching will fail entirely. These settings are based on the default settings of the normal BLAST web interface.

When the BLAST request is appropriately configured, it returns a text file with the hundred best matching sequences that were found. Ranked at the top is a group of highly scoring sequences, each with similar matching scores. One of these is the following:

```
EM_HUM:AJ404614; AJ404614 Homo sapiens mRNA for B-cell ly...
```

The entry for this gene can subsequently be looked up in the Ensemble database. For this, the WSDbFetch web service hosted by the EBI can be used. Also here, EBI has provided a Java client that uses the Axis web services toolkit. Using this client, the web service can be invoked as follows:

```
java -Djava.ext.dirs=lib -jar WSDbfetch.jar fetchData embl:AJ404614
```

It returns a text file with the Ensemble entry for the AJ404614 gene. The text file includes a cross-reference to the entry of the corresponding protein in the UniProtKB/Swiss-Prot database

```
FT                   /db_xref="UniProtKB/Swiss-Prot:Q9C0K0"
```

The same web service, and client, can be used to look up this protein entry:

```
java -Djava.ext.dirs=lib -jar WSDbfetch.jar fetchData uniprot:Q9C0K0
```

It returns a text file with the information stored in the UniProt database. It includes the amino acid sequence of the protein:

```
SQ   SEQUENCE   894 AA;  95519 MW;  DF6C467AE2EEC122 CRC64;
     MSRRKQGNPQ HLSQRELITP EADHVEAAIL EEDEGLEIEE PSGLGLMVGG PDPDLLTCGQ
     CQMNFPLGDI LVFIEHKRKQ CGGSLGACYD KALDKDSPPP SSRSELRKVS EPVEIGIQVT
     PDEDDHLLSP TKGICPKQEN IAGPCRPAQL PAVAPIAASS HPHSSVITSP LRALGALPPC
     LPLPCCSARP VSGDGTQGEG QTEAPFGCQC QLSGKDEPSS YICTTCKQPF NSAWFLLQHA
     QNTHGFRIYL EPGPASSSLT PRLTIPPPLG PEAVAQSPLM NFLGDSNPFN LLRMTGPILR
     DHPGFGEGRL PGTPPLFSPP PRHHLDPHRL SAEEMGLVAQ HPSAFDRVMR LNPMAIDSPA
     MDFSRRLREL AGNSSTPPPV SPGRGNPMHR LLNPFQPSPK SPFLSTPPLP PMPPGGTPPP
     QPPAKSKSCE FCGKTFKFQS NLIVHRRSHT GEKPYKCQLC DHACSQASKL KRHMKTHMHK
     AGSLAGRSDD GLSAASSPEP GTSELAGEGL KAADGDFRHH ESDPSLGHEP EEEDEEEEEE
     EEELLLENES RPESSFSMDS ELSRNRENGG GGVPGVPGAG GGAAKALADE KALVLGKVME
     NVGLGALPQY GELLADKQKR GAFLKRAAGG GDAGDDDDAG GCGDAGAGGA VNGRGGGFAP
     GTEPFPGLFP RKPAPLPSPG LNSAAKRIKV EKDLELPPAA LIPSENVYSQ WLVGYAASRH
     FMKDPFLGFT DARQSPFATS SEHSSENGSL RFSTPPGDLL DGGLSGRSGT ASGGSTPHLG
     GPGPGRPSSK EGRRSDTCEY CGKVFKNCSN LTVHRRSHTG ERPYKCELCN YACAQSSKLT
     RHMKTHGQIG KEVYRCDICQ MPFSVYSTLE KHMKKWHGEH LLTNDVKIEQ AERS
```

To subsequently match the protein sequence with the original SEREX sequence, local alignment tools that are part of EMBOSS can be used. For example, the water and matcher tools can both be used. Like the transeq tool described earlier, these tools are also made available as web services by EBI using Soaplab.

The water alignment tool can be invoked as follows:

```
run/run-analysis -e http://www.ebi.ac.uk/soaplab/services -name
     alignment_local.water asequence_direct_data :transeq-frame_1.txt
     bsequence_direct_data :uniprot-Q9C0K0.txt -w -r
```

This matches the first 5'3' Frame with the protein sequence, and results in a low score: 77.0. Similarly matching the second 5’3’ Frame gives a much higher score of 734.0.

Similar results are obtained when using the matcher tool:

```
run/run-analysis -e http://www.ebi.ac.uk/soaplab/services -name
     alignment_local.matcher asequence_direct_data :transeq-frame_1.txt
     bsequence_direct_data :uniprot-Q9C0K0.txt -w -r outfile
```

The first 5'3' Frame returns a low score of 46, whereas the second 5'3' Frame returns a high score of 642. This is as expected, because the second frame indeed contained the characteristic cloning sequences.

6.3.1.2 Conclusion

The main workflow of Step 1 of SC2 could be successfully carried out using only web services. There are further characterisations that could be done on the genes and proteins, such as obtaining pathway and domain information. For this initial exploration, these queries were considered out of scope. It is, however, to be expected that some of these more advanced and specific queries can not that easily be carried out using web services. The reason is simply that not all public gene and protein databases are yet providing web service interfaces. However, this initial exploration indicates that for the most common bio-molecular workflow steps there is good web services support. In the cases where a specific database does not offer a web service interface, it may still be possible to query it using web services nonetheless. For instance, the GowLab toolkit can be used to provide web service interfaces around services that have a plain web interface.

All web services that were used were provided by the EBI. This is not mere coincidence. Although other institutes host bio-molecular web services as well, the EBI offers a wide variety of services that are well supported. This made it convenient to use their web services whenever this was possible.

Each web service was invoked using a dedicated client that was provided along-side it. In each case, it was a fairly basic client that can be invoked from the command line. When, however, one or more web services need to be integrated into a larger application, these clients are not suitable, and one has to develop specific client interfaces for invoking the web service(s). In this case, interoperability becomes more of an issue, and this is explored in more detail in the following section.

6.3.2 Building a web service client (web service interoperability)

Most public gene and protein databases have a web interface, which is convenient for manual querying of the data. Additionally, database providers have started providing web service interfaces to their databases to support machine-machine interaction, i.e. enabling access to their database by other applications and software components.

Although one of the main ideas behind web services is to support interoperable machine-to-machine interaction, in practice there are still many incompatibilities, even at the level of the two most basic standards, WSDL and SOAP [3]. This becomes an issue in accessing public databases. These are typically hosted by a third party, which implies that there is no direct control over the interfaces to the data. Different databases may have used different web service platforms to implement the web service interfaces. In theory this should not matter but in practice it does. The promise of interoperability is that it should not matter which web service platform was used to implement the interface of a specific service. When building a client of the service a different web service platform can be chosen. This would indeed be the case if both adhere to the same standards and additionally the standards are sufficiently specific. The web service standards, and the platforms supporting these, are however not yet sufficiently mature so that this is the case. To illustrate the types of issues one might encounter, a specific interoperability problem is documented below.

6.3.2.1 Interoperability example

EBI provides many useful web services [2], one of which is WSDbFetch. This can be used to retrieve entries from various up-to-date biological databases, such as UniProt and Ensemble. EBI also provides a very simple client that can be used query the service. This works fine, as illustrated in Section 6.3.1. However, the functionality of the client is (intentionally) very

limited. It is provided for illustratory purposes. The idea is that developers will develop their own client software for using the web service, tailored to their needs.

The EBI web services, and sample clients that are provided alongside it, have been developed using the Apache Axis web service toolkit. As an experiment, we have attempted building a client using a different web service toolkit. We chose to use the JAX-WS API that is part of Sun's Java EE 5 SDK.

The first step in developing a web service client using the Java EE 5 SDK is to use the wsimport command to automatically generate the Java interface files from the WSDL file that describes the service. This failed, with the following messages:

```
warning: src-resolve: Cannot resolve the name 'soapenc:Array' to a(n)
     'type definition' component.
error: undefined simple or complex type 'soapenc:Array'
error: undefined attribute 'soapenc:arrayType'
```

After a little bit of investigation, we found a way to resolve this issue. This involved downloading the WSDL file and modifying it locally. At the places where the "http://schemas.xmlsoap.org/soap/encoding/" namespace was imported, we added a "schemaLocation" attribute to explicitly specify the location of the namespace (which happens to be the same as its name).

Running wsimport again now gave the following error:

```
error: rpc/encoded wsdls are not supported in JAXWS 2.0.
```

This was a more fundamental problem that is not that easy to resolve. It is a known interoperability issue. The Basic Profile is a set of recommendations developed by the WS-I (Web Services Interoperability organisation) on how to use web services specifications to maximize interoperability. It offers clarifications and additional constraints on the use of the basic web services standards in the areas where they are not sufficiently clear or specific.

The WS-I basic profile does not allow use of rpc/encoded:

- "RPC/encoded is specifically disallowed in the Basic Profile because it is thought to duplicate the function of XML namespaces, as well as provide too limited a set of data types." [3]
  - See Basic Profile 1.0 [4] and Basic Profile 1.1 [5] (R2706)
- "The choice of encoded (still the default for some toolkits, including even Apache Axis 1.3, but now beginning to change [...]) was the usual combination with RPC, and is now heavily discouraged by the Basic Profile." [3]

JAX-WS supports the Basic Profile 1.1 standard, and therefore does not support WSDL files that use rpc/encoded. However, the latter has been the default for many toolkits, including Apache Axis which was used by EBI to implement their web services interfaces.

In response to an inquiry about this with EBI's support team, we received the following reply (dd. 4 Sept 2006): "Currently, EBI web services are rpc/encoded, and JAX-WS does not support that combination, we are aware of the WS-I recommendation and will be offering a doc/lit implementation very soon".

6.3.2.2 Discussion

This is one example of an interoperability issue, but there are more [3]. Although these will hopefully be eventually resolved, interoperability will remain an issue in the foreseeable future. Firstly, it takes time to fix and extend the existing standards. Subsequently, it takes time before these standards are fully supported by the various web service development platforms. Finally, it will take time before the various public databases support the latest

standards. Especially when changes to (implementations of) a standard introduce incompatibilities, there may be reluctance to upgrade, in particular when there are many clients, developed by others, out in the field already.

6.3.3 OGSA-DAI wrappers for DICOM access

To gain hands-on experience with OGSA-DAI, and to become aware of possible practical difficulties associated with it, we started building wrappers in OGSA-DAI for accessing DICOM image databases. Firstly, it should be noted that the MEDICUS project, which was introduced in Section 6.2.1.5, does offer this functionality. The main aim of MEDICUS is to share DICOM images across a Virtual Organisations spanning multiple hospitals. For this purpose, images are stored and further replicated on the grid. There is an OGSA-DAI interface to query images that are stored on the grid. However, images that are stored on an existing PACS and not replicated to the grid, cannot be queried. So there is still a need for an additional, more generic, OGSA-DAI data access interface for querying DICOM servers.

The DICOM protocol is very extensive. Its full description spans over three thousand pages [16]. So to effectively interact with a DICOM server you need a library that implements the standard. Fortunately, there is an open source implementation of DICOM. It is provided by the dcm4che project, by the toolkit they provide with the same name which is implemented in Java. We have been using Version 2.0.7 of the dcm4che toolkit.

There are two different modes of operation that need to be supported when wrapping DICOM data sources, which correspond to the two functionalities provided by the DICOM Query/Retrieve service class. Firstly, it needs to be possible to query a server to find out information about the images that it stores. Secondly, it needs to be possible to retrieve specific images

6.3.3.1 Implementing the query functionality

Given an implementation of the DICOM standard, the OGSA-DAI functionality required to query a DICOM server can be implemented relatively straightforwardly. The DICOM standard includes an enumeration of the various fields that can be included in a query. The number of fields that is supported is limited, and therefore it is natural to provide a specific XML element for each one. This way, the XML schema for the DICOM query activity can describe to a large extent how queries need to be specified, which helps to make sure that queries are well formed. The query activity can return the query results as a Java object, as returned by the dcm4che toolkit. Subsequent activities can then take this Java object as input, and process it as needed. For example, one activity could transfer it into an XML representation. This is similar to the way that OGSA-DAI supports SQL queries, where the SQL query statement returns the results as a rowset object (see for example Figure 20). Figure 25 illustrates how querying of DICOM servers can be achieved using OGSA-DAI.

6.3.3.2 Implementing the retrieve functionality

Implementing the functionality in OGSA-DAI to subsequently retrieve DICOM images is more complicated than the implementation of the query functionality. The reason is that the DICOM protocol does not support retrieval of images directly. Images can be retrieved, but they will be transferred separately from the query results. To retrieve images, a DICOM application entity needs to be specified in the query message to which the images can be sent. This means that retrieval of images requires the running of a DICOM application entity at the server where the OGSA-DAI wrapper resides. It is then the responsibility of the application entity to forward any images it receives to the activity that initiated the transfer. Subsequently, the images can be processed by further activities, for instance, to deliver them to file. This implementation is outlined in Figure 26.

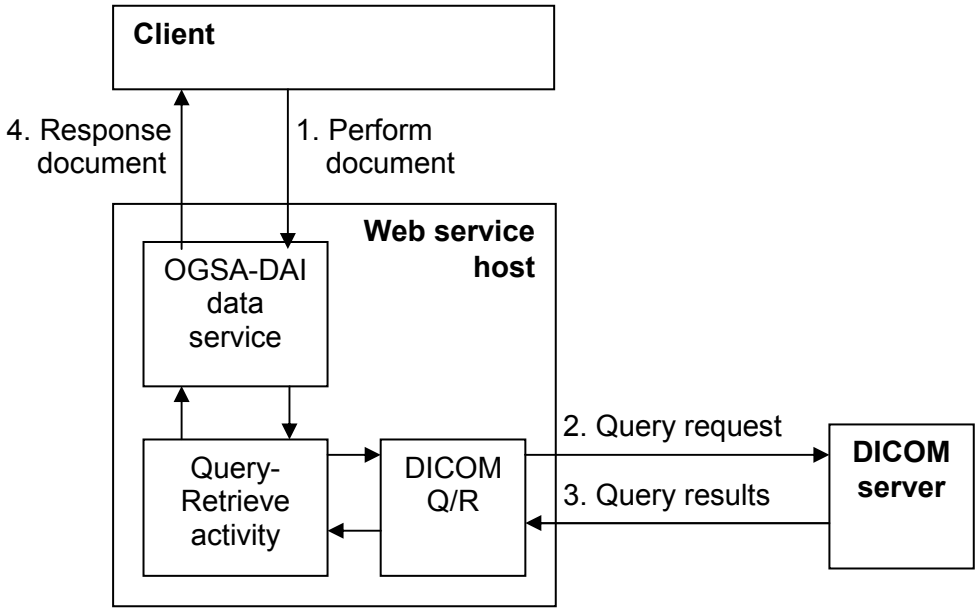


Figure 25 DICOM image query using OGSA-DAI.

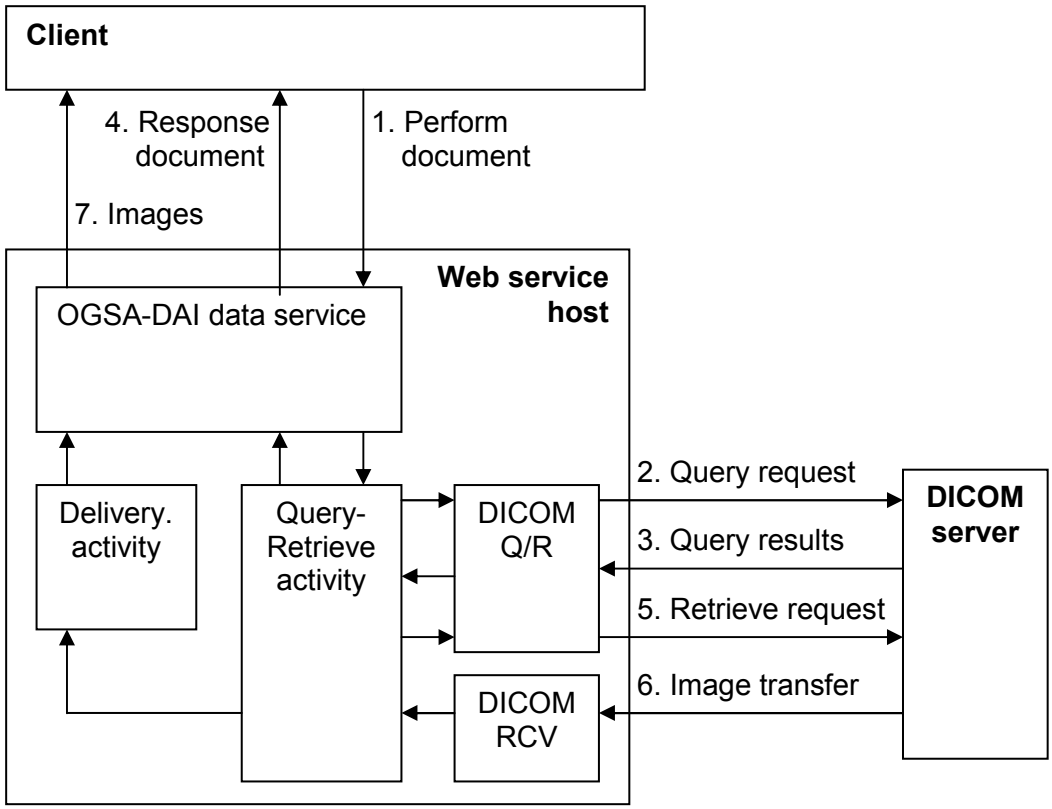


Figure 26 DICOM image query and retrieval using OGSA-DAI.

