SEVENTH FRAMEWORK PROGRAMME

Capacities Specific Programme

Research Infrastructures

Project No.: 227887

SERIES

SEISMIC ENGINEERING RESEARCH INFRASTRUCTURES FOR EUROPEAN SYNERGIES

Workpackage [WP2]

Deliverable [D2.5] – 2nd version of Distributed Database and of Data Access Portal including user manual, documentation and guidelines

Deliverable/Task Leader: UPAT

Revision: Final

February, 2012

ABSTRACT

This deliverable presents the 2nd version of Distributed Database and includes design and implementation issues related to the SERIES Data Access Portal (DAP). The Data Access Portal is developed following a user-centred iterative design cycle aiming to provide useful and usable services related to information retrieval functionalities to a wide range of stakeholders, organizations and individuals. The objective of this deliverable is to provide an overview of the 2nd version of the SERIES Data Access Portal including user manual, documentation and guidelines. The Data Access Portal is public accessible and can be reviewed in detail through the following unified resource location address: http://www.dap.series.upatras.gr.

Keywords: Data Access Portal, Series Distributed Database, User Manual, Documentation, Guidelines

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 227887.

DELIVERABLE CONTRIBUTORS

UPAT	Ioannidis Ioannis
UOXF-DF	Ignacio Lamata Martínez
JRC	Pierre Pegon
UPAT	Fidas Christos
UPAT	Bousias Stathis
UPAT	Avouris Nikolaos

CONTENTS

1 User Manual of the Data Access Portal

The aim of the SERIES Data Access Portal (DAP) is to provide a centralized way for accessing all the public projects from the SERIES community. The Data Access Portal presents the information of the available projects by following the structure of the Exchange Data Format (Deliverable 2.1) and having a basic understanding of EDF (see as well section 1.1.2) is considered **useful** for understanding how the Data Access Portal is structured. The Data Access Portal provides a brief description related to the Exchange Data Format.

Conceptual Design of the Data Access Portal

From a conceptual point of view the Data Access Portal has been designed to act as an information space. Organizing functionality and content into a structure that users are able to navigate intuitively is not a trivial task. Researching the suitable Information Architecture of the DAP environment is of great importance. Effective information architecture enables users to step logically through a system aiming to supporting them getting closer to the information they require. Lacking a suitable Information flow increases the risk of creating great content and functionality that no one can ever find. The proposed Information Architecture is based on the fact that the content is not going to be created by a group of administrators or content authors. The content will be mostly fed into the system by the distributed databases that are maintained on the laboratories sides. However, the distributive character of the database makes the decision of the suitable information containers much more difficult. Two questions are the most prominent in this decision process:

- What is important and for whom?
- What has to be accessible and for whom?

The Information Architecture of the system needs to provide rational answers to these questions satisfying the majority of – if not all – users. The proposed platform uses a “Pull” (or self-subscribe) rather than a “Push” model for the Information flow and the Notification system, in

order to fulfil the above statement. That means, that each user selects what is important for him and thus reaches it with less effort ("Push" functionality regarding the notification of users could be available, but that does not reflect the general philosophy of the platform).

In terms of user interaction functionalities the Data Access Portal supports two complementary modes of information retrieval: a) direct search functionality (see as well section 1.2) and b) direct navigation functionalities (see as well section 1.3) which are explained further in the Data Access Portal overview section .

1.1.1 External Actors

From an architectural point of view, the Data Access Portal has been designed to support two different external actors:

- The external users who will interact with the SERIES web portal in order to perform information retrieval tasks and
- The Laboratory Web Services, which will interact with the Central Site (more specifically with the Central Web Service) in order to exchange content and configuration. The security model which will be used among the Web Services for their communication has been described in previous deliverables.

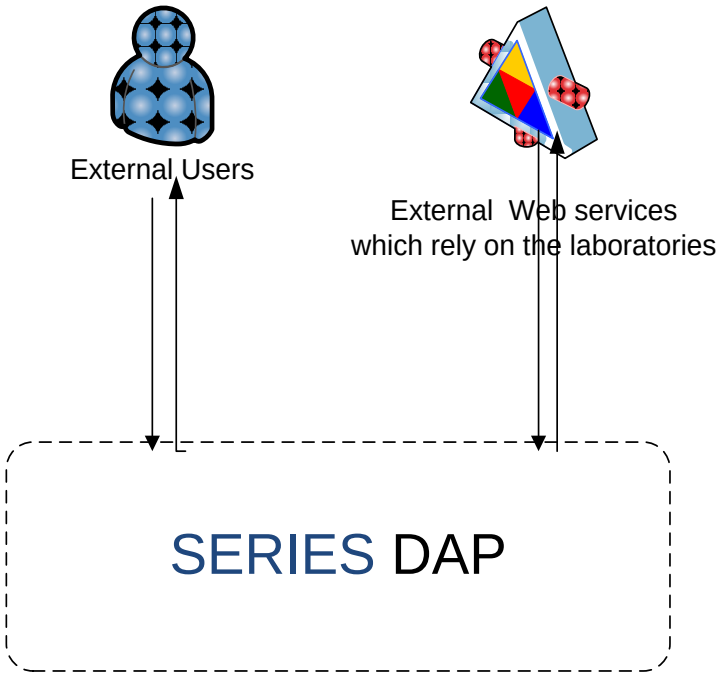


Figure 1: External Actors of the SERIES Central Site

From a software component point of view, as shown in Figures 2 and 3, the Data Access Portal consists of the following components:

- the SERIES distributed database, which entails the searchable part of the published projects (an overview of the Entity Relation Diagram is shown in Appendix A)
- the SERIES central web services, which communicate with the laboratories in order to exchange information on published projects but as well configuration settings related to privacy issues (an example of some Web Services are shown in Appendix A)
- the SERIES web server, which also hosts the Data Access Portal which is described in this deliverable

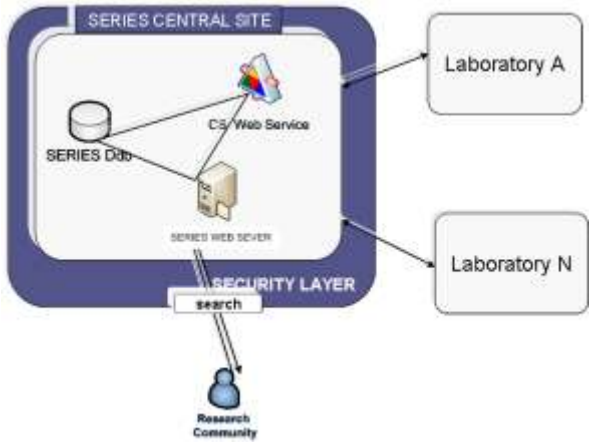


Figure 2: Component View of the SERIES Central Site

A more detailed component view, which entails as well the software components relying on the laboratory side, can be seen in Figure 3.

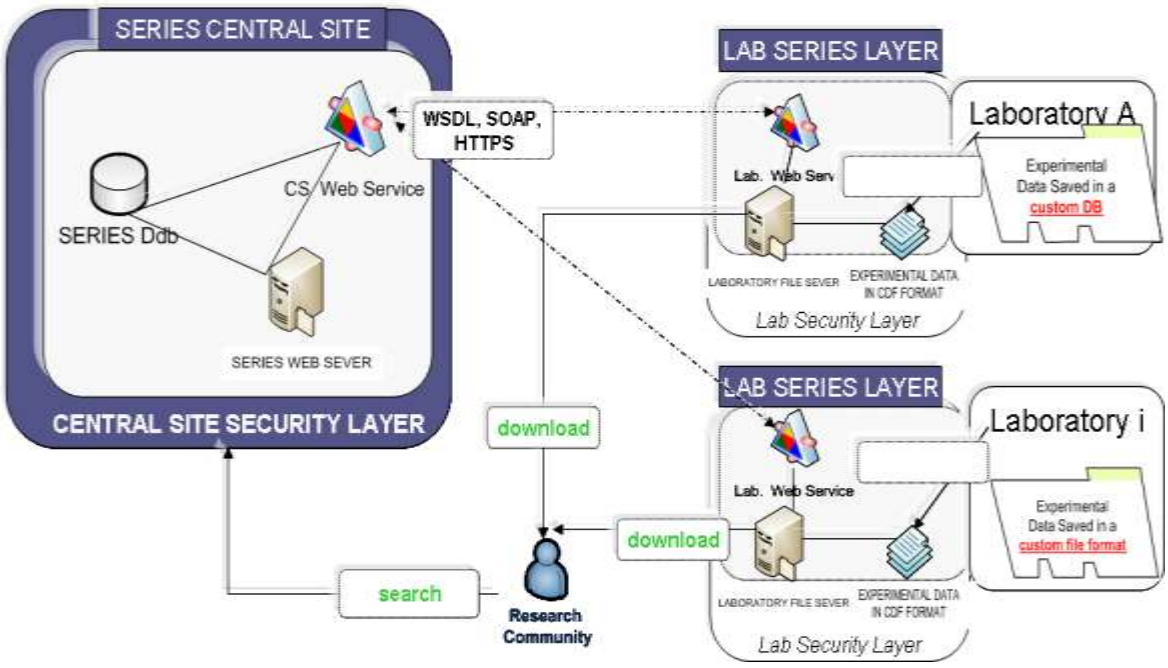


Figure 3: Component View of the SERIES and Laboratories Site

1.1.2 SERIES Projects Types and Exchange Data Format

The European scientific community is currently fragmented with each laboratory holding experimental data with a unique local data model and user interface, language and scheme. As a consequence, the dissemination and use of these experimental results outside of the laboratory where they are produced can be problematic. To overcome this, it is proposed to add a layer on top of the existing local databases that is accessible through a unique Data Access Portal. The aim is not to build a central database where local databases either migrate or merge but instead to provide centralised access to database nodes that are distributed over the network which are able to dialog with a central portal in a uniform manner.

In this context two district types of projects are supported: a) public and b) partner projects. These supported types are distinguished based on the privacy level they utilize. The public projects are available to any visitor of the Data Access Portal whereas the Partner projects are available only to the member of the SERIES consortium.

The Data Access Portal presents information related to published projects according to the Data Exchange Format. According to the Exchange Data Format a published project, in the frame of the SERIES community, embrace information organized on several levels of abstraction (i.e. specimen level, experiment level, computation level and signal level). As shown in figure 5, each specimen consists of information related to the specimen, the experiment and the computation level, whereas, each experiment or computation embraces as well information related to the signal level.

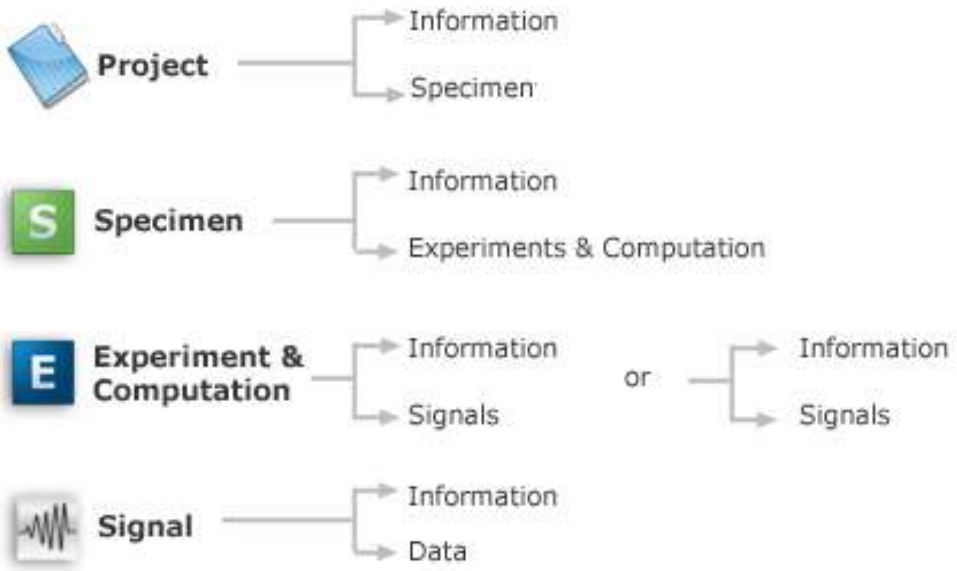


Figure 4: Exchange Data Format

A detailed presentation of the Exchange Data Format is presented in the deliverable D2.1 and is available through the entry page of the Data Access Portal.