6.3.3.3 Conclusion

We have not yet finished the implementation of the DICOM wrappers in OGSA-DAI, which means that we cannot yet give an indication of the performance penalty associated with it, or report any other issues that are encountered when using these wrappers.

However, our experiences so far are promising. As has been explained, the DICOM image retrieval differs from the standard query and response protocol used to access generic database systems. Nevertheless, the design of the OGSA-DAI platform is sufficiently flexible so that it can be used to implement the DICOM wrappers in a way that fits in the overall OGSA-DAI platform. Doing so has the advantage that the activities that are available within the OGSA-DAI platform can be applied to the querying of DICOM servers as well. This for instance means that it should be relatively straightforward to deliver the DICOM images using the various mechanisms provided by OGSA-DAI.

7 Conclusion

In this document we have taken the initial requirements as defined by WP2, and further extended these with respect to distributed data access. Our initial focus is on the querying of data, i.e. access to existing data, as various other work package rely on this functionality. However, we realise that clinical users would also like to see additional support for data entry. As mentioned in Section 2.2.1, user wishes also include a system for management and reporting of SAEs/SUSARs, a generic clinical database management system, and a management system for storing bio-medical results. Initially, this will be addressed to a limited extent. This is discussed in more detail in Section 7.1. Subsequently, Section 7.2 describes what is to be done next in WP5 with respect to data access.

7.1 Services for the creation and the management of clinico genomic trials on cancer

One goal of WP5 is the integration of currently running trials into the ACGT environment by providing wrappers for the existing databases. However, as identified in Section 2.2.1, users would like a homogenization of all the procedures required for future clinico genomic trials from design to analysis phase. Therefore, as a long term goal, it is highly desirable for WP5 to provide services to allow easily setting up ACGT-compliant clinico genomic trials allowing generating automatically data management services for the conduction of the trial in Virtual Organizations within the ACGT environment.

The need for a methodology and tools to set up standardized clinico genomic trials is obvious since currently data definition for most clinical trials is done from scratch resulting in incomplete or non existing annotation and standardization of the data collected in clinical trials. Therefore, often even experts are only rarely able to make good use of data collected on studies they were not directly involved with. This fact will also make the semantic integration of legacy databases into ACGT error prone and time intensive despite the best mapping tools provided.

To overcome these drawbacks it is necessary that the tools for setting up a clinical trial focus on a standardized data definition during the design phase of the trial allowing to collect all kind of clinical and genomic data with comprehensive metadata during running the trial.

Therefore, we identified the need for WP5 to concentrate in the long term on developing services for the creation and management of clinical trials with the following requirements:

- The services have to allow domain experts (clinicians) to define requirements for a clinical trial and the definition of all kind of research and administrative data that has to be collected during the trial in a standardized, user-friendly way. During this process it has to be possible to design standardized user interfaces, called Case Report Forms (CRFs), which can be used to collect the defined data during the trial.
- In many trials similar or equal data is collected. So it is highly desirable to store CRFs or parts of them, once specified, in a repository in the ACGT environment for their reuse in later trials.
- It has to be possible to deploy data management services automatically from the defined requirements for the conduction of the trial in the ACGT environment. A trial database with comprehensive metadata has to be deployed during that process. During the conduction of the trial it has to be possible that authorized persons can enter data from remote sites using the defined electronic CRFs into the database. That will allow clinicians at the participating centers to enter clinical data directly during examining the patient as well as entering biomedical analysis results directly from the laboratories.

- The data collected in the generated trial databases can be automatically integrated into the mediator architecture of ACGT.
- All legal, ethical and security requirements have to be fulfilled by the services.

A promising solution to fulfil the goals of the described services is to base them on the ACGT master ontology since this ontology is used to semantically integrate all kind of data into the ACGT environment. The trial data definition as well as the metadata for the trial databases has to be based on the ACGT master ontology. The ACGT master ontology seems well suited for this purpose since it is a formal ontology and therefore, comprises logical descriptions that serve to computationally define terms as well as human-readable definitions. That means the ontology can help a human to define the data that has to be collected during the clinical trial. This definition can then be used as machine processable metadata for the trial database. The approach of basing the development of software applications on ontologies is in the literature known as ontology driven software development. Research on this field is still in its infancy. For a further discussion of advantages and drawbacks of this approach in general and for the described services see D7.1.

It is crucial for the long term success of ACGT that WP5 will focus on the described services since they will allow simplifying standardized set up of clinical trials saving time and costs for the participating organizations. CRFs are standardized based on the underlying ontologies and the trial database has extensive human and computer understandable metadata. In this way we can achieve semantic interoperability between the data collected during the clinical trials from the beginning and the data can be automatically integrated into the mediator architecture of ACGT.

Since the main focus of WP5 lies on the integration of heterogeneous data sources, it is not yet clear if the resources of the project will allow developing the services with all described functionalities within the four years of project duration. The described approach of ontology driven software development seems to be a very promising approach to solve the problem of semantic interoperability in the future and so it is highly relevant for the project to investigate this approach further. This will be done in WP7 and therefore it is intended to develop the described services in strong collaboration with WP7. For further description see D7.1.

Through this approach the vision of machine understandable metadata and automatic, intelligent processing of the data may become true in the future.

7.2 Data access

In the short term, work in WP5 on data access will proceed in various areas. Our initial focus will be on wrapping trial-specific databases as opposed to public databases. This choice was made in Requirements Engineering workshop held during the Consortium meeting in Crete. This focus should help to ensure that effort is not diluted and visible results can be obtained soon. It also avoids some of the practical web service interoperability problems that currently exist, as described in Section 6.3.2.

We will continue our implementation of OGSA-DAI wrappers for access to DICOM data. The aim is to share this functionality with the other technical partners in ACGT project soon. This can provide first hands-on experience with OGSA-DAI to query and access clinical data.

Furthermore, we hope to soon obtain the database schema that are used by the Jules Bordet institute to store the CRF data in the Oracle Clinical database. We have already received extensive information but still require some addional information. Once we have communicated what we need and subsequently received it, we will process it in order to present it in a similar format as is used in Appendix 4 and 5.

We will also have to take into account the consolidated requirement documents that are finalised at the same time as this deliverable. These can be used as a starting point for further discussion, to further clarify the interactions of WP5 with the other work packages. Of special importance is the interaction with WP7 and WP11. Interaction with WP7 is needed to specify the interface between the mediator and the wrapper services. D7.1 [23] does not give clear requirements on the wrapper interfaces. Further discussion is therefore necessary. For this, it may be useful to first agree the types of queries that need to be supported. Subsequently, the semantics and syntax can be agreed upon. Interaction with WP11 is necessary to decide on how trial-specific data will be made available so that all legal and ethical obligations are met. To facilitate this discussion, we outlined in Section 5.1 the main approaches that can be taken. D11.1 [24] proposes the development of pseudonymisation tools and services, and we should work out how these can be used by WP5. Furthermore, we obviously need to take into account the initial ACGT ethical and legal requirements to be provided by WP10 in month 12.

Looking further ahead, one of the research questions for WP5 with respect to data access is what the wrapper query language should be. Ideally the query language should be the same, irrespective of the underlying data source. To achieve this, one could define a query language that is the lowest common denominator over all query languages used by the various data sources. This has the obvious drawback that there is a loss of expressiveness. It will be impossible to express certain complex queries in this common query language, even though the underlying database may support it. Alternatively, one can define a wrapper query language that is more powerful than that supported by some of the underlying data sources. This can in theory be implemented[2] but typically comes with significant implementation difficulties and performance penalties that it is not feasible in practice. Therefore, it may be inevitable that wrapper services for different types of data source will sometimes use (slightly) different query mechanisms. Further research is needed here, taking into account the differences between the various data sources that need to be accessed, which have been listed in Section 4.3.

A relatively minor but interesting question is how the wrappers should support bio-molecular specific query operations, such as the approximate sequence matching performed by BLAST. Although these could be syntactically expressed as string matching, the results are different from normal string-matching. Therefore, the wrapper should be able to firstly, express that the matching is not standard string matching, and secondly, express the details of the approximate matching scheme and the parameters that can be set to configure it.

Another area of research within WP5, related to the choice of query language, is how the wrapper services should describe the schema and content of the underlying database such that queries can be constructed appropriately. As mentioned in Section 4.3, this may require more than describing the database schema alone. The content of meta-tables in relational databases also needs to be included, either as part of the local ontology, or with additional mappings to the schema.

In addressing these questions, it is also important to take existing standards into account as much as possible. For example, OGSA-DAI web services currently do not interpret or convert queries in anyway, and the standard offers no support for this. Queries are submitted to OGSA-DAI services as plain text, and passed on directly to the underlying databases. Furthermore, there is no support in OGSA-DAI to describe the schema of the underlying database sources. So it needs to be investigated if the OGSA-DAI can be suitably extended, or whether there are other standards that fill the gap.

[2] For example, by letting the wrapper services fully mirror the data that they are wrapping so that they are not constrained by possible limitations of the query language of the wrapped data source.

Finally, one of the main challenges in WP5 will be a practical one, and involves dealing with the many third party systems and middleware software. Most of the data sources that need to be accessed are not under control of WP5. Each will offer their own interface for accessing the data, or worse, some may even lack suitable interfaces entirely, which would then need to be addressed. These interfaces may also change at any time, for example as a result of updates to the database. For the public database we have no control over these changes, but even for trial-specific databases our control will be limited at best. Also, the implementation of wrapper services will rely on many different third-party software components, from many different sources. WP5 heavily relies on grid and web service software. Grid middleware in itself is already fairly complex, consisting of many software tools with occasional subtle interactions. There are also many different web service standards. Even the most basic ones, including SOAP, WSDL, and WS-I, are still evolving. This also affects the platforms that support web service technologies. Web services generated using different web servce platforms, or even different versions of the same platform, may not be interoperable. This can be problematic, as we have no control over the platforms that are used by third parties that provide web services. Section 6.3.2 gave an example of such a problem occurring in practise. Furthermore, additional drivers and platforms will be needed to access specific types of databases, such as for example a toolkit that implements the DICOM protocol as was mentioned in Section 6.3.3. There may be various choices, each which varying levels of maturity. Documentation of toolkits may also be lacking, or partly incorrect. Few, if any, of these toolkits will be free of bugs. Some bugs may be fixeable, or can be worked around, but others will be showstoppers. This also means that when there is a choice of software components that might be suitable, the best one may not be apparent quickly. Sometimes only experience may tell. Although each of these issues is mundane, the number of them that we can expect to encounter in WP5 means that it is an issue to be reckoned with, which will take up a considerable amount of our time.

References

[1] G. Muller, “CAFCR: A Multi-view Method for Embedded Systems Architecting; Balancing Genericity and Specificity”, Ph.D. thesis, Technical University Delft, 2004

[2] Pillai S., Silventoinen V., Kallio K., Senger M., Sobhany S., Tate J., Velankar S., Golovin A., Henrick K., Rice P., Stoehr P., and Lopez R. “SOAP-based services provided by the European Bioinformatics Institute”, *Nucleic Acids Res*. 33(1):W25-W28, 2005

[3] Madeleine Wright, "A detailed investigation of interoperability for web services", MSc thesis, Rhodes University, December, 2005

[4] WS-I, Basic Profile Version 1.0, 16 April 2004, http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[5] WS-I, Basic Profile Version 1.1, 24 August 2004, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

[6] Andreas Pfitzmann and Marit Hansen, “Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management – A consolidated Proposal for Terminology”, Version v0.28, May 2006, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

[7] Lao H. Saal, Carl Troein, Johan Vallon-Christersson, Sofia Gruvberger, Åke Borg and Carsten Peterson, “BioArray Software Environment: A Platform for Comprehensive Management and Analysis of Microarray Data”, *Genome Biology*, 2002 3(8)

[8] BASE web site: http://base.thep.lu.se

[9] Said Mirza Pahlevi, Isao Kojima, "OGSA-WebDB: An OGSA-Based System for Bringing Web Databases into the Grid", *International Conference on Information Technology: Coding and Computing (ITCC'04)*, Volume 2, p. 105, 2004.

[10] OGSA-DAI web site: http://www.ogsadai.org.uk/

[11] M. Antonioletti, M.P. Atkinson, R. Baxter, A. Borley, N.P. Chue Hong, B. Collins, N. Hardman, A. Hume, A. Knox, M. Jackson, A. Krause, S. Laws, J. Magowan, N.W. Paton, D. Pearson, T. Sugden, P. Watson, and M. Westhead. “The Design and Implementation of Grid Database Services in OGSA-DAI”. *Concurrency and Computation: Practice and Experience*, Volume 17, Issue 2-4, Pages 357-376, February 2005.

[12] K. Karasavvas, M. Antonioletti, M.P. Atkinson, N.P. Chue Hong, T. Sugden, A.C. Hume, M. Jackson, A. Krause, C. Palansuriya. “Introduction to OGSA-DAI Services”. *Lecture Notes in Computer Science*, Volume 3458, Pages 1-12, May 2005.

[13] Simple Object Access Protocol (SOAP) 1.1, W3C Note, 8 May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[14] Web Services Description Language (WSDL) 1.1, W3C Note, 15 March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315/

[15] Web Services Architecture, W3C Working Draft, 14 November 2002, http://www.w3.org/TR/2002/WD-ws-arch-20021114/ - Web Services Architecture, W3C Working Draft, 14 November 2002

[16] National Electrical Manufacturers Association, “Digital Imaging and Communications in Medicine (DICOM)”, 2004, http://medical.nema.org/dicom/2004.html

[17] W. Dean Bidgood Jr., Steven C. Horii, Fred W. Prior, Donald E. van Syckle, “Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging”, *Journal of the American Medical Informatics Association*, Volume 4, Number 3, May/June 1997

[18] Martin Senger, Peter Rice, Tom Oinn, "Soaplab - a unified Sesame door to analysis tools", *Proceedings UK e-Science All Hands Meeting*, Simon J Cox (Editor), pp. 509-513, ISBN - 1-904425-11-9, September 2003

[19] P. Rice, I. Longden, A. Bleasby, "EMBOSS: The European Molecular Biology Open Software Suite", *Trends in Genetics* 16 (6) pp 276—277, 2000

[20] S.G. Erberich, M. Bhandekar, A. Chervenak, M.D. Nelson, C. Kesselman. "DICOM grid interface service for clinical and research PACS: A globus toolkit web service for medical data grids". *Int Journal of Computer Assistant Radiology and Surgery*, 2006, 1:87-105; pp 100—102, Springer, Heidelberg

[21] S.G. Erberich, M. Dixit, A. Chervenak, V. Chen, M.D. Nelson, C. Kesselmann. "Grid Based Medical Image Workflow and Archiving for Research and Enterprise PACS Applications". *PACS and Imaging Informatics, SPIE Medical Imaging*, pp. 6145—32, 2006

[22] Manolis Tsiknakis (Editor), Deliverable 2.1: "User requirements and specification of the ACGT internal clinical trial", 13 September 2006

[23] Luis Martín (Editor), Deliverable 7.1: "Consolidated Requirements on ontological approaches for integration of multi-level biomedical information", final version to appear in January 2007

[24] Brecht Clearhout and Stefan Castille (Editors), Deliverable 11.1: "Consolidation of security requirements of ACGT and initial security architecture", final version to appear January 2007

Appendix 1 – Abbreviations and acronyms

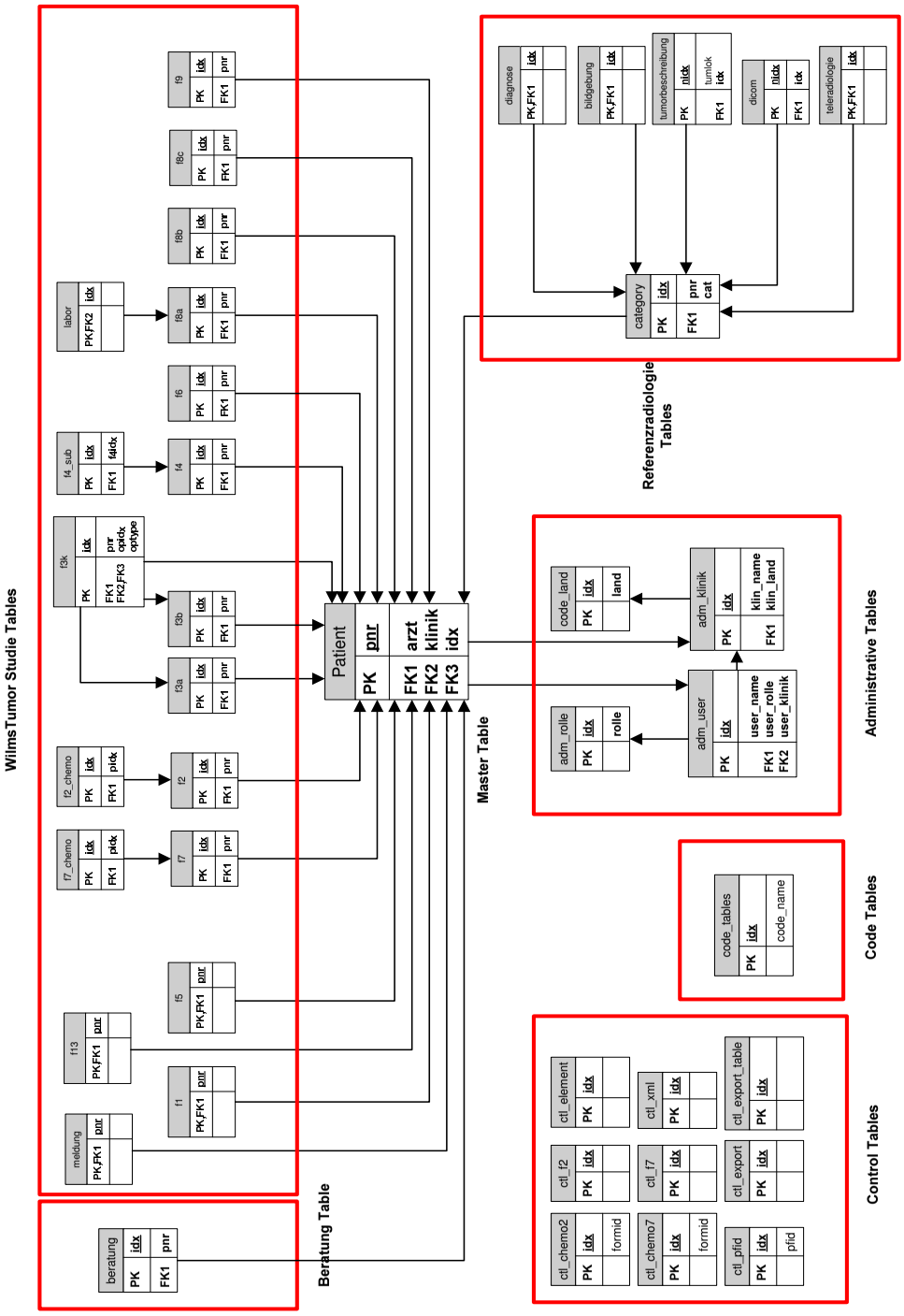
This glossary lists various acronyms that are used throughout the deliverable. It does not include all acronyms that are used in this deliverable. Acronyms that are only introduced and used in a particular section, and not referred to subsequently, are typically excluded.

ACGT	Advancing Clinico Genomic Trials (on Cancer). The 6th Framework project under which the work reported here is carried out.
BASE	Bio-Array Software Environment. A database for managing microarray data.
CAFCR	An architectural reasoning method, consisting of five views: Customer, Application, Functional, Conceptual and Realization
CDA	Clinical Document Architecture. An HL7 standard for marking up documents for exchange.
CDMS	Clinical Data Management System.
CRF	Clinical Report Form.
CT	Computed Tomography. A medical imaging method.
D2.1	An ACGT deliverable: “User requirements and specification of the ACGT internal clinical trials”
D5.1	This ACGT deliverable: “Consolidated requirements and specifications for data access”
D7.1	An ACGT deliverable: “Consolidated requirements on ontological approaches for integration of multi-level biomedical information”
D11.1	An ACGT deliverable: “Consolidation of security requirements of ACGT and initial security architecture”
DGIS	DICOM Grid Interface Service. A part of MEDICUS.
DICOM	Digital Imaging and Communications in Medicine. A standard for exchanging medical image data.
EBI	European Bio informatics institute.
ECG	Electrocardiogram.
EMBOSS	European Molecular Biology Open Software Suite. A software analysis package for molecular biology.
GIF	Graphics Interchange Format. A generic image standard.
HL7	Health Level Seven. A standards institute for the clinical and

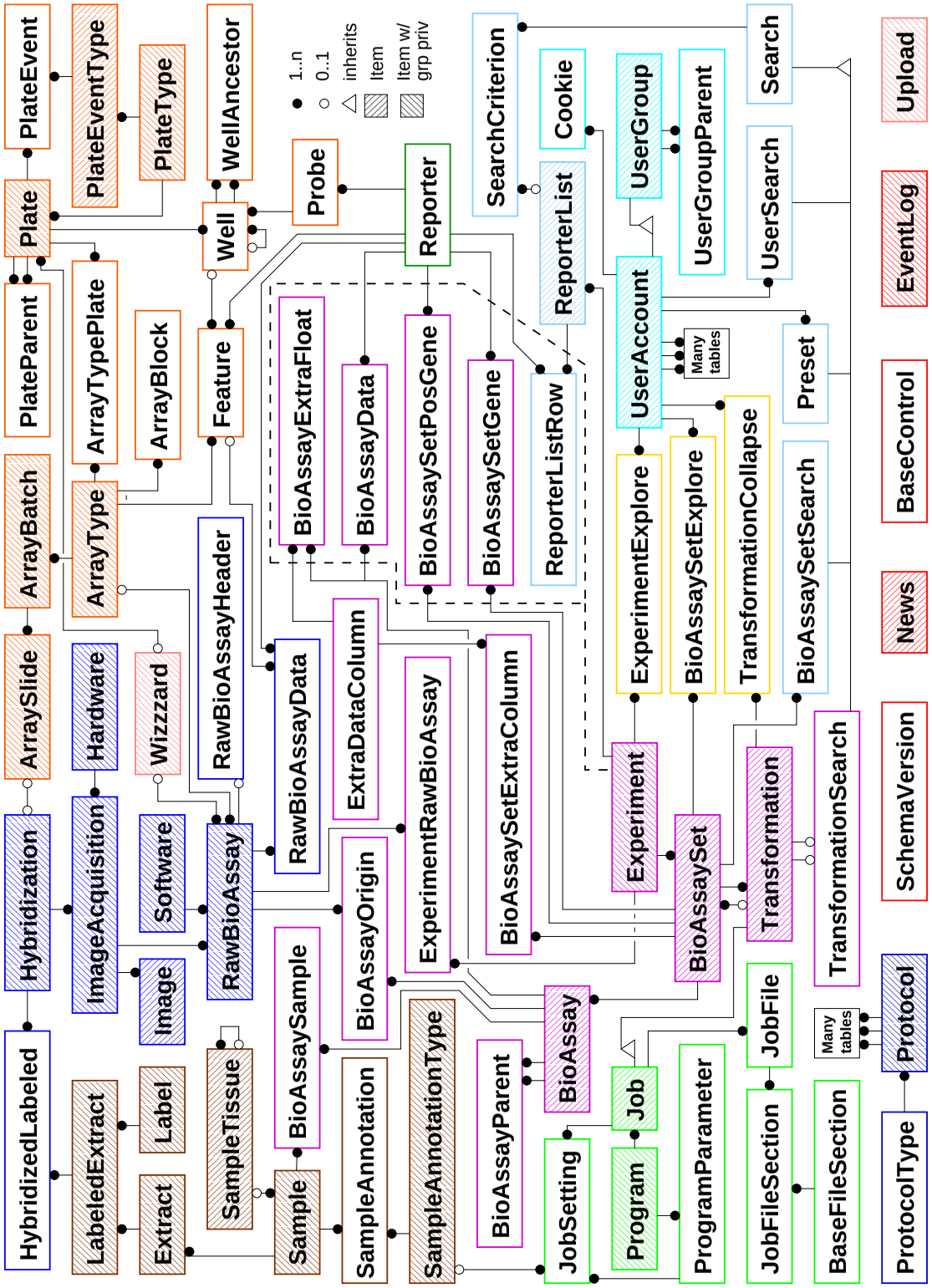
	administrative data domain.
HTTP	Hypertext Transfer Protocol.
JPEG	A generic image standard proposed by the Joint Photographic Experts Group.
MEDICUS	Medical Imaging and Computing for Unified Information Sharing
MRI	Magnetic Resonance Imaging. A medical imaging method.
OGSA-DAI	OGSA standard for Data Access and Integration. A middleware product that supports the exposure of data resources, such as relational or XML databases, on to grids.
PACS	Picture Archiving and Communication System
PET/CT	Positron Emission Tomography. A medical imaging method.
RPC	Remote Procedure Call.
SAE	Suspected Adverse Event.
SC2	Scenario described in D2.1: “Identification of Nephroblastoma antigens”
SC5	Scenario described in D2.1: “In-Silico modelling of tumor response to therapy”
SEREX	Serological expression of cDNA expression libraries. A biomolecular analysis method.
SOAP	An XML-based, extensible message envelope format.
Soaplab	Toolkit for providing web service wrappers around command-line applications.
SUSAR	Suspected Unexpected Severe Adverse Reaction.
SQL	Standard Query Language. A query language for relational databases.
US	Ultrasound. A medical imaging method.
WADO	Web Access to DICOM Objects. A standard that is part of the DICOM specification.
WP2	ACGT work package: “User Needs Analysis & Specifications”
WP3	ACGT work package: “Architecture and Standards”
WP5	ACGT work package: “Distributed Data Access and Applications”
WP6	ACGT work package: “Data Mining and Knowledge Discovery Tools”

WP7	ACGT work package: “Ontologies and Semantic Mediation Tools”
WP9	ACGT work package: “The Integrated ACGT Environment”
WP10	ACGT work package: “Ethics, Legal and QA issues”
WP11	ACGT work package: “Trust and Security”
WP12	ACGT work package: “Clinical Trials”
WSDL	Web Service Description Language. An XML format that allows service interfaces to be described.
WS-I	Web Services Interoperability. Industry organisation promoting web services interoperability.
WSRF	Web Services Resource Framework.
XML	Extensible Mark-up Language. The syntax that is used by all web service standards.

Appendix 2 – Main relations in the SIOP Clinical database



Appendix 3 – Main relations in the BASE (v 1.2) database



Appendix 4 – Schema of the SIOP Clinical database

For each of the tables in the database the name and data types of every column is given in the tables that follow, starting at the next page.

Table 7 adm_abfrage columns.

Name	Data type
idx	Long Integer
name	Memo
abfdat	DateTime
rolle	Long Integer
sqlstr	Memo

Table 8 adm_klinik columns.

Name	Data type
idx	Long Integer
klin_name	Memo
klin_str	Memo
klin_plz1	Long Integer
klin_plz2	Long Integer
klin_ort	Memo
klin_art	Long Integer
klin_land	Long Integer
klin_abt	Long Integer
klin_abt_sonst	Memo
klin_tel	Memo
klin_fax	Memo
klin_pbox	Memo

Table 9 adm_rolle columns.

Name	Data type
idx	Long Integer
rolle	Memo
patadmin	Long Integer
patsiop9	Boolean
patsiop93	Boolean
patsiop2001	Boolean
patberat	Boolean
patrefrad	Boolean
patmolek	Boolean
patbest	Boolean
patpath	Boolean
patop	Boolean
patchemo	Boolean
recht_stud	Boolean
recht_berat	Boolean
recht_export	Boolean
recht_admin	Boolean
recht_wilms	Boolean
recht_ref	Boolean
recht_cpat	Boolean
recht_apat	Boolean
recht_delpat	Boolean
recht_freez	Boolean
recht_siopnr	Boolean
recht_sts	Boolean
recht_code	Boolean
recht_bugfix	Boolean
recht_meld	Boolean
meldung	Boolean
f1	Boolean
f2	Boolean
f3a	Boolean
f3b	Boolean
f3k	Boolean
f4	Boolean
f5	Boolean
f6	Boolean
f7	Boolean
f8a	Boolean
f8b	Boolean
f8c	Boolean
f9	Boolean
f13	Boolean
patup	Boolean
frag	Boolean
bild	Boolean
diag	Boolean
tube	Boolean
meta	Boolean
tele	Boolean
dicom	Boolean

Table 10 adm_user columns.

Name	Data type
idx	Long Integer
user_dat	DateTime
user_einv	Long Integer
user_einvdat	DateTime
user_lock	Long Integer
user_cxlogin	Memo
user_name	Memo
user_vorname	Memo
user_rolle	Long Integer
user_sex	Long Integer
user_pas	Memo
user_klinik	Long Integer
user_abteilung	Long Integer
user_abteilung_sonst	Memo
user_beruf	Long Integer
user_beruf_sonst	Memo
user_title	Memo
user_tel1	Memo
user_tel2	Memo
user_fax	Memo
user_email	Memo
user_komm_mit	Long Integer
user_protokol	Long Integer

Table 11 begruendung columns.

Name	Data type

idx	Long Integer
pnr	Long Integer
beg_user	Long Integer
pfid	Memo
beg_dat	DateTime
beg	Long Integer
beg_rueck	Long Integer
pro_beg	Memo

Table 12 beratung columns.

Name	Data type
idx	Long Integer
sts	Long Integer
anf_erf	Long Integer
dat_anf	DateTime
anf_tag	Long Integer
anf_uhr	DateTime
anf_fei	Long Integer
beant_zeit	Long Integer
anz_ruech_klin	Long Integer
anz_rueck_ref	Long Integer
dat_beant	DateTime
beant_uhr	DateTime
beant_dauer	Long Integer
beant_erf	Long Integer
name_berat	Long Integer
name_berat_sonst	Memo
pos_berat	Long Integer
pos_berat_sonst	Memo
klinik	Long Integer
beruf	Long Integer
beruf_sonst	Memo
berat_pat	Long Integer
berat_pat_dk	Long Integer
berat_pat_de	Long Integer
berat_pat_op	Long Integer
berat_pat_rand	Long Integer
berat_pat_chemo	Long Integer
berat_pat_kom	Long Integer
berat_pat_rez	Long Integer
berat_pat_man	Long Integer
berat_pat_stra	Long Integer
berat_pat_gen	Long Integer
berat_pat_sonst	Long Integer
berat_pat_sonst_com	Memo
pnr	Long Integer
pat_gem	Long Integer
berat_stu_dok	Long Integer
berat_stu_web	Long Integer
berat_stu_rde	Long Integer
berat_stu_feh	Long Integer
berat_stu_stud	Long Integer
berat_stu_rand	Long Integer
berat_zus_pro	Long Integer
berat_zus_lit	Long Integer
berat_anf_teil	Long Integer
berat_anf_wilm	Long Integer
berat_anf_pro	Long Integer
berat_anf_log	Long Integer
berat_anf_sonst	Long Integer
berat_anf_sonst_com	Memo
fragestellung	Memo
bew_in_klin	Long Integer
bew_dring	Long Integer
anf_dient	Long Integer
anf_dient_sonst_com	Memo
anf_weiter	Long Integer
stud_disk	Long Integer
klinikintern	Long Integer
studienpro	Long Integer
bean_lit	Long Integer
bean_dat	Long Integer
bean_anf	Long Integer
weit_refstrah	Long Integer
weit_refrad	Long Integer
weit_refchir	Long Integer
weit_refpath	Long Integer
weit_gess	Long Integer
int_studie	Long Integer
weit_sonst	Long Integer
weit_sonst_com	Memo
beraten	Memo
kontakt	Long Integer
kon_ja	Long Integer
kon_sonst_com	Memo
wer_berat	Long Integer
wer_berat_sonst_com	Memo
berat_erf	Long Integer

Table 13 bildgebung columns.

Name	Data type
idx	Long Integer
sts	Long Integer
us_hard	Long Integer
us_cdrom	Long Integer
us_online	Long Integer
ct_hard	Long Integer
ct_cdrom	Long Integer
ct_online	Long Integer
mrt_hard	Long Integer
mrt_cdrom	Long Integer
mrt_online	Long Integer
us_dicom	Long Integer
ct_dicom	Long Integer
mrt_dicom	Long Integer
us_format_sonst	Long Integer
us_format_sonst_com	Memo
ct_format_sonst	Long Integer
ct_format_sonst_com	Memo

mrt_format_sonst	Long Integer
mrt_format_sonst_com	Memo
us_scan	Long Integer
ct_scan	Long Integer
mrt_scan	Long Integer
mrtab	Long Integer
t1_ax	Long Integer
t1_cor	Long Integer
t1_sag	Long Integer
t1km_ax	Long Integer
t1km_cor	Long Integer
t1km_sag	Long Integer
t2_ax	Long Integer
t2_cor	Long Integer
t2_sag	Long Integer
fatsat_com	Memo
mrt_sonst_com	Memo
mrt_ausw	Long Integer
mrt_ir	Memo
mrt_quali	Long Integer
ctab	Long Integer
ctab_nativ	Long Integer
ctab_km	Long Integer
ctab_spiral	Long Integer
ctab_dicke	Double
ctab_kv	Double
ctab_mas	Double
ctab_quali	Long Integer
us	Long Integer
us_tum	Long Integer
us_nier	Long Integer
us_kont	Long Integer
us_vc	Long Integer
us_lk	Long Integer
us_le	Long Integer
us_sonst	Long Integer
us_sonst_com	Memo

us_format	Long Integer
us_quali	Long Integer
aug	Long Integer
mibg	Long Integer
cavo	Long Integer
angio	Long Integer
sonst	Long Integer
sonst_com	Memo
roetx	Long Integer
roe_paap	Long Integer
roe_seit	Long Integer
roe_quali	Long Integer
cttx	Long Integer
cttx_nativ	Long Integer
cttx_km	Long Integer
cttx_dicke	Double
cttx_kv	Double
cttx_quali	Long Integer
ausreich	Long Integer
weitere	Long Integer
bef_thx	Long Integer
bef_us	Long Integer
bef_ctab	Long Integer
bef_mrt	Long Integer
bef_cttx	Long Integer
klin_an	Long Integer
klin_nachfr	Long Integer
erg_mrt_t1	Long Integer
erg_mrt_t1km	Long Integer
erg_mrt_t2	Long Integer
erg_mrt_sonst	Long Integer
erg_mrt_sonst_com	Memo
erg_ctab_native	Long Integer
erg_ctab_km	Long Integer
erg_us_ab	Long Integer
erg_us_nier	Long Integer
erg_us_kont	Long Integer

erg_us_vc	Long Integer
erg_us_lk	Long Integer
erg_us_le	Long Integer
erg_us_sonst	Long Integer
erg_us_sonst_com	Memo
erg_roe_paap	Long Integer
erg_roe_seit	Long Integer
erg_cttx_nativ	Long Integer
erg_cttx_km	Long Integer
erg_mibg	Long Integer
erg_sonst	Long Integer
erg_sonst_com	Memo

Table 14 category columns.