Direct Navigation Functionalities

In Figure 1, the home page of the Data Access Portal is shown which is divided into two main panels, the left panel which contains all the available projects and the right panel which provides general information related to the Data Access Portal, project specific information and also the search functionality.

Figure 5: The Main Information Presentation Areas

In this context, the right pane of the DAP provides information related to:

- The general purpose of the Data Access Portal and its mission and vision which is the creation of the distributed database aims to improve the dissemination and use of experimental results and to foster the impact of earthquake engineering research on practice, innovation and earthquake risk mitigation.
- The Exchange Data Format: A small introduction about the Exchange Data Format and a direct link to the whole specification and detailed analysis of the Exchange Data Format. An understanding of the EDF format is considered useful in order to understand in short time the structure of the published projects on the Data Access Portal
- Information related to the last published project: The title and a small description of the last published project
- The user manual: The user manual of the Data Access Portal which is the current document and is accessible through the central page of the Data Access Portal

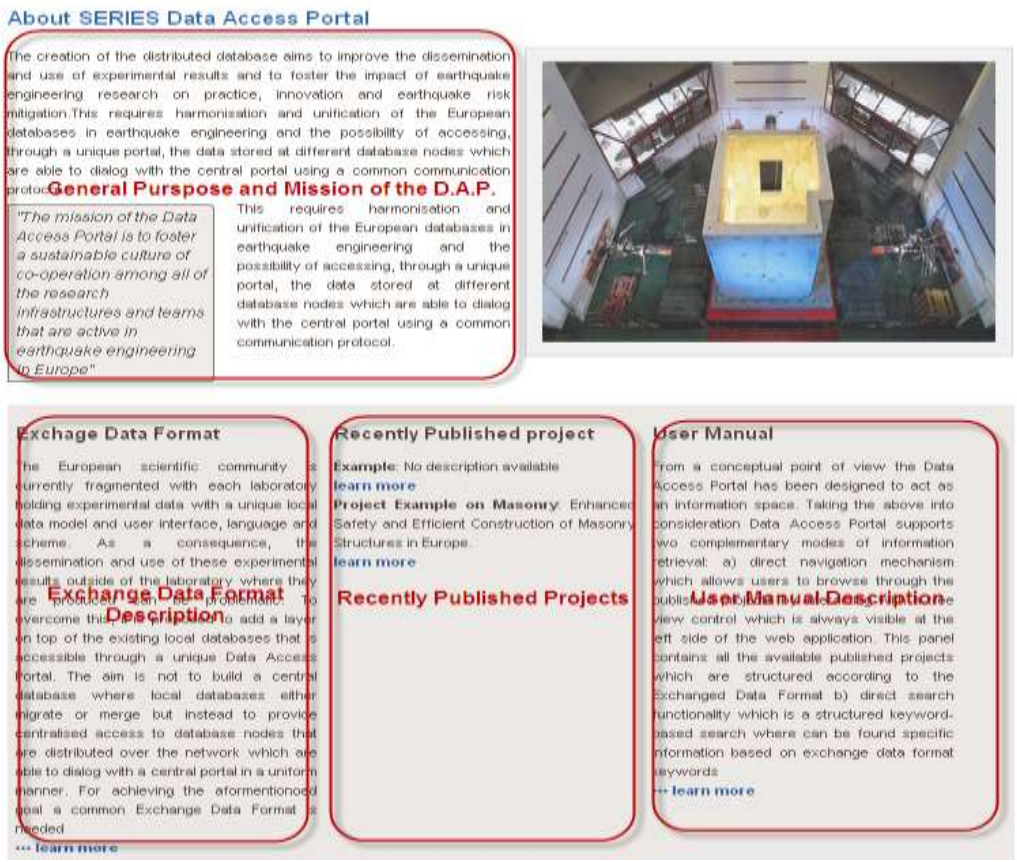


Figure 6: Right Pane

1.1.3 Projects Ordering Options

Aiming to adapt content presentation according to users individual needs the Data Access Portal implements multiple data presentation features implemented through visual direct manipulation control. As it can be seen in figure 6, the tree view control can be structured with three (3) different ways using the ***"PROJECTS ORDER BY"*** list box:

- ***"Project Creation Date"***: Through this selection the projects are ordered according to their Creation Date which is also the default selected value
- ***"Project Name"***: Through this selection the projects are ordered alphabetically according to their Project Name
- ***"Laboratory Name"***: Through this selection the projects are ordered according to the laboratory they belong to. In such case, the laboratory names are displayed on the left pane of the Data Access Portal along with the project information.

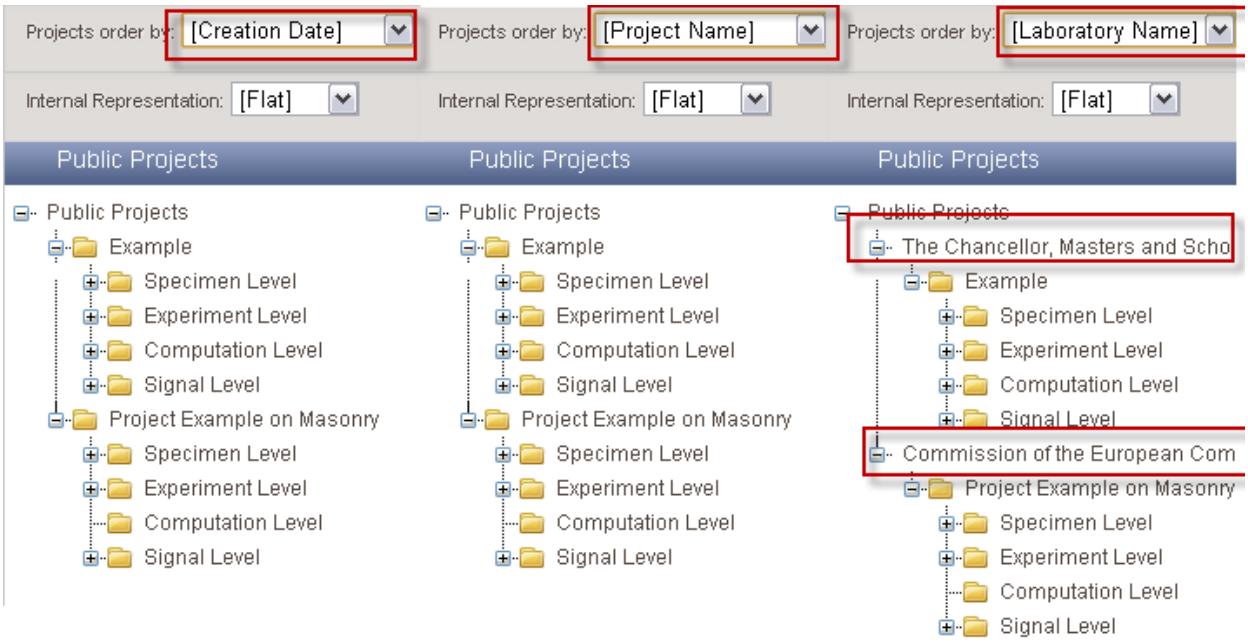


Figure 7: Projects Ordered by List and its effect on the Tree View

1.1.4 Project Internal Representation Options

Furthermore, having the aim to provide bootstrapped functionalities to diverse user groups the Data Access Portal offers two complementary ways of presenting information of available projects. These diverse information representation modes ("***Flat"*** and "***Layers"***) are available through the ***"INTENAL REPRESENTATION"*** list box.

As it is shown in Figure 7, the flat option keeps all the levels visible under one level:

- ***"Specimen Level"***: The specimens that a project contains are visible under the specimen level. Even though a specimen may have experiments and computations these are not visible in the specimen level, but in the next level.
- ***"Experiment Level***": All the available experiments are shown here. Experiments are presented under the specimen they belong to. Specimen that doesn't have experiments are not **included** here.
- ***"Computation Level***": All the available computations are shown here. Computations are presented under the specimen they belong to. Specimens that doesn't have computations are not **shown** here.

- ***"Signal Level"***: All the available signals are presented in this level. The Signals are connected to the experiment or computation they belong to.

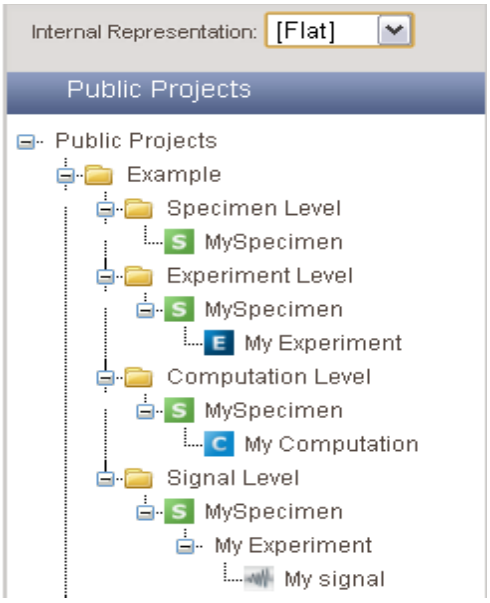


Figure 8: Flat Internal Representation

As it is shown in Figure 8, the ***"Layer"*** option is an exact representation of the Exchange Data Format therefore underlying levels are represented in a hierarchical way.

- ***"Specimen Level"***: The specimens that a project contains are visible under the specimen level. Expanding a specimen, the "Experiment Level" and "Computation Level" are available. Further expanding the "Experiment" or "Computation" level the "Signal" are presented.
- ***"Experiment Level***": The experiments that are contained into each specimen are visible in this level.
- ***"Computation Level***": The computations are presented under the specimen they belong to.
- ***"Signal Level"***: Expanding a computation or an experiment all the available signals that belong to are presented in this level. Signals like the experiments and computations are connected to the experiment or computation they belong to.

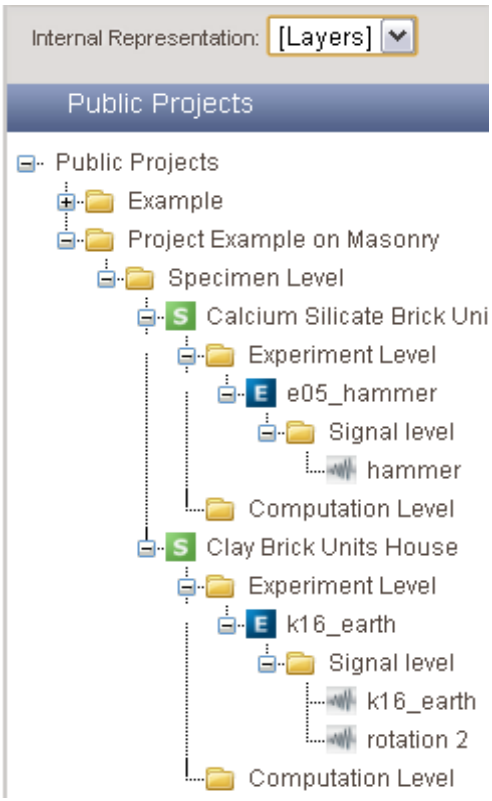


Figure 9: Layers Internal Representation

1.1.5 Project General Info Tab

Whenever a node from the tree structure is clicked the right pane is loading the following information:

"Project General Info": General information regarding the project is displayed here which includes:

- Project Info: project start date, project end date, sponsor of the project, acronym of the project and a small description about the project
- Investigators info: investigator name, investigator role, institution acronym, institution name
- Infrastructure info: location name and resource name