Name	**Data type**
idx	Long Integer
pnr	Long Integer
sts	Long Integer
bildvon	Long Integer
us_bilddat	DateTime
mrt_bilddat	DateTime
ct_bilddat	DateTime
refdat	DateTime
alterj	Long Integer
alterm	Long Integer
zeitbisref	Long Integer
cat	Long Integer
fr_ther	Long Integer
fr_tumvol	Long Integer
fr_rez	Long Integer
fr_prog	Long Integer
fr_sonst	Long Integer
fr_sonst_com	Memo
rezauf	Long Integer
rezsonst_com	Memo
fr_bes_com	Memo

Table 15 chemo2 columns.

Name	Data type
idx	Long Integer
pidx	Long Integer
cday	Long Integer
cweek	Long Integer
dat	DateTime
act	Double
vcr	Double
dox	Double
gewicht	Double
tvr	Double
tvl	Double
dosred	Long Integer
grund_red	Long Integer
dosred_com	Memo

Table 16 chemo7 columns.

Name	Data type
idx	Long Integer
pidx	Long Integer
cday	Long Integer
cweek	Long Integer
dat	DateTime
act	Double
vcr	Double
dox	Double
carbo	Double
vp16	Double
cpm	Double
cyclo	Double
gewicht	Double
dosred	Long Integer
grund_red	Long Integer
dosred_com	Memo

Table 17 code_abtklin columns.

Name	Data type
idx	Long Integer
name	Memo

Table 18 code_alter columns.

Name	Data type
idx	Long Integer
alter	Memo

Table 19 code_ana_subtyp columns.

Name	Data type
idx	Long Integer
ana_subtyp	Memo

Table 20 code_anf_dient columns.

Name	Data type
idx	Long Integer
anf_dient	Memo

Table 21 code_ankuend columns.

Name	Data type
idx	Long Integer
ankuend	Memo

Table 22 code_anlass columns.

Name	Data type
idx	Long Integer
anlass	Memo

Table 23 code_anzop columns.

Name	Data type
idx	Long Integer
anzop	Memo

Table 24 code_anztum columns.

Name	Data type
idx	Long Integer
anztum	Memo

Table 25 code_artklin columns.

Name	Data type
idx	Long Integer
artklin	Memo

Table 26 code_artmess columns.

Name	Data type
idx	Long Integer
artmess	Memo

Table 27 code_asse columns.

Name	Data type
idx	Long Integer
name	Memo

Table 28 code_aus columns.

Name	Data type
idx	Long Integer
aus	Memo

Table 29 code_ausdeh columns.

Name	Data type
idx	Long Integer
name	Memo

Table 30 code_austl columns.

Name	Data type
idx	Long Integer
austl	Memo

Table 31 code_ausw columns.

Name	Data type
idx	Long Integer
ausw	Memo

Table 32 code_az columns.

Name	Data type
idx	Long Integer
az	Memo

Table 33 code_bean_anf columns.

Name	Data type
idx	Long Integer
bean_anf	Memo

Table 34 code_beant_zeit columns.

Name	Data type
idx	Long Integer
beant_zeit	Memo

Table 35 code_bef columns.

Name	Data type
idx	Long Integer
bef	Memo

Table 36 code_bef_subj columns.

Name	Data type
idx	Long Integer
bef_subj	Memo

Table 37 code_befund columns.

Name	Data type
idx	Long Integer
befund	Memo

Table 38 code_befzeit columns.

Name	Data type
idx	Long Integer
befzeit	Memo

Table 39 code_begruendung columns.

Name	Data type
idx	Long Integer
begruendung	Memo

Table 40 code_beh_klin columns.

Name	Data type
idx	Long Integer
beh_klin	Memo

Table 41 code_berat_erf columns.

Name	Data type
idx	Long Integer
berat_erf	Memo

Table 42 code_berat_pat columns.

Name	Data type
idx	Long Integer
berat_pat	Memo

Table 43 code_beruf columns.

Name	Data type
idx	Long Integer
beruf	Memo

Table 44 code_beurt columns.

Name	Data type
idx	Long Integer
beurt	Memo

Table 45 code_bild_sp columns.

Name	Data type
idx	Long Integer
bild_sp	Memo

Table 46 code_bildvon columns.

Name	Data type
idx	Long Integer
bildvon	Memo

Table 47 code_biop_spez columns.

Name	Data type
idx	Long Integer
biop_spez	Memo

Table 48 code_cat columns.

Name	Data type
idx	Long Integer
cat	Memo

Table 49 code_category columns.

Name	Data type
idx	Long Integer
category	Memo

Table 50 code_cav_aus columns.

Name	Data type
idx	Long Integer
cav_aus	Memo

Table 51 code_color columns.

Name	Data type
idx	Long Integer
color	Memo

Table 52 code_ctc columns.

Name	Data type
idx	Long Integer
ctc	Memo

Table 53 code_dgsich columns.

Name	Data type
idx	Long Integer
dgsich	Memo

Table 54 code_diag columns.

Name	Data type
idx	Long Integer
name	Memo

Table 55 code_diag_klin columns.

Name	Data type
idx	Long Integer
diag_klin	Memo

Table 56 code_diag_refrad columns.

Name	Data type
idx	Long Integer
diag_refrad	Memo

Table 57 code_diaggleich columns.

Name	Data type
idx	Long Integer
diaggleich	Memo

Table 58 code_dok columns.

Name	Data type
idx	Long Integer
dok	Memo

Table 59 code_due_nm columns.

Name	Data type
idx	Long Integer
due_nm	Memo

Table 60 code_due_per columns.

Name	Data type
idx	Long Integer
due_per	Memo

Table 61 code_due_schluess columns.

Name	Data type
idx	Long Integer
due_schluess	Memo

Table 62 code_due_von columns.

Name	Data type
idx	Long Integer
due_von	Memo

Table 63 code_due_weg columns.

Name	Data type
idx	Long Integer
due_weg	Memo

Table 64 code_echo columns.

Name	Data type
idx	Long Integer
echo	Memo

Table 65 code_einv columns.

Name	Data type
idx	Long Integer
einv	Memo

Table 66 code_erf columns.

Name	Data type
idx	Long Integer
erf	Memo

Table 67 code_f8acat columns.

Name	Data type
idx	Long Integer
name	Memo

Table 68 code_geeig columns.

Name	Data type
idx	Long Integer
geeig	Memo

Table 69 code_geraet columns.

Name	Data type
idx	Long Integer
geraet	Memo

Table 70 code_gr_malign columns.

Name	Data type
idx	Long Integer
gr_malign	Memo

Table 71 code_grund columns.

Name	Data type
idx	Long Integer
grund	Memo

Table 72 code_grundl_bef columns.

Name	Data type
idx	Long Integer
grundl_bef	Memo

Table 73 code_hepatox columns.

Name	Data type
idx	Long Integer
hepatox	Memo

Table 74 code_histo1 columns.

Name	Data type
idx	Long Integer
histo1	Memo

Table 75 code_histo2 columns.

Name	Data type
idx	Long Integer
histo2	Memo

Table 76 code_histosubtyp columns.

Name	Data type
idx	Long Integer
name	Memo

Table 77 code_histotyp columns.

Name	Data type
idx	Long Integer
histotyp	Memo

Table 78 code_hom columns.

Name	Data type
idx	Long Integer
hom	Memo

Table 79 code_indikation columns.

Name	Data type
idx	Long Integer
indikation	Memo

Table 80 code_kap_rup_art columns.

Name	Data type
idx	Long Integer
kap_rup_art	Memo

Table 81 code_kap_rup_wann columns.

Name	Data type
idx	Long Integer
kap_rup_wann	Memo

Table 82 code_klin_an columns.

Name	Data type
idx	Long Integer
klin_an	Memo

Table 83 code_km_aufn columns.

Name	Data type
idx	Long Integer
km_aufn	Memo

Table 84 code_km_aufnst columns.

Name	Data type
idx	Long Integer
km_aufnst	Memo

Table 85 code_kontakt columns.

Name	Data type
idx	Long Integer
kontakt	Memo

Table 86 code_land columns.

Name	Data type
idx	Long Integer
land_name	Memo

Table 87 code_lk_ent columns.

Name	Data type
idx	Long Integer
lk_ent	Memo

Table 88 code_lk_tum columns.

Name	Data type
idx	Long Integer
lk_tum	Memo

Table 89 code_lok_intra columns.

Name	Data type
idx	Long Integer
lok_intra	Memo

Table 90 code_lok_lu columns.

Name	Data type
idx	Long Integer
lok_lu	Memo

Table 91 code_lok_nier columns.

Name	Data type
idx	Long Integer
lok_nier	Memo

Table 92 code_lok_soli columns.

Name	Data type
idx	Long Integer
lok_soli	Memo

Table 93 code_lokal columns.

Name	Data type
idx	Long Integer
lokal	Memo

Table 94 code_lokal_prob columns.

Name	Data type
idx	Long Integer
name	Memo

Table 95 code_maengel columns.

Name	Data type
idx	Long Integer
maengel	Memo

Table 96 code_massn_subj columns.

Name	Data type
idx	Long Integer
massn_subj	Memo

Table 97 code_meh_art columns.

Name	Data type
idx	Long Integer
meh_art	Memo

Table 98 code_meh_ent columns.

Name	Data type
idx	Long Integer
meh_ent	Memo

Table 99 code_mekt_erf columns.

Name	Data type
idx	Long Integer
mekt_erf	Memo

Table 100 code_mekt_gen columns.

Name	Data type
idx	Long Integer
mekt_gen	Memo

Table 101 code_met_chron columns.

Name	Data type
idx	Long Integer
met_chron	Memo

Table 102 code_metbild columns.

Name	Data type
idx	Long Integer
metbild	Memo

Table 103 code_nephrec columns.

Name	Data type
idx	Long Integer
name	Memo

Table 104 code_nier_erh columns.

Name	Data type
idx	Long Integer
nier_erh	Memo

Table 105 code_op_grund columns.

Name	Data type
idx	Long Integer
op_grund	Memo

Table 106 code_op_zugang columns.

Name	Data type
idx	Long Integer
op_zugang	Memo

Table 107 code_op3a columns.

Name	Data type
idx	Long Integer
op3a	Memo

Table 108 code_op3b columns.

Name	Data type
idx	Long Integer
op3b	Memo

Table 109 code_opart columns.

Name	Data type
idx	Long Integer
opart	Memo

Table 110 code_organ columns.

Name	Data type
idx	Long Integer
name	Memo

Table 111 code_orgaus columns.

Name	Data type
idx	Long Integer
orgaus	Memo

Table 112 code_part_neph columns.

Name	Data type
idx	Long Integer
part_neph	Memo

Table 113 code_pat_gem columns.

Name	Data type
idx	Long Integer
pat_gem	Memo

Table 114 code_patadmin columns.

Name	Data type
idx	Long Integer
patadmin	Memo

Table 115 code_patart columns.

Name	Data type
idx	Long Integer
patart	Memo

Table 116 code_pathtyp columns.

Name	Data type
idx	Long Integer
name	Memo

Table 117 code_per_aus columns.

Name	Data type
idx	Long Integer
per_aus	Memo

Table 118 code_pos_berat columns.

Name	Data type
idx	Long Integer
pos_berat	Memo

Table 119 code_praeop_beh columns.

Name	Data type
idx	Long Integer
praeop_beh	Memo

Table 120 code_prim_beh columns.

Name	Data type
idx	Long Integer
prim_beh	Memo

Table 121 code_prim_feld columns.

Name	Data type
idx	Long Integer
prim_feld	Memo

Table 122 code_prog columns.

Name	Data type
idx	Long Integer
name	Memo

Table 123 code_proz columns.

Name	Data type
idx	Long Integer
proz	Memo

Table 124 code_quali columns.

Name	Data type
idx	Long Integer
quali	Memo

Table 125 code_radikal columns.

Name	Data type
idx	Long Integer
radikal	Memo

Table 126 code_radikal2 columns.

Name	Data type
idx	Long Integer
radikal2	Memo

Table 127 code_radikal3 columns.

Name	Data type
idx	Long Integer
radikal3	Memo

Table 128 code_rand columns.

Name	Data type
idx	Long Integer
rand	Memo

Table 129 code_rand_erg columns.

Name	Data type
idx	Long Integer
rand_erg	Memo

Table 130 code_rand_erg_2 columns.

Name	Data type
idx	Long Integer
name	Memo

Table 131 code_rand_erg_3 columns.

Name	Data type
idx	Long Integer
name	Memo

Table 132 code_random columns.

Name	Data type
idx	Long Integer
random	Memo

Table 133 code_refdiagwie columns.

Name	Data type
idx	Long Integer
refdiagwie	Memo

Table 134 code_resgrund columns.

Name	Data type
idx	Long Integer
resgrund	Memo

Table 135 code_resorg columns.

Name	Data type
idx	Long Integer
resorg	Memo

Table 136 code_response columns.

Name	Data type
idx	Long Integer
response	Memo

Table 137 code_restyp columns.

Name	Data type
idx	Long Integer
name	Memo

Table 138 code_rezauf columns.

Name	Data type
idx	Long Integer
rezauf	Memo

Table 139 code_rueck columns.

Name	Data type
idx	Long Integer
name	Memo

Table 140 code_ruecks columns.

Name	Data type
idx	Long Integer
ruecks	Memo

Table 141 code_sat_aus columns.

Name	Data type
idx	Long Integer
sat_aus	Memo

Table 142 code_seite columns.

Name	Data type
idx	Long Integer
seite	Memo

Table 143 code_seite_mb columns.

Name	Data type
idx	Long Integer
seite_mb	Memo

Table 144 code_sex columns.

Name	Data type
idx	Long Integer
sex	Memo

Table 145 code_softw columns.

Name	Data type
idx	Long Integer
softw	Memo

Table 146 code_soli columns.

Name	Data type
idx	Long Integer
soli	Memo

Table 147 code_sort columns.

Name	Data type
idx	Long Integer
sortname	Memo

Table 148 code_stadium columns.

Name	Data type
idx	Long Integer
stadium	Memo

Table 149 code_status columns.

Name	Data type
idx	Long Integer
status	Memo

Table 150 code_status_ta columns.

Name	Data type
idx	Long Integer
status_ta	Memo

Table 151 code_status2 columns.

Name	Data type
idx	Long Integer
status2	Memo

Table 152 code_sts columns.

Name	Data type
idx	Long Integer
sts	Memo
stslock	Long Integer
stscolor	Long Integer

Table 153 code_stud_disk columns.

Name	Data type
idx	Long Integer
name	Memo

Table 154 code_studie columns.

Name	Data type
idx	Long Integer
studie	Memo

Table 155 code_tag columns.

Name	Data type
idx	Long Integer
tag	Memo

Table 156 code_ther_dox columns.

Name	Data type
idx	Long Integer
ther_dox	Memo

Table 157 code_thermit columns.

Name	Data type
idx	Long Integer
thermit	Memo

Table 158 code_thora_ct columns.

Name	Data type
idx	Long Integer
thora_ct	Memo

Table 159 code_thr_lokal columns.

Name	Data type
idx	Long Integer
thr_lokal	Memo

Table 160 code_thr_morph columns.

Name	Data type
idx	Long Integer
thr_morph	Memo

Table 161 code_throm columns.

Name	Data type
idx	Long Integer
throm	Memo

Table 162 code_todurs columns.

Name	Data type
idx	Long Integer
todurs	Memo

Table 163 code_toxgrad columns.

Name	Data type
idx	Long Integer
toxgrad	Memo

Table 164 code_tum_mat columns.

Name	Data type
idx	Long Integer
tum_mat	Memo

Table 165 code_tumlok columns.

Name	Data type
idx	Long Integer
tumlok	Memo

Table 166 code_tumstruk columns.

Name	Data type
idx	Long Integer
tumstruk	Memo

Table 167 code_tumvol columns.

Name	Data type
idx	Long Integer
tumvol	Memo

Table 168 code_uebel columns.

Name	Data type
idx	Long Integer
uebel	Memo

Table 169 code_urs columns.