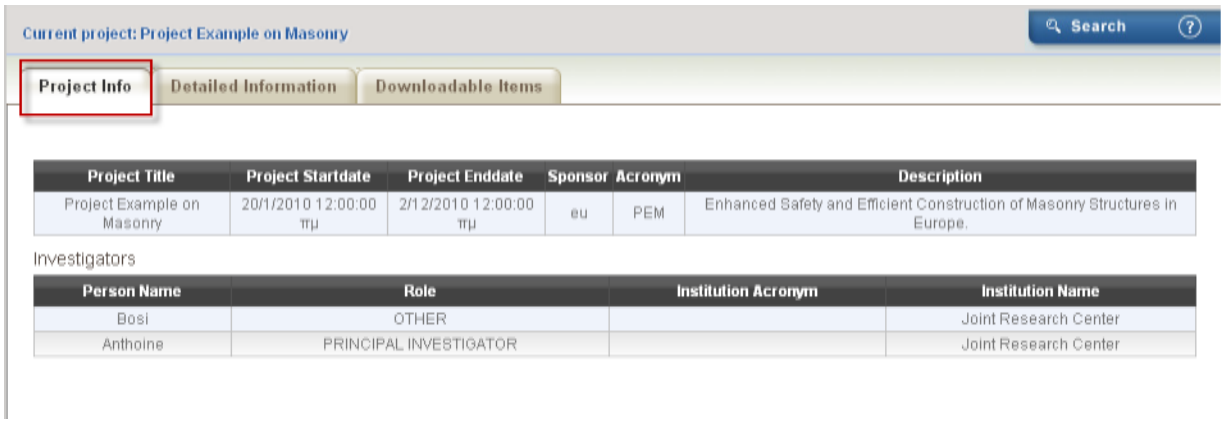


Figure 10: Project Information Tab

1.1.6 Project Detailed Info Tab

***"Detailed Information"*:** Detailed information tab provides information about the node that has been clicked by a user on the tree view. Information is presented in alias with the Exchange Data Format levels:

Project Level

- Project general data: Project Title, Project Acronym, Project Sponsor, Project Main Focus, Project Summary, Project Start Date, Project End date, Project Status
- Project Investigator
- Project Infrastructure
- Project Documents

Specimen Level

The information included in the specimen level, as depicted in Figure 9, is related to the following:

- Specimen data
- Structural elements
- Structural element material
- Material nominal properties
- Material actual properties

- Specimen documents
- Specimen images
- Scaling

Figure 11: The Specimen Level

Computation Level

The information included in the computation level provides information related to:

- General computation data
- Computation agents
- Computation document
- Computation images
- Detailed loading characteristics (DLCH)
- Original loading signal (OLS)
- Mesh model
- Mesh model images
- Computer system and software

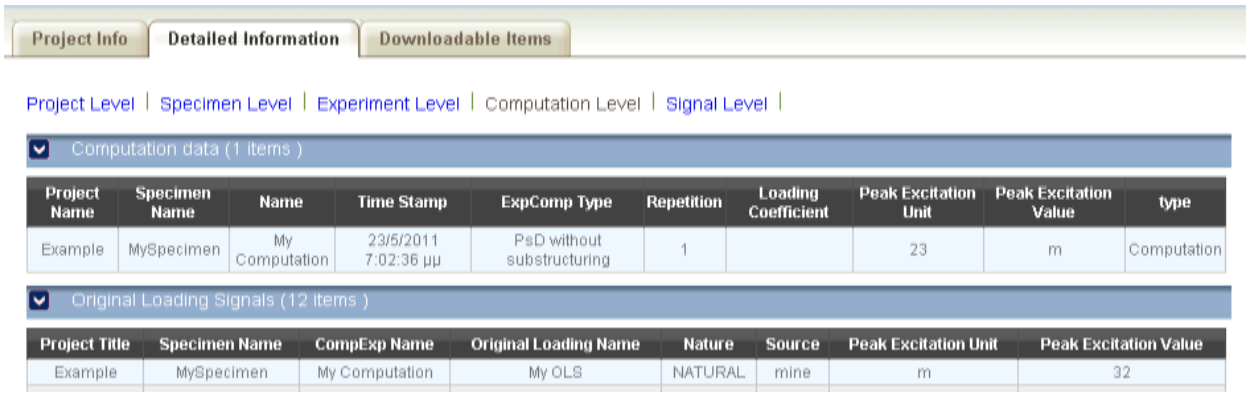


Figure 12: The Computation Level

Experiment Level

The experiment level provides the information, as depicted in figure 11, which is related to:

- General experiment data
- Experiment agents
- Experiment document
- Experiment images
- Experiment video
- Detailed Loading Characteristics (DLCH)
- Original Loading Signal (OLS)

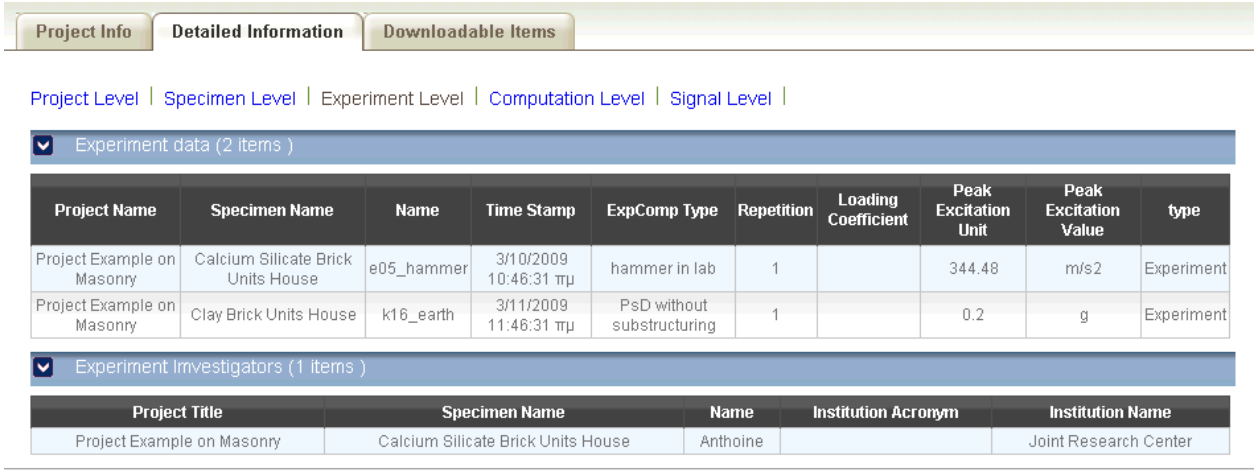


Figure 13: The Experiment Level

Signal Level

The signal level provides the information that is related to attributes, physical and type attributes of the signal as depicted in figure 12.

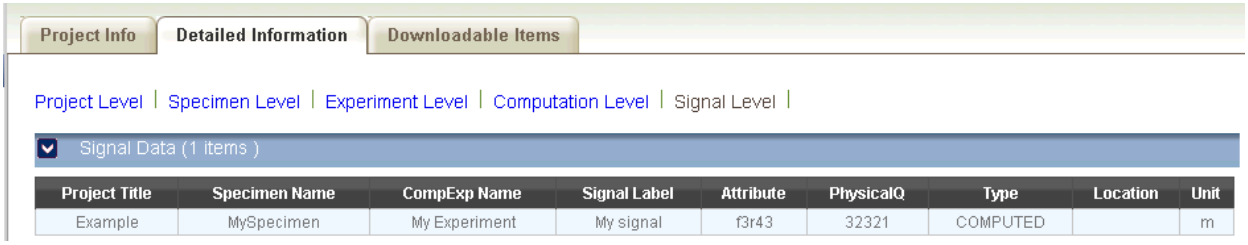


Figure 14: The Signal Level

Clicking on a item under whichever level the information related to this item are highlighted.

Figure 15: The Signal Level

1.1.7 Project Download Info Tab

"Downloadable Items": All the downloadable items of a project are available in this section. This tab like the "general project info" tab is showing the same information as long as nodes clicked are within the same projects.

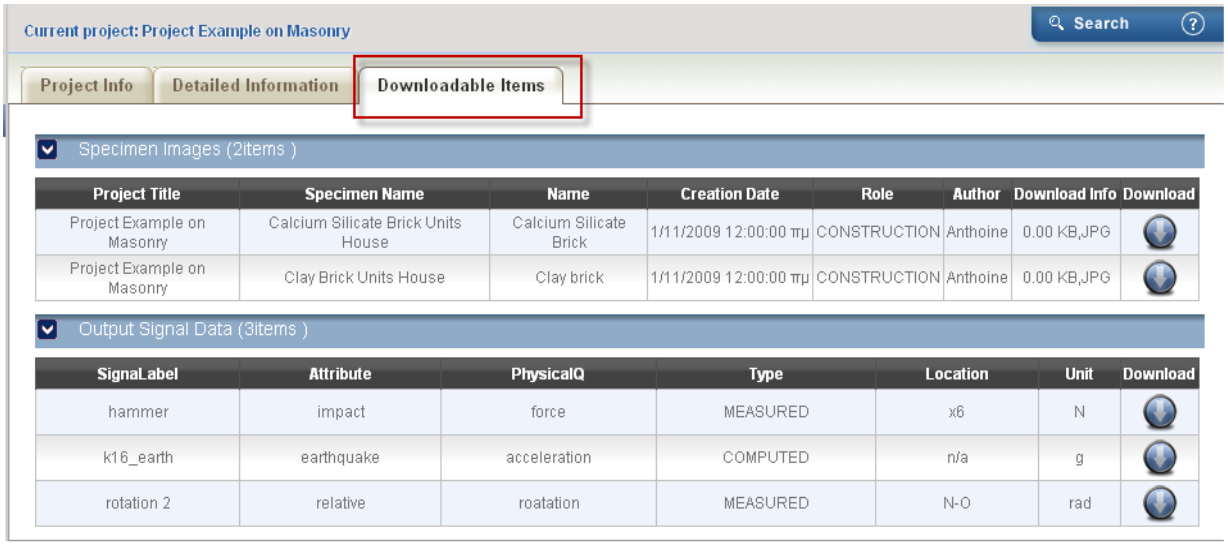


Figure 16: Download Tab

Files are grouped by according to the category they belong to

- Project Documents
- Specimen Documents
- Specimen Images
- Mesh Model Images
- Mesh Model Documents
- Experiment Images
- Experiment Documents
- Computation Documents
- Experiment Video
- Signals
- Detailed loading characteristics
- Original loading signals
- Signal

1.1.8 Terms and Conditions of Using Downloadable Items

Clicking on the download icon the "***Term and Conditions***" page is displayed. A user must accept the term and conditions before the download process begins. The text on the "Term and Conditions" page includes the following:

"*By using proprietary experimental data, supporting documentation or any other information (hereinafter the "Data") provided to you by a body, institute or laboratory within the project "SEISMIC ENGINEERING RESEARCH INFRASTRUCTURES FOR EUROPEAN SYNERGIES" (hereinafter "SERIES"), you agree to be bound by the following terms and conditions, and any policies or amendments thereto that may be subsequently introduced.*

All intellectual property rights in the data including, but not limited to, copyright and database rights are vested in their respective right holders (hereinafter the "Providers"). You are authorised - on a non-exclusive basis - to access, extract, reproduce, store, create derivative works and publish the Data on all media without alteration and subject to the provision of the following acknowledgment and disclaimer in all publications containing the Data:

(Acknowledgment) "The authors would like to thank the data providers and the SERIES Project (funded by the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 227887) for giving access to the Data."

(Disclaimer) "The views expressed herein are those of the author(s) and do not necessarily reflect the official position or interpretation of the data providers. All rights in the data are the property of the respective owners."