Name	Data type
idx	Long Integer
urs	Memo

Table 170 code_us_format columns.

Name	Data type
idx	Long Integer
us_format	Memo

Table 171 code_v_aus columns.

Name	Data type
idx	Long Integer
v_aus	Memo

Table 172 code_va_neph columns.

Name	Data type
idx	Long Integer
va_neph	Memo

Table 173 code_verlauf columns.

Name	Data type
idx	Long Integer
verlauf	Memo

Table 174 code_verlauf_2 columns.

Name	Data type
idx	Long Integer
verlauf_2	Memo

Table 175 code_volumen columns.

Name	Data type
idx	Long Integer
volumen	Memo

Table 176 code_vorl columns.

Name	Data type
idx	Long Integer
vorl	Memo

Table 177 code_yn columns.

Name	Data type
idx	Long Integer
yn	Memo

Table 178 code_yne columns.

Name	Data type
idx	Long Integer
yne	Memo

Table 179 code_yni columns.

Name	Data type
idx	Long Integer
yni	Memo

Table 180 code_ynn columns.

Name	Data type
idx	Long Integer
ynn	Memo

Table 181 code_zweit_malig columns.

Name	Data type
idx	Long Integer
zweit_malig	Memo

Table 182 ctl_chemo2 columns.

Name	Data type
idx	Long Integer
formid	Text
did	Long Integer
wid	Long Integer
act	Boolean
vcr	Boolean
dox	Boolean
gewicht	Boolean
tv	Boolean

Table 183 ctl_chemo7 columns.

Name	Data type
idx	Long Integer
formid	Text
did	Long Integer
wid	Long Integer
act	Boolean
vcr	Boolean
dox	Boolean
carbo	Boolean
vp16	Boolean
cpm	Boolean
cyclo	Boolean
gewicht	Boolean

Table 184 ctl_convert_value columns.

Name	Data type
idx	Long Integer
valuename	Text
valueid	Long Integer
org_value	Text
exp_value	Text

Table 185 ctl_export columns.

Name	Data type
idx	Long Integer
table_nr	Long Integer
att_name	Text

Table 186 ctl_export_n columns.

Name	Data type
idx	Long Integer
table_nr	Long Integer
att_name	Text

Table 187 ctl_export_table columns.

Name	Data type
idx	Long Integer
table_name	Text

att_count	Long Integer
table_sql	Memo

Table 188 ctl_f2 columns.

Name	Data type
idx	Long Integer
formid	Text
week	Long Integer
vf1	Boolean
tf1	Memo

Table 189 ctl_f7 columns.

Name	Data type
idx	Long Integer
formid	Text
week	Long Integer
vf1	Boolean
vf2	Boolean
vf3	Boolean
vf4	Boolean
vf5	Boolean
vf6	Boolean
vf7	Boolean
vf8	Boolean
vf9	Boolean

Table 190 ctl_fields columns.

Name	Data type
idx	Long Integer
fidx	Long Integer
fieldname	Memo
fieldobj	Memo

Table 191 ctl_forms columns.

Name	Data type
idx	Long Integer
name	Memo

Table 192 ctl_pfid columns.

Name	Data type
idx	Long Integer
pfid	Memo
meldung	Boolean
f1	Boolean
f2	Long Integer
f3a	Boolean
f3b	Boolean
f3k	Boolean
f4	Boolean
f5	Boolean
f6	Boolean
f7	Long Integer
f8a	Boolean
f8b	Boolean
f8c	Boolean
f9	Boolean
f13	Boolean
protokol	Long Integer

Table 193 ctl_tables columns.

Name	Data type
idx	Long Integer
table_name	Memo
form_name	Memo
tablepk	Memo
reg_nr	Long Integer

Table 194 ctl_treeview_main columns.

Name	Data type
idx	Long Integer
nodeidx	Long Integer
att_name	Memo
obj_name	Memo
node_name	Memo

Table 195 ctl_treeview_main_master columns.

Name	Data type
idx	Long Integer
node_name	Memo

Table 196 diagnose columns.

Name	Data type
idx	Long Integer
sts	Long Integer
diag_klin	Long Integer
diag_klin_com	Memo
refdiagwie	Long Integer
diag_refrad	Long Integer
diag_refrad_com	Memo
va_neph	Long Integer
diaggleich	Long Integer
lok_nier	Long Integer
lok_lk	Long Integer
lok_vc	Long Integer
lok_lu	Long Integer
lok_le	Long Integer
lok_kno	Long Integer
lok_hirn	Long Integer
lok_sonst	Long Integer
lok_sonst_com	Memo
comment_ext	Memo
comment_int	Memo

ther_durchf	Long Integer
primop	Long Integer
patart	Long Integer

Table 197 dicom columns.

Name	Data type
idx	Long Integer
sts	Long Integer
catidx	Long Integer
type	Long Integer
name	Memo
dir	Memo

Table 198 diffdiag_klin columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
diag_klin	Long Integer
diag_klin_com	Memo

Table 199 diffdiag_refrad columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
diag_refrad	Long Integer
diag_refrad_com	Memo

Table 200 Einfgefehler columns.

Name	Data type
idx	Long Integer
table_nr	Long Integer
att_name	Text
att_pk	Long Integer
att_type	Long Integer
att_del	Long Integer
att_end	Long Integer
pk	Text
vvalue	Memo
valueid	Long Integer
source	Memo
condition	Memo
groupby	Long Integer

Table 201 f1 columns.

Name	Data type
pnr	Long Integer
sts	Long Integer
studie_teil	Long Integer
studie	Long Integer
studie_com	Memo
w_ther	Long Integer
w_ther_com	Memo
v_andklin	Long Integer
v_andklin_com	Memo
beh_klin	Long Integer
gr_malign	Long Integer
gr_malign_com	Memo
n_sym	Long Integer
n_sym_com	Memo
vortum	Long Integer
vortum_com	Memo
syndr	Long Integer
anir	Long Integer
wagr	Long Integer
urofehl	Long Integer
drash	Long Integer
emg	Long Integer
hemihyp	Long Integer
perl	Long Integer
famwilms	Long Integer
gerinn	Long Integer
hypert	Long Integer
rr_sys	Long Integer
rr_dias	Long Integer
and_syn	Long Integer
and_syn_com	Memo
fambel	Long Integer
fam_elt	Long Integer
fam_elt_com	Memo
fam_ges	Long Integer
fam_ges_com	Memo
fam_sonst	Long Integer
fam_sonst_com	Memo
anzges	Long Integer
mehrling	Long Integer
meh_art	Long Integer
meh_ent	Long Integer
gebjmut	Long Integer
gebjvat	Long Integer
az	Long Integer
diagdat	DateTime
therdat	DateTime
thermit	Long Integer
thermit_com	Memo
lokal	Long Integer
metast	Long Integer
lunge	Long Integer
lu_nachw	Long Integer
mediast	Long Integer
leber	Long Integer
exablk	Long Integer
abdomen	Long Integer
knochen	Long Integer
weicht	Long Integer
gehirn	Long Integer
met_and	Long Integer

met_and_com	Memo
roe	Long Integer
thor_ct	Long Integer
ab_ct	Long Integer
us	Long Integer
mrt	Long Integer
sonst	Long Integer
sonst_com	Memo
roe_anz	Long Integer
ct_anz	Long Integer
dur_lu_met	Long Integer
kat_urin	Long Integer
ref_rad	Long Integer
us_r	Long Integer
ct_r	Long Integer
mrt_r	Long Integer
us_l	Long Integer
ct_l	Long Integer
mrt_l	Long Integer
us_a_r	Double
us_b_r	Double
us_c_r	Double
us_v_r	Double
ct_a_r	Double
ct_b_r	Double
ct_c_r	Double
ct_v_r	Double
mrt_a_r	Double
mrt_b_r	Double
mrt_c_r	Double
mrt_v_r	Double
bild_sp_r	Long Integer
anztum_r	Long Integer
tumstruk_r	Long Integer
biop_r	Long Integer
biop_r_spez	Long Integer
gauche_r	Long Integer

biopdat_r	DateTime
us_a_l	Double
us_b_l	Double
us_c_l	Double
us_v_l	Double
ct_a_l	Double
ct_b_l	Double
ct_c_l	Double
ct_v_l	Double
mrt_a_l	Double
mrt_b_l	Double
mrt_c_l	Double
mrt_v_l	Double
bild_sp_l	Long Integer
anztum_l	Long Integer
tumstruk_l	Long Integer
biop_l	Long Integer
biop_l_spez	Long Integer
gauche_l	Long Integer
biopdat_l	DateTime
protpat	Long Integer
alter	Long Integer
prim_op	Long Integer
op_grund	Long Integer
and_grund	Long Integer
and_grund_com	Memo
vorbeh	Long Integer
beh	Long Integer
beh_com	Memo
bilat	Long Integer
andtum	Long Integer
andtum_com	Memo
fup	Long Integer
fup_com	Memo
patlebt	Long Integer
comment	Memo

Table 202 f13 columns.

Name	Data type
pnr	Long Integer
sts	Long Integer
diag	Long Integer
rec_vcr	Long Integer
rec_act	Long Integer
rec_sonst	Long Integer
rec_sonst_com	Memo
dur_preop	Long Integer
dat_preop	DateTime
prog_left	Long Integer
prog_right	Long Integer
bet_surg	Long Integer
bet_vcr	Long Integer
bet_act	Long Integer
bet_doxo	Long Integer
post_surg	Long Integer
post_av2	Long Integer
post_avd	Long Integer
post_cdcv	Long Integer
post_sonst	Long Integer
post_sonst_com	Memo
radio_pat	Long Integer
dat_treat	Memo
comment	Memo

Table 203 f2 columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
formid	Long Integer
gewicht	Double
groesse	Long Integer
oberfl	Double

tox	Long Integer
vod	Long Integer
respdat_r	DateTime
us_a_r	Double
us_b_r	Double
us_c_r	Double
us_v_r	Double
ct_a_r	Double
ct_b_r	Double
ct_c_r	Double
ct_v_r	Double
mrt_a_r	Double
mrt_b_r	Double
mrt_c_r	Double
mrt_v_r	Double
tumstruk_r	Long Integer
respdat_l	DateTime
us_a_l	Double
us_b_l	Double
us_c_l	Double
us_v_l	Double
ct_a_l	Double
ct_b_l	Double
ct_c_l	Double
ct_v_l	Double
mrt_a_l	Double
mrt_b_l	Double
mrt_c_l	Double
mrt_v_l	Double
tumstruk_l	Long Integer
lunge_roe	Long Integer
lunge_ct	Long Integer
mediast	Long Integer
leber	Long Integer
abdomen	Long Integer
knochen	Long Integer
gehirn	Long Integer
andmet	Long Integer
andmet_com	Memo
diag_met	Long Integer
diag_met_anz	Long Integer
prae_met	Long Integer
prae_met_anz	Long Integer
status2	Long Integer
thora_ct	Long Integer
thora_ct_com	Memo
comment	Memo

Table 204 f3a columns.

Name	**Data type**
idx	Long Integer
pnr	Long Integer
sts	Long Integer
opdat	DateTime
operateur	Long Integer
chirurg	Long Integer
ber_op	Long Integer
klin_op	Long Integer
anzop_kh	Long Integer
anzop_chir	Long Integer
praeop_beh	Long Integer
lokal	Long Integer
lokal_prob	Long Integer
cav_aus_diag	Long Integer
cav_aus_ct	Long Integer
gefaess_chir	Long Integer
op_zugang	Long Integer
op_zugang_com	Memo
opart	Long Integer
nephrec	Long Integer
radikal	Long Integer
part_neph	Long Integer
stadop	Long Integer
lk_makr	Long Integer
per_aus	Long Integer
per_biop	Long Integer
per_biop_com	Memo
per_radikal	Long Integer
nv_aus	Long Integer
nv_biop	Long Integer
nv_biop_com	Memo
nv_radikal	Long Integer
nv_throm	Long Integer
vc_aus	Long Integer
vc_biop	Long Integer
vc_biop_com	Memo
vc_radikal	Long Integer
vc_throm	Long Integer
vc_byp	Long Integer
vc_pro	Long Integer
kap_aus	Long Integer
kap_rup	Long Integer
kap_rup_wann	Long Integer
kap_rup_art	Long Integer
lk_aus_reg	Long Integer
lk_aus_ex	Long Integer
lk_ent_reg	Long Integer
lk_ent_ex	Long Integer
lk_ent_com	Memo
lk_rup	Long Integer
lk_ent_anz	Long Integer
un_int	Long Integer
un_int_com	Memo
nn_aus	Long Integer
nn_ent	Long Integer
fett_aus	Long Integer
fett_ent	Long Integer
ur_aus	Long Integer
ur_ent	Long Integer
le_aus	Long Integer

le_ent	Long Integer
pso_aus	Long Integer
pso_ent	Long Integer
zwe_aus	Long Integer
zwe_ent	Long Integer
mi_aus	Long Integer
mi_ent	Long Integer
pa_aus	Long Integer
pa_ent	Long Integer
co_aus	Long Integer
co_ent	Long Integer
kont_aus	Long Integer
kont_ent	Long Integer
and_aus	Long Integer
and_ent	Long Integer
and_com	Memo
comment	Memo

Table 205 f3b columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
metopdat	DateTime
operateur	Long Integer
chirurg	Long Integer
ber_op	Long Integer
klin_op	Long Integer
indikation	Long Integer
praeopchem	Long Integer
praeoprad	Long Integer
praeopmetop	Long Integer
met_lunge	Long Integer
met_knochen	Long Integer
met_zns	Long Integer
met_leber	Long Integer

met_weich	Long Integer
met_sonst	Long Integer
met_com	Memo
met_chron	Long Integer
seite	Long Integer
mekt_erf	Long Integer
verwachs	Long Integer
met_ent	Long Integer
met_ent_kpl	Long Integer
met_ent_ink	Long Integer
mekt_gen	Long Integer
w_keine	Long Integer
w_chemo	Long Integer
w_radio	Long Integer
w_reop	Long Integer
w_stamm	Long Integer
comment	Memo

Table 206 f3k columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
opidx	Long Integer
optype	Long Integer
op_kompl_dat	DateTime
kompl	Long Integer
tumrupmi	Long Integer
tumrupma	Long Integer
blut	Long Integer
druckabf	Long Integer
herz	Long Integer
gefaess	Long Integer
darm	Long Integer
darmver	Long Integer
milz	Long Integer

leber	Long Integer
sonst	Long Integer
sonst_com	Memo
postop_kompl	Long Integer
post_blut	Long Integer
vc_obs	Long Integer
adhaes	Long Integer
invag	Long Integer
wundinf	Long Integer
wunddeh	Long Integer
narbenh	Long Integer
zwerchfh	Long Integer
post_sonst	Long Integer
post_sonst_com	Memo
resand	Long Integer
resorg	Long Integer
resorg_com	Memo
resgrund	Long Integer
kompmed	Long Integer
kompchir	Long Integer
reop	Long Integer
reopdat	DateTime
tod	Long Integer
tod_com	Memo
spaetf	Long Integer
spaetf_com	Memo
verzoeg	Long Integer
verzanz	Long Integer
comment	Memo

Table 207 f4 columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
pdate	DateTime

lokal_prob	Long Integer

Table 208 f4_sub columns.

Name	Data type
idx	Long Integer
f4idx	Long Integer
rel_his	Long Integer
pathtyp	Long Integer
berdat	DateTime
eingnr	Memo
lok_path	Long Integer
klin_path	Long Integer
anz_sch	Long Integer
anz_blo	Long Integer
chir_inf	Long Integer
prim_beh	Long Integer
lokal	Long Integer
tum_mat	Long Integer
gew_praep	Long Integer
durchm	Double
praep_intakt	Long Integer
kaps_intakt	Long Integer
ob_tusch	Long Integer
multifok	Long Integer
multifok_com	Memo
rand_mak	Long Integer
rand_mik	Long Integer
rand_com	Memo
thromb_nv_mak	Long Integer
thromb_nv_mik	Long Integer
proz_mak	Long Integer
proz_mak_gen	Long Integer
proz_his	Long Integer
proz_his_gen	Long Integer
proz_blas	Long Integer
proz_epi	Long Integer
proz_stroma	Long Integer
reg_niere	Long Integer
reg_perih	Long Integer
reg_perir	Long Integer
reg_lymph	Long Integer
reg_res	Long Integer
vit_niere	Long Integer
vit_perih	Long Integer
vit_perir	Long Integer
vit_lymph	Long Integer
vit_res	Long Integer
inf_niere	Long Integer
inf_hilus	Long Integer
inf_venen	Long Integer
inf_cava	Long Integer
reste	Long Integer
restyp	Long Integer
ana_subtyp	Long Integer
histotyp	Long Integer
histo_sonst	Memo
lk_tum	Long Integer
lk_com	Memo
lk_ent_anz	Long Integer
vit_tum_anz	Long Integer
stadlok	Long Integer
stadlok_com	Memo
stud_ass	Long Integer
stud_vers	Long Integer
mirrbl	Long Integer
comment	Memo

Table 209 f6 columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
nameradio	Long Integer
klin_radio	Long Integer
best_organ	Long Integer
best_organ_sonst	Memo
best_beg	DateTime
best_end	DateTime
geraet	Long Integer
geraet_com	Memo
prim_feld	Long Integer
sat_aus	Long Integer
sat_aus_com	Memo
boost	Long Integer
boost_com	Memo
boost_sat_aus	Long Integer
ges_dos	Double
einz_dos	Double
anz_best	Long Integer
dauer	Long Integer
unterbr	Long Integer
unterbr_com	Memo
leb_dos	Double
kont_dos	Double
boost_dos	Double
b_anz_sitz	Long Integer
b_dauer	Long Integer
b_unterbr	Long Integer
b_unterbr_com	Memo
uebel	Long Integer
erbr	Long Integer
hepatox	Long Integer
hepatox_com	Memo
andtox_com	Memo
hb	Double
leuko	Long Integer
neutro	Long Integer
thrombo	Long Integer
prim_ant_l	Double

prim_ant_b	Double
prim_post_l	Double
prim_post_b	Double
prim_and_l	Double
prim_and_b	Double
boost_ant_l	Double
boost_ant_b	Double
boost_post_l	Double
boost_post_b	Double
boost_and_l	Double
boost_and_b	Double
comment	Memo

Table 210 f7 columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
formid	Long Integer
stadlok	Long Integer
histo1	Long Integer
histo2	Long Integer
volumen	Long Integer
patrand	Long Integer
random	Long Integer
random_com	Memo
sollther	Long Integer
ther_dox	Long Integer
ther_dox_com	Memo
gcsf	Long Integer
tox	Long Integer
vod	Long Integer
tox_verst	Long Integer
status_ta	Long Integer
postop_kompl	Memo
grund_indiv	Memo

gewicht	Double
groesse	Long Integer
oberfl	Double
stad1_spez	Long Integer
stad2_spez	Long Integer
stad3_spez	Long Integer
stad4_spez	Long Integer
stad5_spez	Long Integer
comment	Memo

Table 211 f8a columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
f8acat	Long Integer
catdose	Double
sts	Long Integer
dat	DateTime
arrhyth	Long Integer
arrhyth_ther	Long Integer
herz_klin	Long Integer
infdauer	Double
echo	Long Integer
sf_wert	Double
esws	Double
dias_path	Long Integer
digital	Long Integer
diuret	Long Integer
ckmb	Long Integer
blut_vers	Long Integer
puls	Long Integer
anaemie	Long Integer
fieber	Long Integer
syst	Long Integer
diast	Long Integer
comment	Memo

Table 212 f8b columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
n_comment	Memo
event_com	Memo
meddra_llt_code	Long Integer
toxgrad	Long Integer
beginn	DateTime
ende	DateTime
weiterbest	Long Integer
and_urs	Long Integer
ther_urs	Long Integer
verlauf	Long Integer
verlauf_2	Long Integer
comment	Memo

Table 213 f8c columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
begdat	DateTime
enddat	DateTime
gpt	Double
bili	Double
ascit	Long Integer
gewicht_zu	Double
hepmeg	Long Integer
heppain	Long Integer
lebgroe	Double
splmeg	Long Integer
splpain	Long Integer
milzgroe	Double
actvorvod	Long Integer

datact	DateTime
dosact	Double
gewicht_act	Double
pat_bestr	Long Integer
leb_bestr	Long Integer
ther_prae	Long Integer
ther_post	Long Integer
ther_zeit	Long Integer
ther_zeit_com	Memo
and_fakt	Long Integer
and_fakt_com	Memo
comment	Memo

Table 214 f9 columns.

Name	Data type
idx	Long Integer
pnr	Long Integer
sts	Long Integer
l_unt	DateTime
ther_end	Long Integer
status	Long Integer
rezmet	Long Integer
rem	Long Integer
rez	Long Integer
rezdat	DateTime
met	Long Integer
metdat	DateTime
met_lu	Long Integer
met_le	Long Integer
met_ab	Long Integer
met_zns	Long Integer
met_kno	Long Integer
met_kno_com	Memo
met_lk	Long Integer
met_lk_com	Memo
met_weich	Long Integer
met_weich_com	Memo
met_sonst	Long Integer
met_sonst_com	Memo
zweitrem	Long Integer
zweitremdat	DateTime
kont	Long Integer
symp	Long Integer
rout_unt	Long Integer
klin_unt	Long Integer
bild	Long Integer
us	Long Integer
ct	Long Integer
mrt	Long Integer
vor_rueck	DateTime
roe_thor	DateTime
sono_ab	DateTime
zweittum	Long Integer
zweit_com	Memo
zweit_diagdat	DateTime
zweit_malig	Long Integer
zweit_lok	Memo
zweit_bestgeb	Long Integer
spaet_kompl	Long Integer
spaet_herz	Long Integer
spaet_nier	Long Integer
spaet_skel	Long Integer
spaet_com	Memo
totdat	DateTime
autopsie	Long Integer
tod_com	Memo
todurs	Long Integer
comment	Memo

Table 215 labor columns.

Name	Data type
idx	Long Integer
sts	Long Integer
gewicht	Double
groesse	Long Integer
oberfl	Double
datdox	DateTime
doxein	Long Integer
dox	Double
infdauer	Double
doxkum	Double
blut_vor	Long Integer
blut_end	Long Integer
blut24	Long Integer
blut48	Long Integer
blut5	Long Integer
blut21	Long Integer
dox_vor	Double
dox_end	Double
tro_vor	Double
tro24	Double
tro48	Double
tro5	Double
bnp_vor	Double
bnp21	Double
ein_klinik	Long Integer

Table 216 md_hierarchy columns.

Name	Data type
pt_code	Long Integer
hlt_code	Long Integer
hlgt_code	Long Integer
soc_code	Long Integer
pt_name	Text
hlt_name	Text
hlgt_name	Text
soc_name	Text
soc_abbrev	Text

bull_field	Text
pt_soc_code	Long Integer
primary_soc_fg	Text

Table 217 meldung columns.

Name	Data type
pnr	Long Integer
sts	Long Integer
auf_nr	Memo
diagnose	Memo
stad	Memo
malign	Memo
dgdat	DateTime
lokal	Memo
dgsich	Long Integer
seite_mb	Long Integer
stud	Long Integer
studname	Long Integer
einv	Long Integer
efb	Long Integer
dat	DateTime

Table 218 metastasen columns.

Name	Data type
idx	Long Integer
sts	Long Integer
nematose	Long Integer
anzherde_re	Long Integer
herde_re_diff	Long Integer
anzherde_li	Long Integer
herde_li_diff	Long Integer
herd_re_a	Long Integer
herd_re_b	Long Integer
herd_re_c	Long Integer
herd_re_v	Long Integer
herd_li_a	Long Integer
herd_li_b	Long Integer
herd_li_c	Long Integer
herd_li_v	Long Integer
artmessung	Long Integer
thrombus	Long Integer
thr_lokal	Long Integer
thr_morph	Long Integer
cavacomp	Long Integer
lkmet	Long Integer
lkmet_hilre	Long Integer
lkmet_hilli	Long Integer
lkmet_intab	Long Integer
lkmet_extab	Long Integer
lkmet_sonst	Long Integer
lkmet_sonst_com	Memo
lumetbilder	Long Integer
va_lumet	Long Integer
lumet	Long Integer
lumet_roetxp	Long Integer
lumet_roetxp_sol	Long Integer
lumet_roetxp_anzli	Long Integer
lumet_roetxp_anzre	Long Integer
lumet_roetxs	Long Integer
lumet_roetxs_sol	Long Integer
lumet_cttx	Long Integer
lumet_cttx_sol	Long Integer
lumet_cttx_anzli	Long Integer
lumet_cttx_anzre	Long Integer
metdiaggleich	Long Integer
lemet	Long Integer
lemet_anz	Long Integer
andmetbilder	Long Integer
va_andmet	Long Integer
andmet	Long Integer
knomet	Long Integer
hirnmet	Long Integer
sonstmet	Long Integer
sonstmet_com	Memo
metas_response	Long Integer
lok_response	Long Integer
neu_metas	Long Integer
neu_lokrez	Long Integer

Table 219 patient columns.

Name	Data type
pnr	Long Integer
pfreez	Long Integer
pfid	Memo
siopnr	Long Integer
gpohpid	Memo
izcode	Memo
molid	Memo
molmainz	Long Integer
pat_ref	Long Integer
pat_berat	Long Integer
pat_stud	Long Integer
pat_best	Long Integer
pat_molek	Long Integer
pat_path	Long Integer
pat_op	Long Integer
pat_chemo	Long Integer
pat_ber	Long Integer
pat_wilms	Long Integer
pat_sts	Long Integer
name	Memo
vorname	Memo
sex	Long Integer
gebdat	DateTime
pat_str	Memo
pat_plz	Long Integer
pat_ort	Memo
geb_ort	Memo

land	Long Integer
diag_klinik	Long Integer
klinik	Long Integer
arzt	Long Integer

Table 220 rand columns.

Name	Data type
pnr	Long Integer
sts	Long Integer
name5b	Memo
alter	Long Integer
unilat	Long Integer
keine_met	Long Integer
praeop_chem	Long Integer
stad2_3	Long Integer
echokard	Long Integer
postop_ther	Long Integer
nachbeob	Long Integer
einv	Long Integer
opdat	DateTime
rand_erg	Long Integer
rand_dat	DateTime

Table 221 Switchboard Items columns.

Name	Data type
SwitchboardID	Long Integer
ItemNumber	Integer
ItemText	Text
Command	Integer
Argument	Text

Table 222 teleradiologie columns.

Name	Data type
idx	Long Integer
sts	Long Integer
ankuend	Long Integer
due_von	Long Integer
due_per	Long Integer
due_weg	Long Integer
due_weg_com	Memo
due_schluess	Long Integer
due_schluess_com	Memo
due_nm	Long Integer
due_nm_com	Memo
due_anz	Long Integer
due_lesbar	Long Integer
due_sonst	Long Integer
due_sonst_com	Memo
bilder_geeig	Long Integer
gr_format	Long Integer
gr_quali1	Long Integer
gr_quali2	Long Integer
gr_unueb	Long Integer
gr_beschr	Long Integer
gr_kgesamt	Long Integer
gr_aufloes	Long Integer
gr_klein	Long Integer
gr_sonst	Long Integer
gr_sonst_com	Memo
bef_subj	Long Integer
bearb	Long Integer
abfoto	Long Integer
umformat	Long Integer
bearb_sonst	Long Integer
bearb_sonst_com	Memo
dauervbef	Long Integer
softw	Long Integer
softw_sonst	Long Integer
softw_com	Memo
dauerbef	Long Integer
befzeit	Long Integer
bearb_arch	Long Integer
arch_kopie	Long Integer
arch_foto	Long Integer
arch_sonst	Long Integer
arch_sonst_com	Memo
dauerarch	Long Integer
anzfilme	Long Integer
ruecksend	Long Integer
telklin_anz	Long Integer
telklin_dau	Long Integer
telklin_adr	Long Integer
telklin_klinan	Long Integer
telklin_tech	Long Integer
telklin_nachf	Long Integer
telklin_disk	Long Integer
telklin_bef	Long Integer
telklin_sonst	Long Integer
telklin_sonst_com	Memo
telstudass_anz	Long Integer
telstudass_dau	Long Integer
telstudass_klinan	Long Integer
telstudass_tech	Long Integer
telstudass_nachf	Long Integer
telstudass_disk	Long Integer
telstudass_bef	Long Integer
telstudass_sonst	Long Integer
telstudass_sonst_com	Memo
telstudlt_anz	Long Integer
telstudlt_dau	Long Integer
telstudlt_klinan	Long Integer
telstudlt_tech	Long Integer
telstudlt_nachf	Long Integer
telstudlt_disk	Long Integer
telstudlt_bef	Long Integer
telstudlt_sonst	Long Integer
telstudlt_sonst_com	Memo
mailklin_anz	Long Integer

mailklin_dau	Long Integer
mailklin_adr	Long Integer
mailklin_klinan	Long Integer
mailklin_tech	Long Integer
mailklin_nachf	Long Integer
mailklin_disk	Long Integer
mailklin_bef	Long Integer
mailklin_sonst	Long Integer
mailklin_sonst_com	Memo
mailstudass_anz	Long Integer
mailstudass_dau	Long Integer
mailstudass_klinan	Long Integer
mailstudass_tech	Long Integer
mailstudass_nachf	Long Integer
mailstudass_disk	Long Integer
mailstudass_bef	Long Integer
mailstudass_sonst	Long Integer
mailstudass_sonst_com	Memo
mailstudlt_anz	Long Integer
mailstudlt_dau	Long Integer
mailstudlt_klinan	Long Integer
mailstudlt_tech	Long Integer
mailstudlt_nachf	Long Integer
mailstudlt_disk	Long Integer
mailstudlt_bef	Long Integer
mailstudlt_sonst	Long Integer
mailstudlt_sonst_com	Memo
grundl_bef	Long Integer
massn_subj	Long Integer
probleme	Memo

Table 223 tumourbeschreibung columns.

Name	Data type
idx	Long Integer
sts	Long Integer
catidx	Long Integer
tumlok	Long Integer
lok_soli	Long Integer
lok_intra	Long Integer
lok_cra	Long Integer
lok_cau	Long Integer
lok_zen	Long Integer
lok_ven	Long Integer
lok_dor	Long Integer
lok_dif	Long Integer
lok_sonst	Long Integer
lok_sonst_com	Memo
lok_anz	Long Integer
tumvol	Long Integer
mrt_a	Double
mrt_b	Double
mrt_c	Double
mrt_v	Double
ct_a	Double
ct_b	Double
ct_c	Double
ct_v	Double
us_a	Double
us_b	Double
us_c	Double
us_v	Double
morph_hom	Long Integer
morph_eizys	Long Integer
morph_typ	Long Integer
morph_extra	Long Integer
morph_ober	Long Integer
morph_gef	Long Integer
morph_verk	Long Integer
morph_cys	Long Integer
morph_sonst	Long Integer
morph_const_com	Memo
ausdeh	Long Integer
nekrosen	Long Integer
einblut	Long Integer
subkaps_fl	Long Integer
va_rup	Long Integer
fl_ab	Long Integer
inf_pso	Long Integer
inf_le	Long Integer
inf_zwerch	Long Integer
inf_sonst	Long Integer
inf_sonst_com	Memo
intrathor	Long Integer
sonst	Long Integer
sonst_com	Memo
us_echo_nie	Long Integer
us_echo_leb	Long Integer
us_echo_sonst	Long Integer
us_echo_sonst_com	Memo
us_hom	Long Integer
ctnat_echo_nie	Long Integer
ctnat_echo_leb	Long Integer
ctnat_echo_sonst	Long Integer
ctnat_echo_sonst_com	Memo
ctnat_hom	Long Integer
ctkm_aufn	Long Integer
ctkm_aufnst	Long Integer
ctkm_hom	Long Integer
mrtt1nat_echo_nie	Long Integer
mrtt1nat_echo_leb	Long Integer
mrtt1nat_echo_sonst	Long Integer
mrtt1nat_echo_sonst_com	Memo
mrtt1nat_hom	Long Integer
mrtt1km_aufn	Long Integer
mrtt1km_aufnst	Long Integer
mrtt1km_hom	Long Integer
mrtt1km_echo_nie	Long Integer
mrtt1km_echo_leb	Long Integer
mrtt1km_echo_sonst	Long Integer
mrtt1km_echo_sonst_com	Memo

mrtt2_echo_nie	Long Integer
mrtt2_echo_leb	Long Integer
mrtt2_echo_sonst	Long Integer
mrtt2_echo_sonst_com	Memo
mrtt2_hom	Long Integer
mrtsonst_echo_nie	Long Integer
mrtsonst_echo_leb	Long Integer
mrtsonst_echo_sonst	Long Integer
mrtsonst_echo_sonst_com	Memo
mrtsonst_hom	Long Integer
tumstruk_com	Memo
nier_erh	Long Integer
nematose	Long Integer
harnst	Long Integer
path_sonst	Long Integer
path_sonst_com	Memo
kont_unauf	Long Integer
kont_reste	Long Integer
kont_nematose	Long Integer
kont_hyp	Long Integer
kont_harnst	Long Integer
kont_dysp	Long Integer
kont_zyst	Long Integer
kont_sonst	Long Integer
kont_sonst_com	Memo

Table 224 vsoc_hlgt_dupes columns.

Name	Data type
soc_code	Long Integer
hlgt_code	Long Integer

Table 225 v_adm_rolle columns.

Name	Data type
idx	Long Integer
r_berat	Long Integer

Table 226 v_meddra_hlgt columns.

Name	Data type
hlgt_code	Long Integer
hlgt_name	Text
soc_code	Long Integer

Table 227 v_meddra_hlt columns.

Name	Data type
hlt_code	Long Integer
hlt_name	Text
hlgt_code	Long Integer

Table 228 v_meddra_llt columns.

Name	Data type
llt_code	Long Integer
llt_name	Text
pt_code	Long Integer

Table 229 v_meddra_pt columns.

Name	Data type
pt_code	Long Integer
pt_name	Text
hlt_code	Long Integer

Table 230 v_meddra_soc columns.