THE DATA IS PROVIDED TO THE HIGHEST POSSIBLE QUALITY AVAILABLE ACCORDING TO THE BEST PRACTICE AVAILABLE AT THE TIME OF ITS GENERATION. HOWEVER, YOU EXPRESSLY AGREE THAT THE USE OF THE DATA IS AT YOUR OWN RISK. TO THE MAXIMUM PERMITTED BY LAW, THE PROVIDERS EXPRESSLY DISCLAIM ALL WARRANTIES AND CONDITIONS OF ANY KIND, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE USE, QUALITY AND SUITABILITY OF THE DATA REMAINS WITH YOU. THE PROVIDERS WILL NOT BE LIABLE FOR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT OR INDIRECT DAMAGES INCLUDING, BUT NOT LIMITED TO, THE LOSS OF DATA, LOSS OF PROFITS, OR ANY OTHER FINANCIAL LOSS ARISING FROM THE USE OF THE DATA EVEN IF THE POSSIBILITY OF SUCH DAMAGES WERE FORESEEN, FORESEEABLE OR KNOWN BY THE PROVIDERS OR IF THE PROVIDERS WERE ADVISED OF SUCH RISK IN ADVANCE.

ANY REPRODUCTION OR DUPLICATION OF ALL OR ANY PART OF THE SERIES DATABASE IS PROHIBITED. ALL RIGHTS RESERVED."

Figure 17: Downloaded Icon

Search Functionality

1.1.9 Search criteria composition

The search functionality of the Data Access Portal is a structured keyword-based search. Keywords are separated according to the level that they are belonging to. Representative users are able to select any of the desired keywords from each category and click on the search button. The creation of complex queries is also supported by allowing a user to make multiple selections, as shown in Figure18.

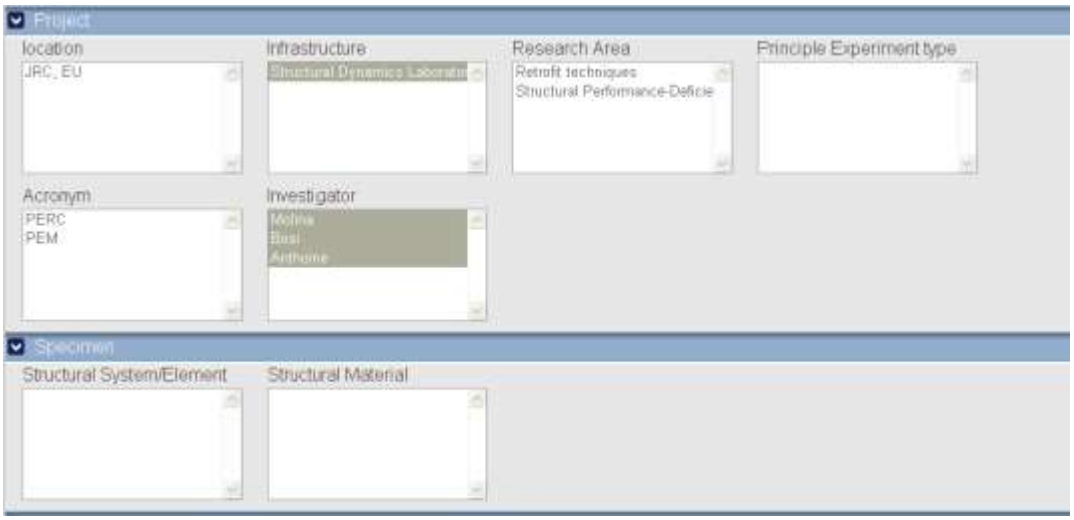


Figure 18: The Search Functionality

1.1.10 Search results presentation

The search results are presented in a structured approach as it can be seen in Figure 19, embrancing the description of each project and direct links to the download and the detailed project description web pages. This way a brief overview is presented for each project providing the opportunity to a user to navigate directly to the download page.

Figure 19: The Search Results

Furthermore, the search results are also displayed on a tree control on the left panel of the web page in which, the EDF levels that contain the search criteria are marked with red.

Figure 20: The Search Results View on a Tree Control

Privacy Options Related to Published Projects

In the frame of SERIES two district types of projects are supported: a) public and b) partner projects. These supported types are distinguished based on the privacy level they utilize. The

public projects are available to any visitor of the Data Access Portal whereas the Partner projects are available only to the member of the SERIES consortium.

Initially the Data Access Portal presents the public projects and not the partners projects which privacy status is defined from the laboratory the project belongs to. Only when a visitor uses the partner login feature of the Data Access Portal he will be authorized to access the partner projects. Partners of SERIES can use the credentials that already have from the main portal of the SERIES.

1.1.11 Partner Login

In Figure 21, the partner login process is presented, which entails the authentication of a user based on its credentials (i.e. username and password).

Figure 21: Partner Login

If a user enters wrong credentials he won't be able to access the partners but a ***"Partner not found"*** message will be displayed upper right corner. Instead, when a user enters the correct credentials the following actions are happening:

- The name of the partner and the logout option will be displayed in the upper right corner
- All the partner projects are displayed on the left pane under a new Tree View
- Public projects tree view is being hidden and the partner project is being highlighted (see as well Figure 23)

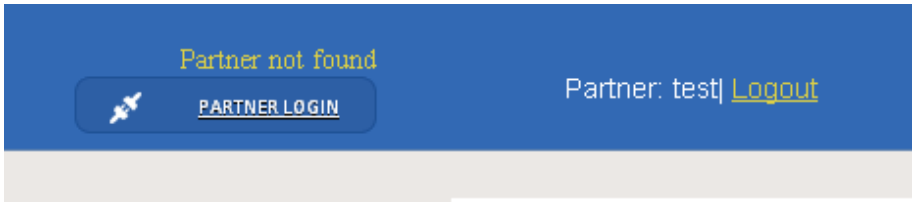


Figure 22: Successful and Unsuccessful Login

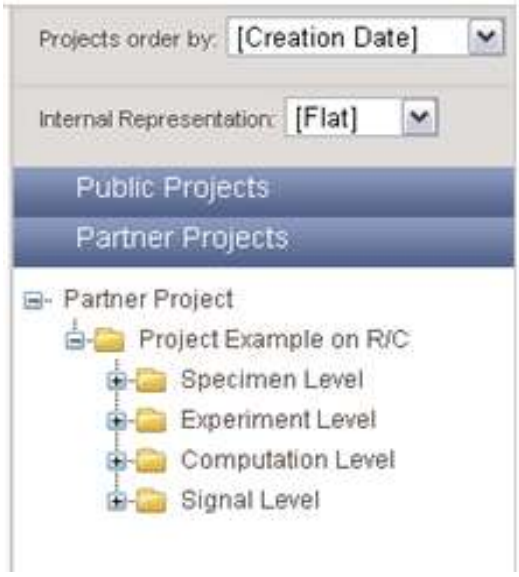


Figure 23: Partner Project Tree View

Public projects tree view can become visible again clicking on their title as it is shown in Figure 24

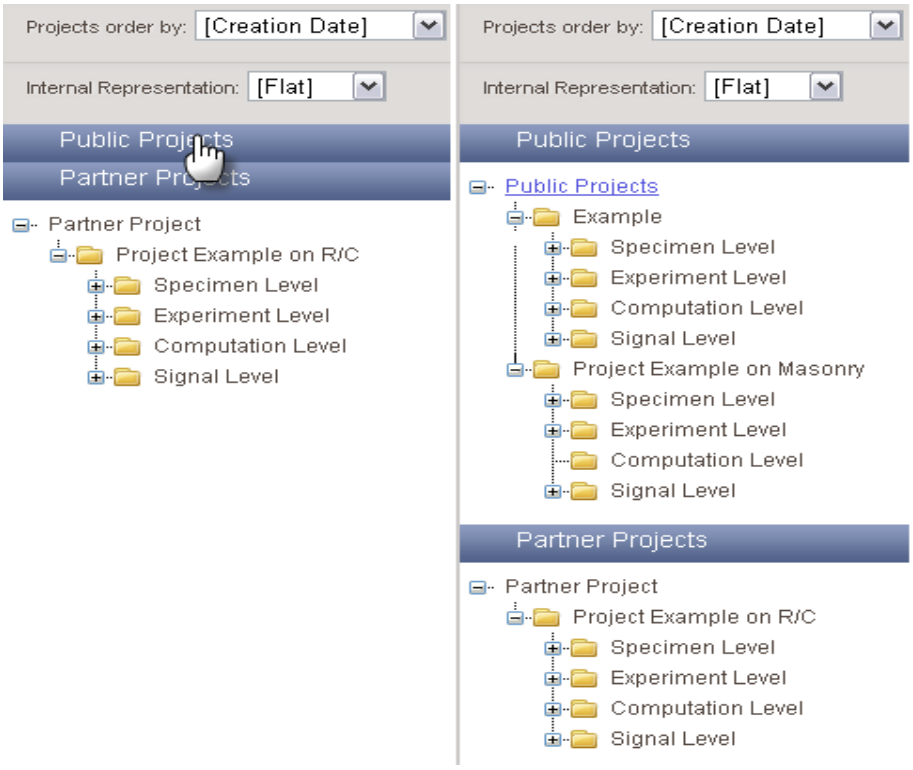


Figure 24: Tree View Expansion

2 Documentation of the SERIES Architecture

General Architecture

2.1.1 Constituents

There are two main parts involved in the architecture of a common repository:

• **A Central Site**, which holds the Virtual Database and the DAP (*Data Access Portal*). Physically, the SERIES central site is at *University of Patras* (Greece).

• **Laboratories, nodes or partners**, which have the experiment results in a local repository. Physically, the nodes are distributed around Europe (UK, Italy, France, Greece, Portugal, etc).

2.1.2 Client-Server architecture

The typical distributed architecture uses a Client-Server model. In this model, one computer is the server and provides services to the rest of the computers, known as clients. Server and clients are usually connected through a network, as *Figure 2* shows.

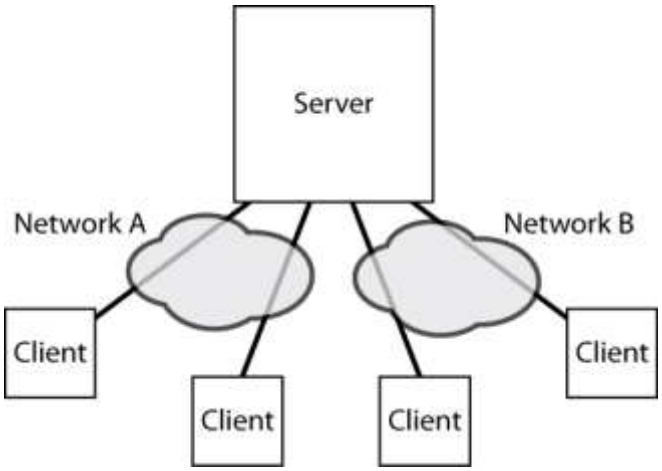


Figure 25: Client Server Architecture

In this model, the main load falls on the server as it has to deal with many clients. Some effort has been made to remove some load from the server side, resulting in client-side technologies.

A well-known example of a C-S architecture is the Web, where a Web client (typically a Web browser) connects to a Web server as the *Figure 3* example shows.

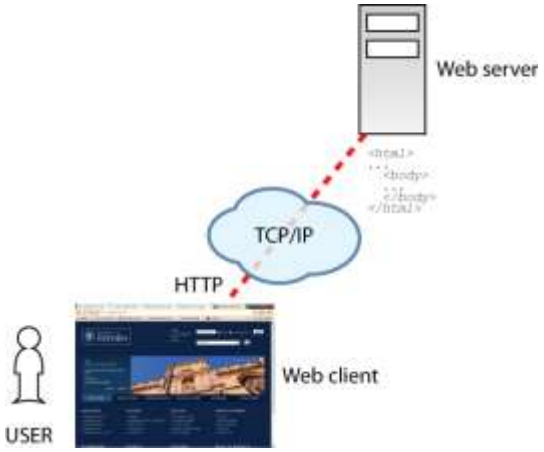


Figure 26: The Web as a example of a Client Server architecture

The SERIES Virtual Database operation suits perfectly in the C-S model. The C-S model can be applied in two different ways.

2.1.3 Global schema

Following the C-S model, S.ER.I.E.S. has a set of partners (or laboratories or nodes) connected to the SERIES Central Site. A schematic of the suggested system is presented in *Figure 6*.

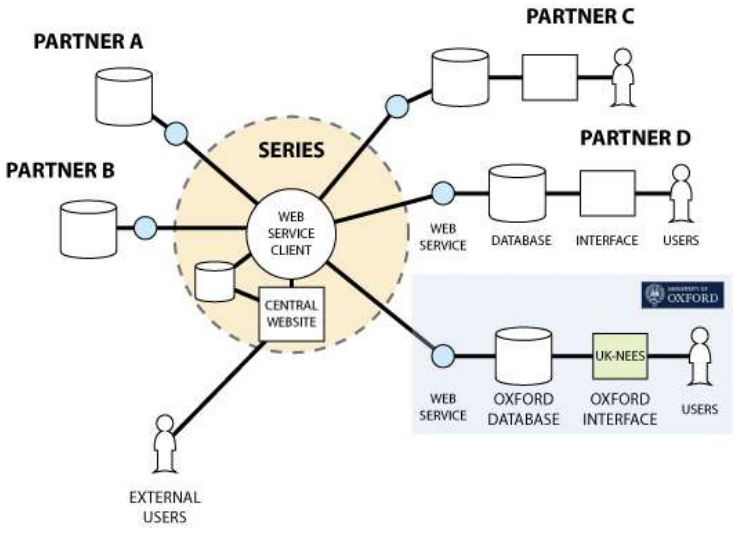


Figure 27: global architecture

According to this figure can be noted that:

• The Virtual Database **does not store** all the information from all the partners.

• The external user **uses only one interface**. Regardless of users getting some information directly from a specific partner, they must only use one interface: the Data Access Portal (DAP). The reasons for that are that we should homogenize the procedure to access information, nobody wants to learn how to use several interfaces and each partner is not forced to create its own interface.

• No direct **partner-to-partner communication** is provided, although this could change in future (new laboratories-collaboration possibility).

• The Central Virtual Database is **automatically updated**, without any need for human interaction. The possibility of requesting an update could exist, but it is not be the only way.

There exist three main layers on the global schema, as presented in *Figure 7*:

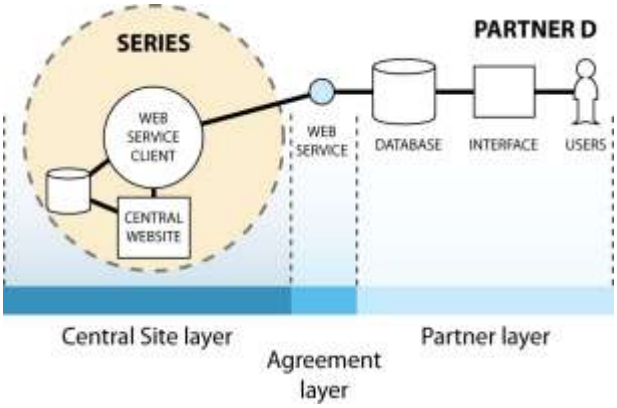


Figure 28: Architecture's three layers

- **Central Site layer**, with the Virtual Database and the DAP.
- **Agreement layer**, which every partner should conform to.
- **Partner layer**, with the various systems and repositories of each partner.

Each of these layers will be discussed in later sections of this document.

2.1.4 Global technical solution

As seen in *Figure 6*, a key role is played by **Web Services**.

Within distributed systems, such as the one we find in SERIES, **SOA** (*Service Oriented Architecture*) is an architectural paradigm that focuses in connecting heterogeneous systems under the control of different owners. This methodology allows interoperability between different systems.

One of the ways to implement SOA is by means of Web Services. Thus, the recommended method to communicate partners and Central Site is **via Web Services.** In order to implement WS, the **SOAP** (*Simple Object Access Protocol*) alternative was chosen. Some other systems, such as *REST* or a proprietary one, would also work well but the SOAP choice is mainly due to its standarization and some of its complementary possible specifications (like *WS-Security*). This solution is *XML based*, which complies with the SERIES DoW (*Description of Work*).

Of course, SOAP has advantages and disadvantages. While can save the reinvention of the wheel, in some situations it can be harder to implement than other solutions.

In any case, from a technical point of view it is recommended to use *wrapped document style* and *document/literal binding* for the SOAP approach to maintain maximum compatibility

The Central Site layer

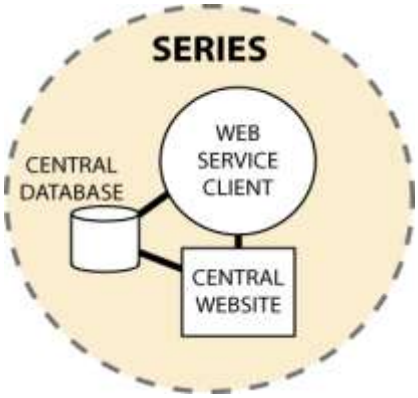


Figure 29: Central site layer

The main purpose of the Central Site layer is to provide access to a repository implemented as a "Virtual Database".

The structure can be divided in three main parts, as seen in *Figure 8*:

• **Central Database**, is the component that best resembles a Virtual Database. It stores information and allows first level data consultation.

• **Web Service client**, which connects with all partners' WS in order to obtain the data for the Virtual Database.

• **Central Website** or **DAP**, an interface to consult the Virtual Database.

Notice that the Virtual Database does not only correspond to the Central Database. The Virtual Database concept is wider, and includes the referenced information stored locally by each partner

2.1.5 Central Database

In the implemented solution, the Central Database can be considered like a cache of all partners' repositories. The Central Database is storing cardinal information and all searchable data, if possible.

Allowing this data storage as a cache, an external user can access the first-level information very quickly. Without cardinal information being cached by the central database, the central site would need to connect to partner repositories in order to attain this information, and the system would appear to be very sluggish to the user.

Thus, the Central Database is storing information about the project, such as the project title, project acronym, etc. No large file is stored in this database, but metadata about large files are stored. Following this, database stores: document's title, size or format but not the document itself (see *Figure 9*).

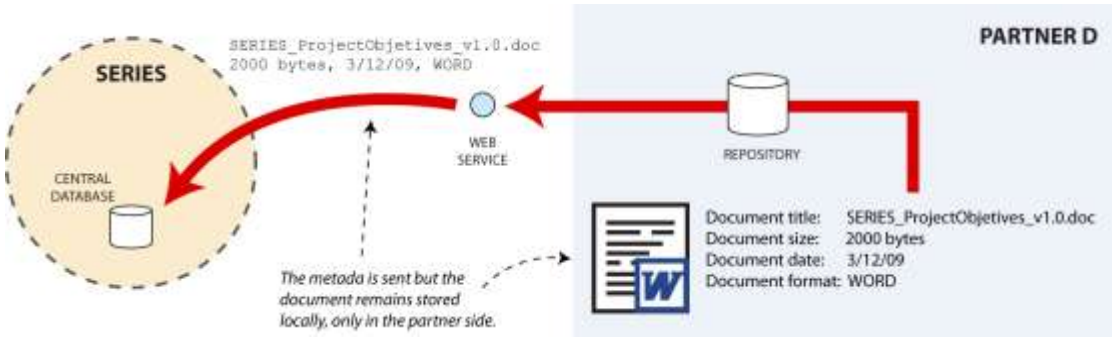


Figure 30: Large files storage

This scheme is comparable to a search engine like *Google*. In some way, Google acts as a cache of other Websites, retrieving some specific information (but not all!) and storing it locally. When users query Google, they do not have to wait too long to access the information (or at least, not as long as having to wait for Google to consult all the indexed Websites at that moment).

A structure has been created within the database to store partner information such as for example:

• *Partner name*.

• *Location of the service*, so it is not "hard-coded" in the WS client code.

• *Administrative contact details*.

• *Technical contact details*, in case some technical problem happens.

• *Security aspects*.

• *Updating parameters.*

• *Date of joining* in SERIES.

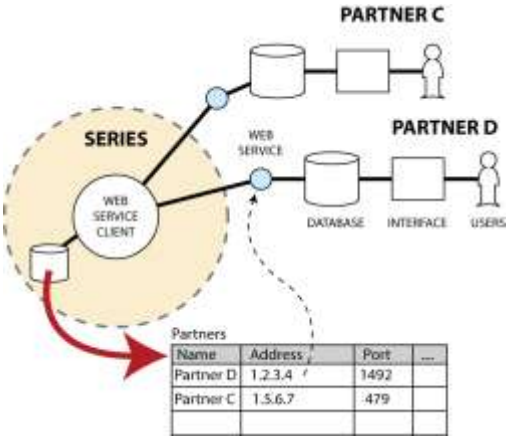


Figure 31: PARTNERS table

The most relevant information is the location of the service, since that will allow connection to the partner Web Service. This document will reference this structure as the table "PARTNERS" (*Figure 10*).

2.1.6 Web Service client

The Web Service in the Central Site is in charge of connecting with all partners in order to get the information that feeds the Virtual Database. It translates all the received information, coming in a common agreed format, to the data for the Central Database.

As long as partners implement a Web Service consumer that complies with the WS specification, the platform and programming language that are employed are of no consequence. One of the benefits of Web Services is this freedom to choose.

The WS client is the core in the Central Site. After all, it is the only "*live*" entity in the Central Site: the Central Database is a sleepy repository and the Central Website is user-driven. In other words, the database is just a program with no initiative, it just answers user's requests. It sleeps in the background and only triggers when the user needs some information. The Central Website is also a "*dead program*", because it is the user who, by using a browser, goes from one page to

another. It creates petitions, but just because the user requests actions. On the other hand, the WS can suddenly wake up to connect to all partners' databases to check their status, see if it can reach a partner DB and then send an email automatically to their administrators if there is a problem to warn them of the situation.

The **WS client-Central Database** communication runs in one way to get location information about the partners from the Database (by using the PARTNERS table) and, in the other, to store the repository data that comes from the partners.

The **Central Website-WS client** communication exists to satisfy external user requests, like, for example, advance searching.

2.1.7 Central Website

The Data Access Portal provides a unique access point for external users to consult the Virtual Database and access its information. From a user point of view, there should only exist one interface, one single Website, and all the information received or downloaded should seem as though it came from the same place, even if it actually comes or is downloaded from different sources.

An important aspect of the Website is to comply with Web Standards, such as *XHTML*, *CSS* and *Accessibility*. Those standards should be properly validated. *Usability* should also be taken into consideration and tested.

Clearly, the Data Access Portal has direct access to the Central Database, and has been developed considering not only the access to the Virtual Database. In that way, for instance, the Data Access Portal is used not only as a *Repository Portal* but as a *Testing Portal*.

As the only and common external access point, it could compromise the legitimate access to the partners, so **security** in here is a priority.

2.1.8 Searching

Searching is one of the key basic features in the Website. It makes possible filtering results for the external user, finding information that he or she is interested in.

Search in the Central Site is done, without connecting to any partner. The reason for that is performance. It is desirable for the search feature to be fast. If the Central Site has to connect to one or several partners to perform a search, then the system will respond slowly.
Searching affects several parts within the Central Site, since it requires a user interface in the Website, some logic that performs or orders the searching and a search-suitable design of the database. The user interface plays a key role for the search feature to succeed.

2.1.9 Adding a new partner

The system is flexible enough to add a new node or partner easily. From a system point of view, adding a new partner involves several steps:

• Creation and sharing of all security elements (for example personal certificates creation).

• Addition of new information in the table "PARTNERS", storing the address of the partner's service.

• Test that the new partner's Web Service is working correctly.

2.1.10 Security

Security is implemented in every part of the system. Nevertheless, it is very important the Central Site implements some security measures, useful for all the partners.
For example, most of the input data is coming from the Central Website (Data Access Portal). Regarding this input, the Central Web Service is communicating with partners' Web Services. If the Central Site does not filter the input received from the Central Website, it can propagate a security risk to the partners' Web Sites as *Figure 14* shows (imagine, for instance, a typical *SQL injection* attack). Of course the partners' Web Services should implement security checks, but a centralized first-level security within the Central Site is preferable and avoids many headaches.