Name	Data type
soc_code	Long Integer
soc_name	Text

Appendix 5 – Schema of the BASE (v2) database

For each of the tables in the database the information about every column is given in the tables that follow, starting at the next page. For each column the name is given, the data type, whether Null values are allowed, and whether or not the values are keys, and if so, of what type. It should be noted that the underlying relational database should not be accessed directly, data is always used as objects in BASE 2. BASE 2 is Java based and the object-relational mapping is handled with Hibernate, http://www.hibernate.org.

Table 231 annotations columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
annotationset_id	int(11)	NO	MUL
annotationtype_id	int(11)	NO	MUL
value_id	int(11)	NO	UNI

Table 232 annotationsets columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
item_type	int(11)	NO	

Table 233 annotationtypecategories columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 234 annotationtypeitems columns.

Name	Data type	Null	Key
annotationtype_id	int(11)	NO	MUL
item_type	int(11)	YES	

Table 235 annotationtypeoptions columns.

Name	Data type	Null	Key
annotationtype_id	int(11)	NO	PRI
value	varchar(255)	NO	
name	varchar(255)	NO	PRI

Table 236 annotationtypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
multiplicity	int(11)	NO	
required_for_miame	bit(1)	NO	
is_enumeration	bit(1)	NO	
height	int(11)	NO	
width	int(11)	NO	
default_value	varchar(255)	YES	
value_type	int(11)	NO	
enumerationvalues_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 237 anytoany columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
from_id	int(11)	NO	
from_type	int(11)	NO	
to_id	int(11)	NO	MUL
to_type	int(11)	NO	
uses_to	bit(1)	NO	

Table 238 arraybatches columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
protocol_id	int(11)	YES	MUL
hardware_id	int(11)	YES	MUL
arraydesign_id	int(11)	NO	MUL
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 239 arraydesignblocks columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
meta_grid_x	int(11)	NO	
meta_grid_y	int(11)	NO	
block_size_x	int(11)	NO	
block_size_y	int(11)	NO	
origin_x	int(11)	NO	
origin_y	int(11)	NO	
spacing_x	int(11)	NO	
spacing_y	int(11)	NO	
block_number	int(11)	NO	
arraydesign_id	int(11)	NO	MUL

Table 240 arraydesignplates columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
position	int(11)	NO	
arraydesign_id	int(11)	NO	MUL
plate_id	int(11)	NO	MUL

Table 241 arraydesigns columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
affy_chip	bit(1)	NO	
has_features	bit(1)	NO	
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 242 arrayslides columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
batch_index	int(11)	NO	
destroyed	bit(1)	NO	
barcode	varchar(255)	YES	
arraybatch_id	int(11)	NO	MUL
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 243 bioassayparents columns.

Name	Data type	Null	Key
bioassay_id	int(11)	NO	PRI
parent_id	int(11)	NO	PRI

Table 244 bioassayrawparents columns.

Name	Data type	Null	Key
bioassay_id	int(11)	NO	PRI
parent_id	int(11)	NO	PRI

Table 245 bioassays columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
bioassayset_id	int(11)	NO	MUL
datacubecolumn_id	int(11)	NO	MUL
spots	int(11)	NO	

Table 246 bioassaysets columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
experiment_id	int(11)	NO	MUL
transformation_id	int(11)	NO	MUL
datacubelayer_id	int(11)	NO	MUL
datacubefilter_id	int(11)	YES	MUL
reporters	int(11)	NO	
spots	int(11)	NO	

Table 247 biomaterialevents columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
event_type	int(11)	NO	
biomaterial_id	int(11)	YES	MUL
used_quantity	float	YES	
hybridization_id	int(11)	YES	MUL
user_id	int(11)	YES	MUL
protocol_id	int(11)	YES	MUL
hardware_id	int(11)	YES	MUL
entry_date	date	NO	
event_date	date	YES	
comment	text	YES	

Table 248 biomaterialeventsources columns.

Name	Data type	Null	Key
biomaterial_id	int(11)	NO	PRI
event_id	int(11)	NO	PRI
used_quantity	float	YES	
dummy	int(11)	NO	

Table 249 biomaterials columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
discriminator	int(11)	NO	
version	int(11)	NO	
external_id	varchar(255)	YES	
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

parent_id	int(11)	YES	MUL
original_quantity	float	YES	
remaining_quantity	float	YES	
pooled	bit(1)	YES	
label_id	int(11)	YES	MUL

Table 250 booleanvalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	bit(1)	NO	

Table 251 categorisedannotationtypes columns.

Name	Data type	Null	Key
category_id	int(11)	NO	PRI
annotationtype_id	int(11)	NO	PRI

Table 252 cliendefaultsettings columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
client_id	int(11)	NO	MUL
name	varchar(255)	NO	
value	text	NO	

Table 253 clients columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
external_id	varchar(255)	NO	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 254 contexts columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
user_id	int(11)	NO	MUL
client_id	int(11)	NO	MUL
name	varchar(255)	NO	
item_type	int(11)	NO	
subcontext	varchar(255)	NO	
is_public	bit(1)	YES	
rows_per_page	int(11)	YES	
page	int(11)	YES	
sort_property	varchar(255)	YES	
sort_direction	int(11)	YES	
include	int(11)	YES	

Table 255 contextsettings columns.

Name	Data type	Null	Key
context_id	int(11)	NO	PRI
value	text	YES	
name	varchar(255)	NO	PRI

Table 256 datacubecolumns columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
datacube_id	int(11)	NO	MUL
column_no	smallint(6)	NO	

Table 257 datacubeextravalues columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
datacube_id	int(11)	NO	MUL
extra_no	smallint(6)	NO	
bytes	bigint(20)	NO	
coordinate_type	int(11)	NO	
value_type	int(11)	NO	

Table 258 datacubefilters columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
datacube_id	int(11)	NO	MUL
filter_no	smallint(6)	NO	
bytes	bigint(20)	NO	

Table 259 datacubelayers columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
datacube_id	int(11)	NO	MUL
layer_no	smallint(6)	NO	
bytes	bigint(20)	NO	

Table 260 datacubes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
virtualdb_id	int(11)	NO	MUL
cube_no	smallint(6)	NO	
bytes	bigint(20)	NO	
filters	smallint(6)	NO	
extravalues	smallint(6)	NO	
layers	smallint(6)	NO	
columns	smallint(6)	NO	
positions	int(11)	NO	

Table 261 datevalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	datetime	NO	

Table 262 directories columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
system_id	varchar(255)	YES	
parent_id	int(11)	YES	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 263 diskusage columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
location	int(11)	NO	
bytes	bigint(20)	NO	
quotatype_id	int(11)	NO	MUL
group_id	int(11)	YES	MUL
user_id	int(11)	NO	MUL

Table 264 doublevalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	double	NO	

Table 265 experimentalfactors columns.

Name	Data type	Null	Key
annotationtype_id	int(11)	NO	PRI
experiment_id	int(11)	NO	PRI

Table 266 experimentrawbioassays columns.

Name	Data type	Null	Key
experiment_id	int(11)	NO	PRI
rawbioassay_id	int(11)	NO	PRI

Table 267 experiments columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
diskusage_id	int(11)	NO	UNI
bytes	bigint(20)	NO	
rawdatatype	varchar(255)	NO	
authors	text	YES	
affiliations	text	YES	
abstract	text	YES	
experiment_design	text	YES	
experiment_type	text	YES	
publication	text	YES	
pubmed_id	varchar(255)	YES	
publication_date	date	YES	
virtualdb_id	int(11)	NO	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 268 extravalues columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
bioassayset_id	int(11)	NO	MUL
extravaluetype_id	int(11)	NO	MUL
datacubeextravalue_id	int(11)	NO	MUL
values	int(11)	NO	

Table 269 extravaluetypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
external_id	varchar(255)	NO	UNI
value_type	int(11)	NO	

Table 270 features columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
column	int(11)	NO	MUL
row	int(11)	NO	
position	int(11)	NO	
arraydesignblock_id	int(11)	NO	MUL
well_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL

Table 271 files columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
diskusage_id	int(11)	NO	UNI
size	bigint(20)	NO	
md5	varchar(32)	YES	
action	int(11)	NO	
location	int(11)	NO	
mimetype	varchar(255)	YES	
directory_id	int(11)	NO	MUL
filetype_id	int(11)	YES	MUL
internalname	varchar(255)	YES	
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 272 filetypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
system_id	varchar(255)	YES	

Table 273 filevalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	int(11)	NO	MUL

Table 274 floatvalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	float	NO	

Table 275 formulaexpressions columns.

Name	Data type	Null	Key
formula_id	int(11)	NO	PRI
formula	varchar(255)	NO	
index	int(11)	NO	PRI

Table 276 formulas columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
type	int(11)	NO	
parser	int(11)	NO	
rawdatatype	varchar(255)	YES	
channels	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 277 globaldefaultsettings columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	UNI
value	text	NO	

Table 278 groupgroups columns.

Name	Data type	Null	Key
parent_id	int(11)	NO	PRI
child_id	int(11)	NO	PRI

Table 279 groupkeys columns.

Name	Data type	Null	Key
group_id	int(11)	NO	PRI
key_id	int(11)	NO	PRI
permission	int(11)	NO	

Table 280 groupprojects columns.

Name	Data type	Null	Key
group_id	int(11)	NO	PRI
project_id	int(11)	NO	PRI
permission	int(11)	NO	

Table 281 groups columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
system_id	varchar(255)	YES	
quota_id	int(11)	YES	MUL

Table 282 hardware columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
hardwaretype_id	int(11)	NO	MUL
version_string	varchar(255)	YES	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 283 hardwaretypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
system_id	varchar(255)	YES	

Table 284 helptexts columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
client_id	int(11)	NO	MUL
external_id	varchar(255)	NO	

Table 285 hybridizations columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
arrayslide_id	int(11)	YES	UNI
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	

itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 286 images columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
file_id	int(11)	YES	MUL
scan_id	int(11)	NO	MUL
format	int(11)	NO	
is_preview	bit(1)	NO	

Table 287 inheritedannotations columns.

Name	Data type	Null	Key
annotation_id	int(11)	NO	PRI
annotationset_id	int(11)	NO	PRI

Table 288 inheritedannotationsets columns.

Name	Data type	Null	Key
annotationset_id	int(11)	NO	PRI
inherited_id	int(11)	NO	PRI

Table 289 integervalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	int(11)	NO	

Table 290 itemvalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
data_class	varchar(255)	NO	
data_class_id	int(11)	NO	

Table 291 jobparameters columns.

Name	Data type	Null	Key
job_id	int(11)	NO	PRI
value_id	int(11)	NO	MUL
name	varchar(255)	NO	PRI

Table 292 jobs columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
type	int(11)	NO	
plugindefinition_id	int(11)	YES	MUL
pluginconfiguration_id	int(11)	YES	MUL
status	int(11)	NO	
status_message	text	YES	
execution_time	int(11)	NO	
percent_complete	int(11)	NO	
priority	int(11)	NO	
project_id	int(11)	NO	
created	datetime	NO	
started	datetime	YES	
ended	datetime	YES	
server	varchar(255)	YES	
owner	int(11)	NO	MUL

Table 293 keys columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
discriminator	int(11)	NO	
version	int(11)	NO	
name	varchar(255)	YES	MUL
description	text	YES	
item_type	int(11)	YES	

Table 294 labels columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 295 longvalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	bigint(20)	NO	

Table 296 messages columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
to_user_id	int(11)	NO	MUL
from_name	varchar(255)	NO	
from_user_id	int(11)	YES	
time_sent	datetime	NO	
job_id	int(11)	YES	MUL
is_read	bit(1)	NO	

Table 297 mimetypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
extension	varchar(255)	NO	UNI

Table 298 news columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
start_date	date	NO	
news_date	date	NO	
end_date	date	YES	

Table 299 parametervalues columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
discriminator	int(11)	NO	
version	int(11)	NO	
label	varchar(255)	YES	
description	text	YES	

Table 300 passwords columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
md5password	varchar(32)	NO	

Table 301 plateevents columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
plate_id	int(11)	NO	MUL
plateeventtype_id	int(11)	NO	MUL
protocol_id	int(11)	YES	MUL
user_id	int(11)	YES	MUL
entry_date	date	NO	
event_date	date	YES	
hardware_id	int(11)	YES	MUL
comment	text	YES	

Table 302 plateeventtypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
platetype_id	int(11)	NO	MUL
ordinal	int(11)	NO	
protocoltype_id	int(11)	YES	MUL

Table 303 plategeometries columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
rows	int(11)	NO	
columns	int(11)	NO	

Table 304 platemappingdetails columns.

Name	Data type	Null	Key
platemapping_id	int(11)	NO	PRI
source_plate	int(11)	YES	
source_row	int(11)	YES	
source_column	int(11)	YES	
destination_plate	int(11)	NO	PRI
destination_row	int(11)	NO	PRI
destination_column	int(11)	NO	PRI

Table 305 platemappings columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
sourcegeometry_id	int(11)	NO	MUL
destinationgeometry_id	int(11)	NO	MUL
source_count	int(11)	NO	
destination_count	int(11)	NO	
image	varchar(255)	YES	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 306 plateparents columns.

Name	Data type	Null	Key

plate_id	int(11)	NO	PRI
source_index	int(11)	YES	
parent_id	int(11)	NO	PRI

Table 307 plates columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
platetype_id	int(11)	NO	MUL
barcode	varchar(255)	YES	
destroyed	bit(1)	NO	
platemapping_id	int(11)	YES	MUL
destination_index	int(11)	NO	
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 308 platetypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
plategeometry_id	int(11)	NO	MUL
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 309 pluginconfigurations columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
plugindefinition_id	int(11)	NO	MUL
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 310 pluginconfigurationvalues columns.

Name	Data type	Null	Key
pluginconfiguration_id	int(11)	NO	PRI
value_id	int(11)	NO	MUL
name	varchar(255)	NO	PRI

Table 311 plugindefinitionguicontexts columns.

Name	Data type	Null	Key
plugindefinition_id	int(11)	NO	MUL
item_type	int(11)	YES	
context_type	int(11)	YES	

Table 312 plugindefinitions columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
removed	bit(1)	NO	
class_name	varchar(255)	NO	UNI
jar_path	text	YES	
name	varchar(255)	NO	
description	text	YES	
version_string	varchar(255)	YES	

copyright	varchar(255)	YES	
contact	varchar(255)	YES	
email	varchar(255)	YES	
url	varchar(255)	YES	
type	int(11)	NO	
interactive	bit(1)	NO	
supports_config	bit(1)	NO	
requires_config	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 313 plugindefinitiontypes columns.

Name	Data type	Null	Key
plugindefinition_id	int(11)	NO	PRI
plugintype_id	int(11)	NO	PRI

Table 314 plugintypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
interface_name	varchar(255)	NO	UNI
jar_path	text	YES	

Table 315 projectkeys columns.

Name	Data type	Null	Key
project_id	int(11)	NO	PRI
key_id	int(11)	NO	PRI
permission	int(11)	NO	

Table 316 projects columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
owner	int(11)	NO	MUL

Table 317 propertyfilters columns.

Name	Data type	Null	Key
context_id	int(11)	NO	PRI
operator	int(11)	NO	
value_type	int(11)	NO	
value	text	YES	
property	varchar(255)	NO	PRI

Table 318 protocols columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
file_id	int(11)	YES	MUL
protocoltype_id	int(11)	NO	MUL
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 319 protocoltypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	

name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
system_id	varchar(255)	YES	

Table 320 quota columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
removed	bit(1)	NO	
system_id	varchar(255)	YES	
name	varchar(255)	NO	MUL
description	text	YES	

Table 321 quotatypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
system_id	varchar(255)	YES	
secondary_location	bit(1)	NO	

Table 322 quotavalues columns.

Name	Data type	Null	Key
quota_id	int(11)	NO	PRI
max_bytes	bigint(20)	NO	
location	int(11)	NO	PRI
quotaType_id	int(11)	NO	PRI

Table 323 rawbioassayheaders columns.