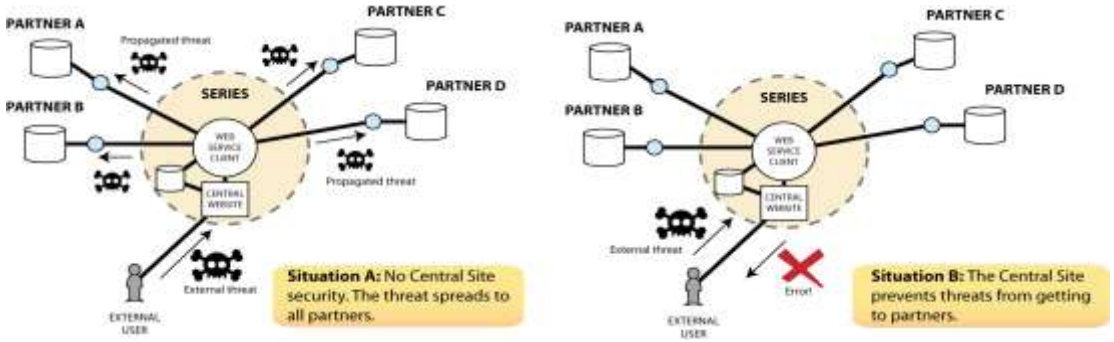


Figure 32: Security

The Agreement layer

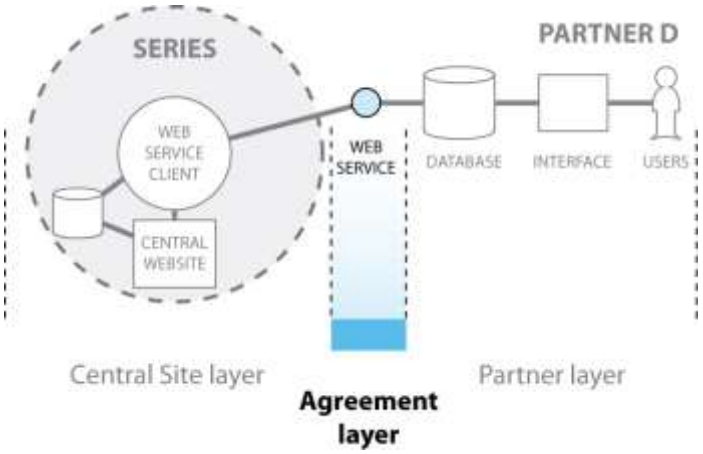


Figure 33: Agreement layer

The agreement layer specifies the **contract** between the Central Site and partners, that is required for successful communication with each other in a uniform, standard way.

As an analogy, this layer is like the language that allows the partners to speak and understand one another. Every partner can speak its own language or dialect at home but when communicate with the outside world, a common language must be adopted using defined grammatical rules.

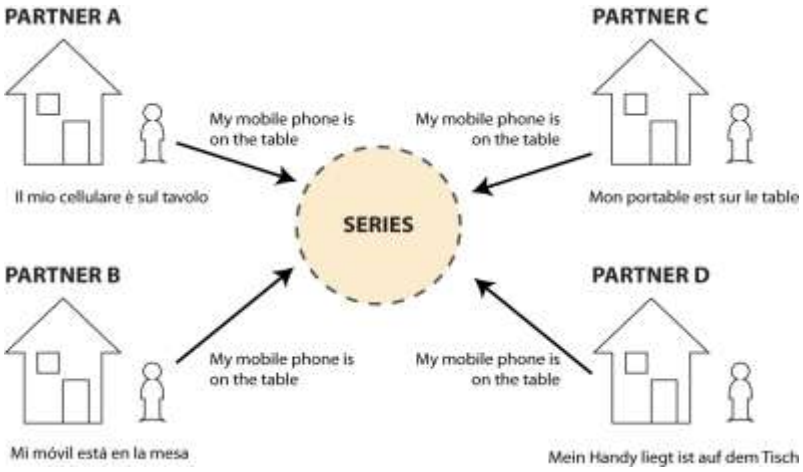


Figure 34: Language analogy

The "contract" in the agreement layer specifies these grammatical rules. Partners agreeing to and complying with the contract will be understood by the Central Site. Technically speaking, the contract defines the services provided, by means of:

• **Operations** that can be called.

• **Messages** to be exchanged for each operation.

• **Data types** of the attributes of the messages.

It is the responsibility of every partner to implement the operations defined in the contract and make sure that this implementation works properly according to the contract. Although a single contract is mentioned, actually there will be one contract for each partner (see *Figure 16*). These contracts will define the services that the partner provides. The initial contract to be implemented by the partners will be referred as the "**common contract**" in comparison to the common contract "**copy**" that actually exists in every node. Partner contracts must be a mirror "copy" of the common contract. This common contract should be fully implemented by each partner, although they could also extend it with new services.

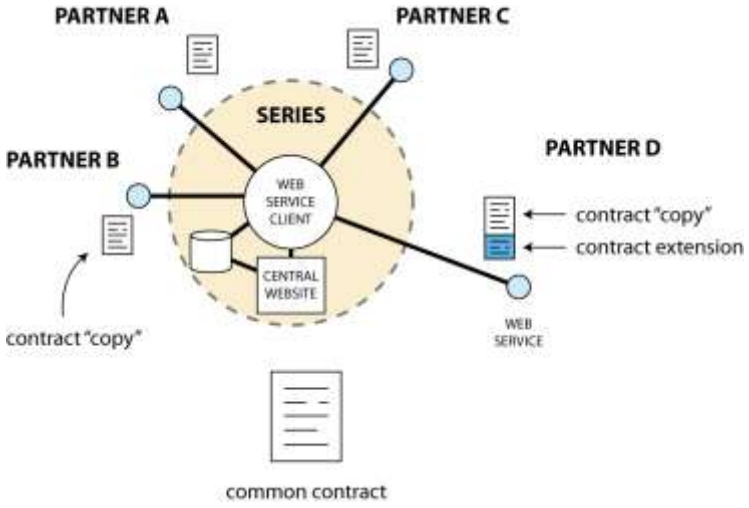


Figure 35: Contracts in the nodes

In the suggested solution, contracts are technically implemented by means of **WSDL** (*Web Service Description Language*).

2.1.11 Services

The defined system is **service oriented**. This means that the way to communicate with the partners is by using the services they provide. The aforementioned contract is used to ensure that services are called correctly.

The *Specification of the Common Database Data Format* is finished and is described in the appendix.

2.1.12 Security and Privacy

Security should be conscientiously implemented on the Web Service. Albeit the Central Site implements a first-level security, each partner has the responsibility of ensuring the security of their own Web Service.

The Central Site - partner communication should be reliable, but it is recommended security checks are performed as if it were not. Also, while the external user must always communicate with the partners through the Data Access Portal, nothing prevents potential malicious users from trying to access the laboratory Web Service directly.

To achieve safety, elements such as certificates, among other measures, are being used.

Even though the external user connects through the Data Access Portal, and the security certificates create a legitimate valid network communication, it does not mean his or her intentions are honest. For example, the Web Service should make sure the external user is only able to read from the repository, and not to write to it, regardless of he/she is using the right communication channels.

In some cases, the external user is connected directly to the partner. For instance, when the external user requests a very large file like a video, it is more efficient to do a direct download via user-node instead of sending all the data through the Central Site. For example, a session can be agreed between Central Site and partner to allow external users to connect to a partner directly, for a period of time and to obtain a specific file. The process of setting a session is transparent for the user.

Another concept related to security is the privacy of the repositories. Partners can store on their repositories information that should not be shared or simply is not of the interest of SERIES.

Partners worried about privacy of their own repositories should notice that the Central Site can not access to information that is not shared by the Web Service. Thus the Web Service on the local node decides which data is sent and has the capacity of filtering and discriminating the shared information.

In a more technical view, the partners do not need to use different repositories -one which shares data with SERIES and one that does not. Having two different repositories (say databases) is an inefficient option. The recommended way for a partner to implement the SERIES solution is by using a single repository and letting the Web Service provide the relevant information (unless some other special reason or local restriction prevents this). Of course this solution implies that the local repository must be modified in order to add specific data about privacy, specifying if the data is “shareable” or not (but it can save some bad headaches and maintenance nightmares on the partner side).

The Partner layer

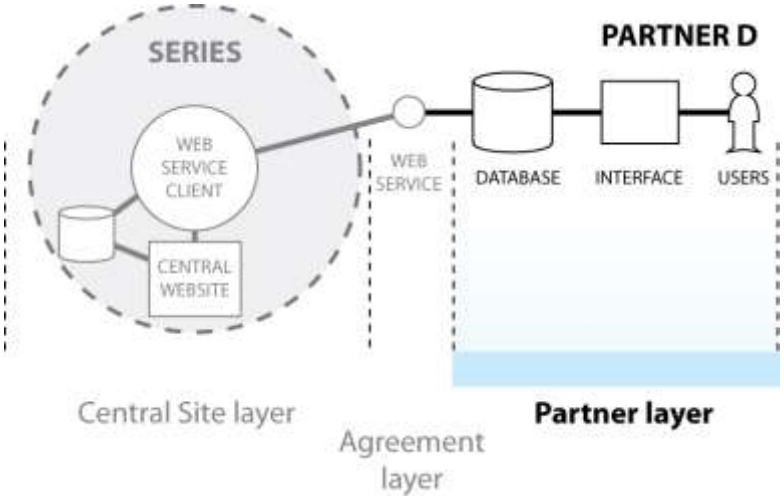


Figure 36: Partner layer

Of the three layers, the partner layer has the highest level of freedom. Some SERIES' partners will already have a defined structure. Some partners will use *Windows*, some others *Linux* or any other operating system. Some partners might have an *Oracle* database and some others might use *Postgre*, *MySQL* or *SQLServer*. This presents no problem. Web Services were created to deal with such heterogeneous maps. SOAP was designed to connect machines regardless of the operating system, CPU or application.

The Central Site could develop its Web Service in a programming language like *Java* whereas a partner could do it in *Perl* or *Python*, and the communication between both will work. This provides a huge flexibility and a very wide range of possible configurations on the partner's side.

2.1.13 Repository

While the partner repositories can be implemented in different ways, the recommended option is to use a database. If this option is taken, many decisions are required: database type, database engine selection, storage of big files in the database or in the local file system, etc.

While not recommended, it does not matter if the information is stored in many different sources, as long as all the relevant data is correctly collected by the Web Service (*Figure 20*).

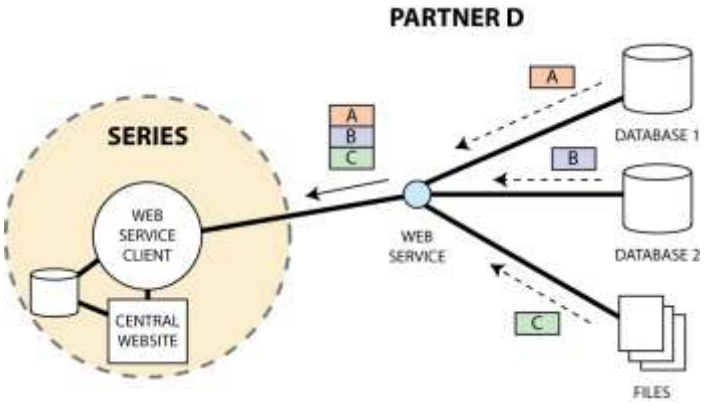


Figure 37: Different sources repository

Regardless of how the repository is designed and implemented, it is very important that every object in the repository has a **unique ID** within its scope. This ID should not vary or otherwise some mechanism is required to translate between the old and the new. The reason for this is that the Central Site may not recognise modifications and reference out-of-data IDs.

As the Client Site can also work as a service, it is suggested to store the Central Site location information (for example, *IP address* and other service provider data) within the local repository. This prevents one from having to hardcode the local programs or WS consumer code.

If a partner has no repository structure at all, it is suggested a new database is implemented with SERIES functionality. For existing repositories, two main options can be considered:

• Keep the current repository and perform the "translation" for SERIES as close as possible, via the partner's Web Service.

• Develop a new repository with SERIES functionality, and perform a migration from the old repository to the new one.

Depending on the current repository format, migration could be a very hard task. A perfect migration might not be possible: SERIES might require data that does not exist in the current repository or that is in a different format. In that case, migration rules or time to adapt the existing data must be considered. Here, the range of possibilities is quite wide and the solution will depend on the specific design of the partner's repository.

2.1.14 Interface

The partner's interface is the main connection between repository and users. By using the interface, users can store their experiment data in an easy way. Otherwise, users should deal with the repository software directly (usually a database), which is usually not user friendly. Only an administrator should have direct access to a database.

If a partner has a repository with no interface, it is recommended that they create one. A Web-based interface might be the most flexible solution. It also simplifies the process, since users do not need to install any extra software to use the interface.

3 Conclusions

This deliverable presents the 2nd version of the SERIES Data Access Portal (DAP) (http://www.dap.series.upatras.gr/). The DAP is developed by following a user centred design approach (UCD) which is an iterative process of requirements engineering, designing, developing and evaluating of interactive systems. As a consequence the 2nd version of the Data Access Portal has been based on the preliminary version which has been described in deliverable D.2.1. The preliminary version of the DAP has been reviewed by partners of the SERIES consortium and their feedback where taken into consideration during the subsequent development cycle of the DAP which, among others, entailed enhancements of existed functionalities and development of new functionalities which embrace the following:

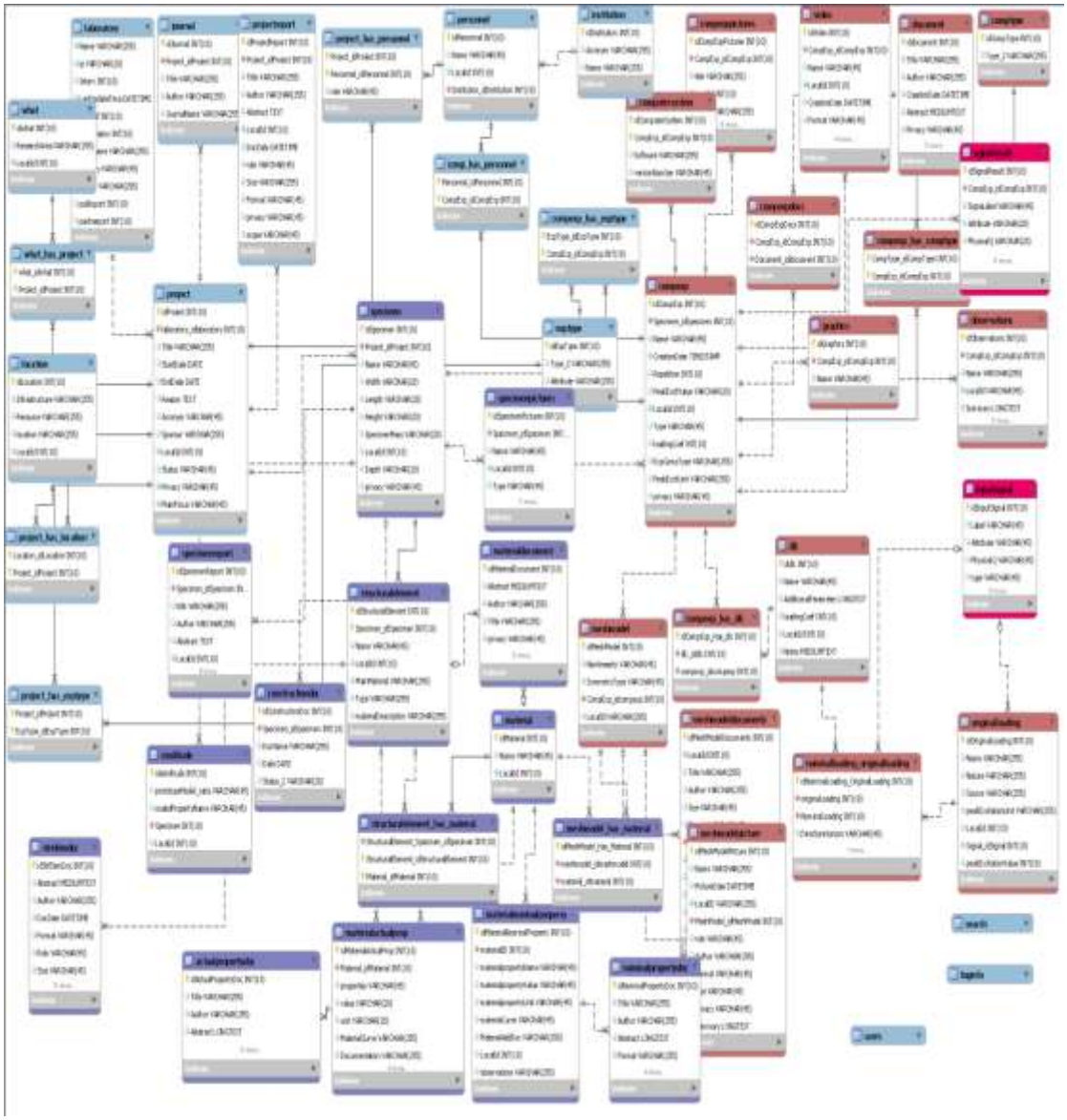
- A new layout of the DAP has been designed and developed which is in alliance with the SERIES Web Portal (www.series.upatras.gr) aiming to provide a similar user experience in terms of visual design, information architecture and interaction behaviour
- A new access control mechanism has been implemented with two levels of permissions related to public or private published projects. Projects which are published with a public flag can be accessed by any visitor of the DAP whereas projects that are published with a private flag can be accessed only by certain members of the SERIES consortium
- A new information presentation functionality of published projects has been implemented embracing two complementary information presentation approaches taken intro consideration the hierarchical information presentation of the Exchange Data Format
- A new main page of the DAP has been designed entailing a quick overview of the available functionalities, including list of recent published projects, overview of the Exchange Data Format and link to the user manual of the DAP

Additionally the web services architecture has been finalized after the successful implementation of the 3 proof of concepts. In the last period of the SERIES project, these mechanisms that were implemented will now be used but the SERIES partners to give access to there own data (at least the data pertaining to the SERIES/transnational access activities) and by SERIES users to obtain them. A report concerning these activities will be provided at the end of the project.

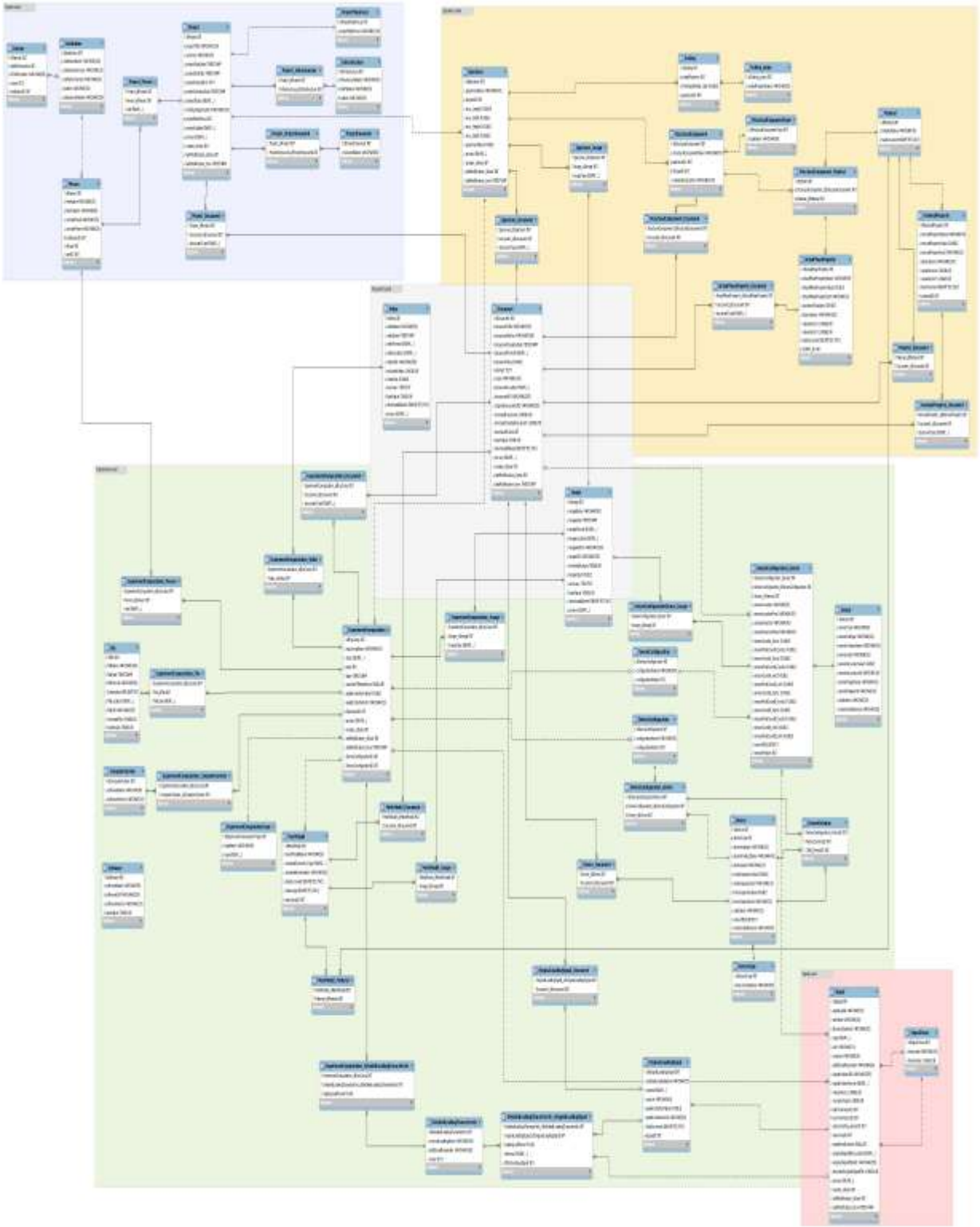
4 Apendix A: DAP Database Design and Implementation

Entity Relation Diagram

4.1.1 Central Site Entity Relation Diagram



4.1.2 Laboratory Entity Relation Diagram



Database Tables

4.1.3 Central Site Database Tables

The most important database tables are documented in this appendix

Database: seriesserver

Database: newseriesserver — Properties

Name:	seriesserver
Default Character Set:	utf8
Default Character Collation:	utf8_general_ci

Tables

Name	Engine	Comment
comp_has_personnel	InnoDB	
compexp	InnoDB	
compexp_has_comptype	InnoDB	
compexp_has_exptype	InnoDB	
comptype	InnoDB	
computersystem	InnoDB	
dlc	InnoDB	
document	InnoDB	
exptype	InnoDB	
graphics	InnoDB	
signal	InnoDB	
institution	InnoDB	
laboratory	InnoDB	
location	InnoDB	

material	InnoDB
meshmodel	InnoDB
personnel	InnoDB
project	InnoDB
project_has_exptype	InnoDB
project_has_location	InnoDB
project_has_personnel	InnoDB
signalresult	InnoDB
similitude	InnoDB
specimen	InnoDB
structuralelement	InnoDB
structuralelement_has_material	InnoDB
video	InnoDB

Tables

Table: comp_has_personnel

Table: comp_has_personnel Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	5461
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
Personnel_idPersonnel	int(10) unsigned	☐	☐		
CompExp_idCompExp	int(10) unsigned	☐	☐		

Table: compexp

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	2730
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	838
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idCompExp	int(10) unsigned	☐	☑		
Specimen_idSpecimen	int(10) unsigned	☐	☐		
Name	varchar(45)	☑	☐		
CreationDate	timestamp	☐	☐	CURRENT_TIMESTAMP	
Repetition	int(10) unsigned	☑	☐		
PeakExcitValue	varchar(20)	☑	☐		
LocalId	int(10) unsigned	☐	☐		
Type	varchar(45)	☑	☐		
loadingCoef	int(10) unsigned	☑	☐		
ExpCompType	varchar(255)	☑	☐		
PeakExcitUnit	varchar(255)	☑	☐		
privacy	varchar(45)	☑	☐		