Name	Data type	Null	Key
rawbioassay_id	int(11)	NO	PRI
value	text	NO	
name	varchar(255)	NO	PRI

Table 324 rawbioassays columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
diskusage_id	int(11)	NO	UNI
scan_id	int(11)	YES	MUL
software_id	int(11)	YES	MUL
protocol_id	int(11)	YES	MUL
arraydesign_id	int(11)	YES	MUL
rawdatatype	varchar(255)	NO	
has_data	bit(1)	NO	
spots	int(11)	NO	
bytes	bigint(20)	NO	
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 325 rawdataagilent columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	

metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
log_ratio	float	YES	
pvalue_log_ratio	float	YES	
g_surrogate_used	float	YES	
r_surrogate_used	float	YES	
g_is_found	bit(1)	YES	
r_is_found	bit(1)	YES	
g_processed_signal	float	YES	
r_processed_signal	float	YES	
g_high_outlier_pixels	int(11)	YES	
r_high_outlier_pixels	int(11)	YES	
g_low_outlier_pixels	int(11)	YES	
r_low_outlier_pixels	int(11)	YES	
g_pixels	int(11)	YES	
r_pixels	int(11)	YES	
g_mean_signal	float	YES	
r_mean_signal	float	YES	
g_median_signal	float	YES	
r_median_signal	float	YES	
g_bg_pixels	int(11)	YES	
r_bg_pixels	int(11)	YES	
g_bg_mean_signal	float	YES	
r_bg_mean_signal	float	YES	
g_bg_median_signal	float	YES	
r_bg_median_signal	float	YES	
g_sat_pixels	int(11)	YES	
r_sat_pixels	int(11)	YES	
g_is_saturated	bit(1)	YES	
r_is_saturated	bit(1)	YES	
pix_correlation	float	YES	
bg_pix_correlation	float	YES	
g_nonuniform_outlier	bit(1)	YES	
r_nonuniform_outlier	bit(1)	YES	
g_bg_nonuniform_outlier	bit(1)	YES	
r_bg_nonuniform_outlier	bit(1)	YES	
g_population_outlier	bit(1)	YES	
r_population_outlier	bit(1)	YES	
g_bg_population_outlier	bit(1)	YES	
r_bg_population_outlier	bit(1)	YES	
manual_flag	bit(1)	YES	
g_net_signal	float	YES	
r_net_signal	float	YES	
bg_correlation	float	YES	
g_significant	bit(1)	YES	
r_significant	bit(1)	YES	
g_pvalue	float	YES	
r_pvalue	float	YES	
g_numbg_used	int(11)	YES	
r_numbg_used	int(11)	YES	
g_above_bg	bit(1)	YES	
r_above_bg	bit(1)	YES	
g_bg_used	float	YES	
r_bg_used	float	YES	
is_normalization	bit(1)	YES	
g_normalized	float	YES	
r_normalized	float	YES	
norm_correlation	float	YES	
error_model	int(11)	YES	
x_dev	float	YES	
g_spatial_trend	bit(1)	YES	
r_spatial_trend	bit(1)	YES	
g_surface_value	float	YES	
r_surface_value	float	YES	

Table 326 rawdataaida columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL

reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
area	float	YES	
ch1_integral	float	YES	
ch1_mean	float	YES	
ch1_median	float	YES	
ch1_bkg	float	YES	
ch1_sat	float	YES	
ch1_hom	float	YES	
ch1_quality	bit(1)	YES	
ch2_integral	float	YES	
ch2_mean	float	YES	
ch2_median	float	YES	
ch2_bkg	float	YES	
ch2_sat	float	YES	
ch2_hom	float	YES	
ch2_quality	bit(1)	YES	

Table 327 rawdatabzscan columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
q_image_const	float	YES	
q_image_var	float	YES	
q_fit_const	float	YES	
q_fit_var	float	YES	
fit_correction	float	YES	
quality_metric	float	YES	
spot_quality	bit(1)	YES	
overshining	float	YES	

Table 328 rawdatachipskipper columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
line_counter	int(11)	YES	
flags	varchar(255)	YES	
gs_flags	varchar(255)	YES	
cy3_volume	int(11)	YES	
cy3_avg	float	YES	
cy3_bg_arith	float	YES	
cy3_bg_geom	float	YES	
cy3_bg_median	float	YES	

cy3_bg_total	float	YES	
cy3_reduced	float	YES	
cy3_area	float	YES	
cy3_saturated	float	YES	
cy5_volume	int(11)	YES	
cy5_avg	float	YES	
cy5_bg_arith	float	YES	
cy5_bg_geom	float	YES	
cy5_bg_median	float	YES	
cy5_bg_total	float	YES	
cy5_reduced	float	YES	
cy5_area	float	YES	
cy5_saturated	float	YES	
rep_vol	float	YES	
ratio	float	YES	
directional_ratio	float	YES	
log2_ratio	float	YES	
cy3_normalised	float	YES	
cy5_normalised	float	YES	
compensated_ratio	float	YES	
comp_dir_ratio	float	YES	
comp_log2_ratio	float	YES	
comment	varchar(255)	YES	
snr_tot	float	YES	
snr_tot_score	float	YES	
sat_pixels	float	YES	
replicas	float	YES	
regulation	varchar(255)	YES	
group_name	varchar(255)	YES	
median_ratios	float	YES	
median_ratios_sdev	float	YES	
ratio_medians	float	YES	
ratio_medians_sdev	float	YES	

Table 329 rawdatagenepix columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
diameter	float	YES	
ch1_fg_median	float	YES	
ch1_fg_mean	float	YES	
ch1_fg_sd	float	YES	
ch1_bg_median	float	YES	
ch1_bg_mean	float	YES	
ch1_bg_sd	float	YES	
ch1_perc_sd1	int(11)	YES	
ch1_perc_sd2	int(11)	YES	
ch1_perc_sat	int(11)	YES	
ch2_fg_median	float	YES	
ch2_fg_mean	float	YES	
ch2_fg_sd	float	YES	
ch2_bg_median	float	YES	
ch2_bg_mean	float	YES	
ch2_bg_sd	float	YES	
ch2_perc_sd1	int(11)	YES	
ch2_perc_sd2	int(11)	YES	
ch2_perc_sat	int(11)	YES	
ratios_sd	float	YES	
rgn_ratio	float	YES	
rgn_r2	float	YES	
fg_pixels	int(11)	YES	
bg_pixels	int(11)	YES	

flags	int(11)	YES	
m_value	float	YES	
cv	float	YES	

Table 330 rawdatagenetac columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
med_cy3	float	YES	
med_cy5	float	YES	
med_ratio	float	YES	
bg_cy3	float	YES	
bg_cy5	float	YES	
bg_ratio	float	YES	

Table 331 rawdataimagene columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
field	varchar(255)	YES	
cy3_flag	int(11)	YES	
cy3_mean_signal	float	YES	
cy3_mean_bg	float	YES	
cy3_median_signal	float	YES	
cy3_median_bg	float	YES	
cy3_mode_signal	float	YES	
cy3_mode_bg	float	YES	
cy3_area_signal	int(11)	YES	
cy3_area_bg	int(11)	YES	
cy3_total_signal	int(11)	YES	
cy3_total_bg	int(11)	YES	
cy3_shape	float	YES	
cy3_area_ignored	int(11)	YES	
cy3_area_spot	int(11)	YES	
cy3_median_ignored	float	YES	
cy3_area_perimeter	float	YES	
cy3_open_perimeter	float	YES	
cy3_x	float	YES	
cy3_y	float	YES	
cy3_diameter	int(11)	YES	
cy3_offset	float	YES	
cy3_offset_x	float	YES	
cy3_offset_y	float	YES	
cy3_x_expected	float	YES	
cy3_y_expected	float	YES	
cy3_x_cm	float	YES	
cy3_y_cm	float	YES	
cy3_offset_cm	float	YES	
cy3_offset_x_cm	float	YES	
cy3_offset_y_cm	float	YES	

cy3_diameter_min	float	YES	
cy3_diameter_max	float	YES	
cy3_control	varchar(255)	YES	
cy3_control_failed	bit(1)	YES	
cy3_bg_contamination	bit(1)	YES	
cy3_signal_contamination	bit(1)	YES	
cy3_ignored_failed	bit(1)	YES	
cy3_perimeter_failed	bit(1)	YES	
cy3_shape_failed	bit(1)	YES	
cy3_areapm_failed	bit(1)	YES	
cy3_offset_failed	bit(1)	YES	
cy3_empty_spot	bit(1)	YES	
cy3_negative_spot	bit(1)	YES	
cy3_selected_spot	bit(1)	YES	
cy3_saturated_spot	bit(1)	YES	
cy5_mean_signal	float	YES	
cy5_mean_bg	float	YES	
cy5_median_signal	float	YES	
cy5_median_bg	float	YES	
cy5_mode_signal	float	YES	
cy5_mode_bg	float	YES	
cy5_area_signal	int(11)	YES	
cy5_area_bg	int(11)	YES	
cy5_total_signal	int(11)	YES	
cy5_total_bg	int(11)	YES	
cy5_shape	float	YES	
cy5_area_ignored	int(11)	YES	
cy5_area_spot	int(11)	YES	
cy5_median_ignored	float	YES	
cy5_area_perimeter	float	YES	
cy5_open_perimeter	float	YES	
cy5_x	float	YES	
cy5_y	float	YES	
cy5_diameter	int(11)	YES	
cy5_offset	float	YES	
cy5_offset_x	float	YES	
cy5_offset_y	float	YES	
cy5_x_expected	float	YES	
cy5_y_expected	float	YES	
cy5_x_cm	float	YES	
cy5_y_cm	float	YES	
cy5_offset_cm	float	YES	
cy5_offset_x_cm	float	YES	
cy5_offset_y_cm	float	YES	
cy5_diameter_min	float	YES	
cy5_diameter_max	float	YES	
cy5_control	varchar(255)	YES	
cy5_control_failed	bit(1)	YES	
cy5_bg_contamination	bit(1)	YES	
cy5_signal_contamination	bit(1)	YES	
cy5_ignored_failed	bit(1)	YES	
cy5_perimeter_failed	bit(1)	YES	
cy5_shape_failed	bit(1)	YES	
cy5_areapm_failed	bit(1)	YES	
cy5_offset_failed	bit(1)	YES	
cy5_empty_spot	bit(1)	YES	
cy5_negative_spot	bit(1)	YES	
cy5_selected_spot	bit(1)	YES	
cy5_saturated_spot	bit(1)	YES	

Table 332 rawdataquantarraybiotin columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	

row	int(11)	YES	
column	int(11)	YES	
number	int(11)	YES	
intensity	float	YES	
background	float	YES	
diameter	float	YES	
area	int(11)	YES	
footprint	float	YES	
cirularity	float	YES	
uniformity	float	YES	
bg_uniformity	float	YES	
snr	float	YES	
confidence	int(11)	YES	
ignore_filter	int(11)	YES	
ratio	int(11)	YES	
precent	float	YES	
diameter_filter	int(11)	YES	
area_filter	int(11)	YES	
footprint_filter	int(11)	YES	
circularity_filter	int(11)	YES	
uniformity_filter	int(11)	YES	
bg_uniformity_filter	int(11)	YES	
snr_filter	int(11)	YES	
replicate_filter	int(11)	YES	

Table 333 rawdataquantarraycy columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
cy3_mean	float	YES	
cy3_mean_bg	float	YES	
cy3_diameter	float	YES	
cy3_area	int(11)	YES	
cy3_footprint	float	YES	
cy3_circularity	float	YES	
cy3_uniformity	float	YES	
cy3_bg_uniformity	float	YES	
cy3_snr	float	YES	
cy3_confidence	int(11)	YES	
cy5_mean	float	YES	
cy5_mean_bg	float	YES	
cy5_diameter	float	YES	
cy5_area	int(11)	YES	
cy5_footprint	float	YES	
cy5_circularity	float	YES	
cy5_uniformity	float	YES	
cy5_bg_uniformity	float	YES	
cy5_snr	float	YES	
cy5_confidence	int(11)	YES	
ignore_filter	int(11)	YES	
cy3_ratio	float	YES	
cy3_percent	float	YES	
cy3_snr_filter	float	YES	
cy5_ratio	float	YES	
cy5_percent	float	YES	
cy5_snr_filter	float	YES	
log_normalised	float	YES	

Table 334 rawdataspotfinder columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI

version	int(11)	NO	
rawbioassay_id	int(11)	NO	MUL
feature_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL
position	int(11)	NO	
x	float	YES	
y	float	YES	
block	int(11)	YES	
metagrid_x	int(11)	YES	
metagrid_y	int(11)	YES	
row	int(11)	YES	
column	int(11)	YES	
int_a	int(11)	YES	
int_b	int(11)	YES	
mean_ratio	float	YES	
area	int(11)	YES	
saturation	float	YES	
median_ratio	float	YES	
mode_ratio	float	YES	
bg_a	int(11)	YES	
bg_b	int(11)	YES	
flag_a	varchar(255)	YES	
flag_b	varchar(255)	YES	
qc_a	float	YES	
qc_b	float	YES	
qc_total	float	YES	
flag	varchar(255)	YES	

Table 335 reporterlists columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
external_id	varchar(255)	YES	
experiment_id	int(11)	YES	MUL
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 336 reporterlistscores columns.

Name	Data type	Null	Key
reporterlist_id	int(11)	NO	PRI
reporter_id	int(11)	NO	PRI
score	float	YES	

Table 337 reporters columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
external_id	varchar(255)	NO	UNI
symbol	varchar(255)	YES	
last_update	datetime	NO	
name	varchar(255)	NO	MUL
description	text	YES	
reportertype_id	int(11)	YES	MUL
species	varchar(255)	YES	
clusterId	varchar(255)	YES	
length	int(11)	YES	
sequence	text	YES	
vector	text	YES	
tissue	text	YES	
library	text	YES	
accession	varchar(255)	YES	
nid	varchar(255)	YES	
chromosome	varchar(255)	YES	
cytoband	varchar(255)	YES	
markers	varchar(255)	YES	
antibiotics	varchar(255)	YES	
locuslink	varchar(255)	YES	

omim	varchar(255)	YES	

Table 338 reportertypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	

Table 339 rolekeys columns.

Name	Data type	Null	Key
role_id	int(11)	NO	PRI
key_id	int(11)	NO	PRI
permission	int(11)	NO	

Table 340 roles columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
system_id	varchar(255)	YES	

Table 341 scans columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
hybridization_id	int(11)	NO	MUL
hardware_id	int(11)	YES	MUL
protocol_id	int(11)	YES	MUL
annotationset_id	int(11)	YES	UNI
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 342 schemaversion columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
schema_version	int(11)	NO	
build	int(11)	NO	

Table 343 sessions columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
user_id	int(11)	NO	MUL
login_time	datetime	NO	
logout_time	datetime	YES	
login_comment	text	YES	
impersonated	bit(1)	NO	
client_id	int(11)	YES	MUL
remote_id	varchar(255)	YES	

Table 344 software columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
softwaretype_id	int(11)	NO	MUL
version_string	varchar(255)	YES	
name	varchar(255)	NO	MUL
description	text	YES	

removed	bit(1)	NO	
itemkey_id	int(11)	YES	MUL
projectkey_id	int(11)	YES	MUL
owner	int(11)	NO	MUL

Table 345 softwaretypes columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
system_id	varchar(255)	YES	

Table 346 spotimages columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
rawbioassay_id	int(11)	NO	UNI
x_scale	int(11)	NO	
y_scale	int(11)	NO	
x_offset	int(11)	NO	
y_offset	int(11)	NO	
spotsize	int(11)	NO	
quality	float	NO	
gamma	float	NO	
redfile_id	int(11)	YES	MUL
greenfile_id	int(11)	YES	MUL
bluefile_id	int(11)	YES	MUL
spotimagesfile_id	int(11)	YES	MUL

Table 347 stringvalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	varchar(255)	NO	

Table 348 textvalues columns.

Name	Data type	Null	Key
id	int(11)	NO	MUL
value	text	NO	

Table 349 transformationrawbioassays columns.

Name	Data type	Null	Key
rawbioassay_id	int(11)	NO	PRI
transformation_id	int(11)	NO	PRI

Table 350 transformations columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
experiment_id	int(11)	NO	MUL
bioassayset_id	int(11)	YES	MUL
job_id	int(11)	YES	MUL

Table 351 userclientsettings columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
user_id	int(11)	NO	MUL
client_id	int(11)	NO	MUL
name	varchar(255)	NO	
value	text	NO	

Table 352 userdefaultsettings columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI

version	int(11)	NO	
user_id	int(11)	NO	MUL
name	varchar(255)	NO	
value	text	NO	

Table 353 usergroups columns.

Name	Data type	Null	Key
user_id	int(11)	NO	PRI
group_id	int(11)	NO	PRI

Table 354 userkeys columns.

Name	Data type	Null	Key
user_id	int(11)	NO	PRI
key_id	int(11)	NO	PRI
permission	int(11)	NO	

Table 355 userprojects columns.

Name	Data type	Null	Key
user_id	int(11)	NO	PRI
project_id	int(11)	NO	PRI
permission	int(11)	NO	

Table 356 userroles columns.

Name	Data type	Null	Key
user_id	int(11)	NO	PRI
role_id	int(11)	NO	PRI

Table 357 users columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
name	varchar(255)	NO	MUL
description	text	YES	
removed	bit(1)	NO	
system_id	varchar(255)	YES	
multiuser_account	bit(1)	NO	
expiration_date	date	YES	
disabled	bit(1)	NO	
external_id	varchar(255)	YES	UNI
login	varchar(255)	NO	UNI
organisation	varchar(255)	YES	
address	varchar(255)	YES	
phone	varchar(255)	YES	
fax	varchar(255)	YES	
email	varchar(255)	YES	
url	varchar(255)	YES	
quotagroup_id	int(11)	YES	MUL
quota_id	int(11)	NO	MUL
homedirectory_id	int(11)	YES	MUL

Table 358 virtualdbs columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
cubes	smallint(6)	NO	
mappings	smallint(6)	NO	

Table 359 wells columns.

Name	Data type	Null	Key
id	int(11)	NO	PRI
version	int(11)	NO	
annotationset_id	int(11)	YES	UNI
plate_id	int(11)	NO	MUL
row	int(11)	NO	
column	int(11)	NO	
parent_id	int(11)	YES	MUL
reporter_id	int(11)	YES	MUL