Table: compexp_has_comptype

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
CompType_idCompType	int(10) unsigned	☐	☐		
CompExp_idCompExp	int(10) unsigned	☐	☐		

Table: compexp_has_exptype

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExpType_idExpType	int(10) unsigned	☐	☐		
CompExp_idCompExp	int(10) unsigned	☐	☐		

Table: comptype Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	1
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idCompType	int(10) unsigned	☐	☑		
Type	varchar(255)	☑	☐		

Table: computersystem — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	16384
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	101
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idComputerSystem	int(10) unsigned	☐	☑		
CompExp_idCompExp	int(10) unsigned	☐	☐		
Software	varchar(255)	☑	☐		
versionNumber	varchar(45)	☑	☐		

Table: dlc

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	910
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	343
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
iddlc	int(10) unsigned	☐	☑		
Name	varchar(45)	☑	☐		
AdditionalParameter	longtext	☑	☐		
loadingCoef	int(10) unsigned	☑	☐		
LocalId	int(10) unsigned	☑	☐		
Notes	mediumtext	☑	☐		

Table: document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	1
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
iddocument	int(10) unsigned	☐	☑		
Title	varchar(255)	☑	☐		
Author	varchar(255)	☑	☐		
CreationDate	datetime	☑	☐		
Abstract	mediumtext	☑	☐		
Privacy	varchar(45)	☑	☐		
Size	varchar(45)	☑	☐		
Format	varchar(45)	☑	☐		
scope	varchar(45)	☑	☐		
localID	int(10) unsigned	☐	☐		
role	varchar(255)	☑	☐		

Table: inputsignal

Table: signal

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	227
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	796
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSignal	int(10) unsigned	☐	☑		
Label	varchar(45)	☑	☐		
Attribute	varchar(45)	☑	☐		
PhysicalQ	varchar(45)	☑	☐		
type	varchar(45)	☑	☐		
Unit	varchar(45)	☑	☐		
Location	varchar(255)	☑	☐		
localid	int(10) unsigned	☐	☐		
repetition	int(10) unsigned	☑	☐		
privacy	varchar(45)	☑	☐		

Table: institution

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	8192
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	5
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idInstitution	int(10) unsigned	☐	☑		
Acronym	varchar(255)	☑	☐		
Name	varchar(255)	☑	☐		

Table: meshmodel

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	16384
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	126
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idMeshModel	int(10) unsigned	☐	☑		
Nonlinearity	varchar(45)	☑	☐		
SymmetryType	varchar(45)	☑	☐		
CompExp_idcompexp	int(10) unsigned	☐	☐		
LocalId	varchar(255)	☑	☐		

Table: personnel

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	2730
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	64
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idPersonnel	int(10) unsigned	☐	☑		
Name	varchar(45)	☑	☐		
LocalId	int(10) unsigned	☐	☐		
Institution_idInstitution	int(10) unsigned	☐	☐		

Table: project

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	3276
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	799
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idProject	int(10) unsigned	☐	☑		
laboratory_idlaboratory	int(10) unsigned	☐	☐		
Title	varchar(255)	☑	☐		
StartDate	date	☑	☐		
EndDate	date	☑	☐		
Reason	text	☑	☐		
Acronym	varchar(45)	☑	☐		
Sponsor	varchar(255)	☑	☐		
LocalId	int(10) unsigned	☐	☐		
Status	varchar(45)	☑	☐		
Privacy	varchar(45)	☐	☐		
MainFocus	varchar(45)	☑	☐		

DateCreated	date	☑	☐

Table: project_has_personnel

Table: project_has_personnel Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	4096
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
Project_idProject	int(10) unsigned	☐	☐		
Personnel_idPersonnel	int(10) unsigned	☐	☐		
role	varchar(45)	☑	☐		

Table: signalresult

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	2730
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	431
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSignalResult	int(10) unsigned	☐	☑		
CompExp_idCompExp	int(10) unsigned	☐	☐		
SignaLabel	varchar(45)	☑	☐		
Attribute	varchar(20)	☑	☐		
PhysicalQ	varchar(20)	☑	☐		
Type	varchar(20)	☑	☐		
Unit	varchar(20)	☑	☐		
Location	varchar(45)	☑	☐		
LocalId	int(10) unsigned	☐	☐		
privacy	varchar(45)	☑	☐		
repetition	int(10) unsigned	☑	☐		

Table: similitude

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	4096
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1324
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idsimilitude	int(10) unsigned	☐	☑		
prototypeModel_ratio	varchar(45)	☑	☐		
scaledPropertyName	varchar(45)	☑	☐		
Specimen	int(10) unsigned	☐	☐		
LocalId	int(10) unsigned	☑	☐		

Table: specimen Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	2340
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1392
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSpecimen	int(10) unsigned	☐	☑		
Project_idProject	int(10) unsigned	☐	☐		
Name	varchar(45)	☑	☐		
Width	varchar(20)	☑	☐		
Length	varchar(20)	☑	☐		
Height	varchar(20)	☑	☐		
SpecimenMass	varchar(20)	☑	☐		
LocalId	int(10) unsigned	☑	☐		
Depth	varchar(20)	☑	☐		
privacy	varchar(45)	☑	☐		

Table: structuralelement

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idStructuralElement	int(10) unsigned	☐	☑		
Specimen_idSpecimen	int(10) unsigned	☐	☐	0	
Name	varchar(45)	☑	☐		
LocalId	int(10) unsigned	☑	☐		
MainMaterial	varchar(255)	☑	☐		
Type	varchar(255)	☑	☐		
materialDescription	varchar(255)	☑	☐		

Table: video

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Avg row length:	8192
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	59
Create time:	11/22/2011
Update time:	
Check time:	
Collation:	utf8_general_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idVideo	int(10) unsigned	☐	☑		
CompExp_idCompExp	int(10) unsigned	☐	☐		
Name	varchar(45)	☑	☐		
LocalId	int(10) unsigned	☐	☐		
CreationDate	datetime	☑	☐		
Format	varchar(45)	☑	☐		
Size	varchar(45)	☑	☐		
Role	varchar(45)	☑	☐		
Summary	mediumtext	☑	☐		
privacy	varchar(45)	☑	☐		

4.1.4 Laboratory Database Tables

Database: seriesdb

Database: seriesdb **Properties**

Name:	seriesdb
Default Character Set:	latin1
Default Character Collation:	latin1_swedish_ci

Tables

Name	Engine	Comment
actualmeanproperty	InnoDB	Material/Struct Com effective values.
actualmeanproperty_document	InnoDB	
computersystem	InnoDB	Configuration of Computations
dbinfo	InnoDB	Information about the Database
detailedloadingcharacteristic	InnoDB	It collects all the information that characterizes the exp
detailedloadingcharacteristic_originalloadingsignal	InnoDB	
device	InnoDB	
device_document	InnoDB	
deviceconfiguration	InnoDB	
deviceconfiguration_device	InnoDB	
devicerelation	InnoDB	To link devices each other
devicetype	InnoDB	
document	InnoDB	Generic document or file.
experimentcomputation	InnoDB	
experimentcomputation_computersystem	InnoDB	
experimentcomputation_detailedloadingcharacteristic	InnoDB	

experimentcomputation_document	InnoDB	For Experiment's input files, Boundary conditions...
experimentcomputation_file	InnoDB	
experimentcomputation_image	InnoDB	For Boundary conditions
experimentcomputation_person	InnoDB	Test agents in an experiment.
experimentcomputation_video	InnoDB	
experimentcomputationtype	InnoDB	For the type of experiments
file	InnoDB	
image	InnoDB	Graphic image or photo.
infrastructure	InnoDB	Location and name of the resources used in the project.
institution	InnoDB	List with all the participants institutions available
material	InnoDB	Represent a material.
material_document	InnoDB	Additional documentation
meshmodel	InnoDB	
meshmodel_document	InnoDB	For Additional Documentation
meshmodel_image	InnoDB	For Undeformed shape picture
meshmodel_material	InnoDB	
nominalproperty	InnoDB	Properties for materials
nominalproperty_document	InnoDB	
originalloadingsignal	InnoDB	Original inputs time-histories and related info.
originalloadingsignal_document	InnoDB	
partner	InnoDB	
person	InnoDB	Human resources working in a Project.
project	InnoDB	

project_document	InnoDB	Reports in a Project.
project_infrastructure	InnoDB	Main infrastructures and facilities used for the project.
project_person	InnoDB	Relationship between Projects and Persons
project_projectkeywords	InnoDB	
projectkeywords	InnoDB	Keywords to define a proj
projectmainfocus	InnoDB	
scaling	InnoDB	Scale factor in a scaled specimen
scaling_props	InnoDB	Properties for Scalation. This table is not essential
sensor	InnoDB	
sensorconfiguration	InnoDB	
sensorconfiguration_sensor	InnoDB	
sensorconfigurationsensor_image	InnoDB	Images of a sensor
servicerecord	InnoDB	Record of WS activity-Clear this table every 2months
signal	InnoDB	
signaltimes	InnoDB	Times for signals. Normally reused for signals.
software	InnoDB	Software used in a Test.
specimen	InnoDB	Specimens tested in a project.
specimen_document	InnoDB	Documents for a Specimen.
specimen_image	InnoDB	Images for a Specimen.
structuralcomponent	InnoDB	Structural Components in a Specimen.
structuralcomponent_document	InnoDB	Additional Documentation
structuralcomponent_material	InnoDB	Materials in a Struct Comp
structuralcomponenttype	InnoDB	Structural Systems or Elements (just names)

testing	InnoDB	For testing purpose.
updaterecord	InnoDB	For updating process
video	InnoDB	Videos.

Table: actualmeanproperty

Table: actualmeanproperty — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Material/Struct Com effective values.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idActualMeanProperty	int(10) unsigned	☐	☑		
ActualMeanPropertyName	varchar(40)	☐	☐		
ActualMeanPropertyValue	double	☑	☐		
ActualMeanPropertyUnit	varchar(20)	☑	☐		
numberOfSamples	tinyint(3) unsigned	☑	☐		
observations	varchar(150)	☑	☐		Observations, comments, conditions for the property or the value (f ex after or before which experiment this was observed).
valueVectorX	longblob	☑	☐		
valueVectorY	longblob	☑	☐		
hasDocument	enum('YES', 'NO')	☐	☐	NO	This col helps to improve performance in tables that are less likely to have associated Documents.
SCMAT_ID	int(10) unsigned	☐	☐		

Table: actualmeanproperty_document

Table: actualmeanproperty_document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
ActualMeanProperty_idActualMeanProperty	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		
documentType	enum('MATERIALCURVE', 'ADDITIONALDOC')	☐	☐		

Table: computersystem

Table: computersystem — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Configuration of Computations

Columns

Name	Data Type	NULL	Identity	Default	Comment
idComputerSystem	int(10) unsigned	☐	☑		
softwareName	varchar(45)	☐	☐		
softwareVersion	varchar(15)	☐	☐		

Table: dbinfo

Table: dbinfo

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	1
Avg row length:	16384
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	2
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Information about the Database

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDBinfo	int(10) unsigned	☐	☑		
databaseVersion	varchar(60)	☐	☐		Version of the Database
databaseDate	timestamp	☐	☐	2000-01-01 00:00:00	Time the database version was released
databaseUpdate	timestamp	☐	☐	CURRENT_TIMESTAMP	Time the database was installed/updated
notes	varchar(200)	☐	☐		
cacheVersion	varchar(30)	☐	☐	No cache	

Table: detailedloadingcharacteristic

Table: detailedloadingcharacteristic

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	It collects all the information that characterizes the exp

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDetailedLoadingCharacteristic	int(10) unsigned	☐	☑		
nominalLoadingName	varchar(40)	☐	☐		
additionalParameter	varchar(100)	☑	☐		
notes	text	☑	☐		

Table: detailedloadingcharacteri stic_originalloadingsigna l

Table: detailedloadingcharacteri stic_originalloadingsigna l

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
DetailedLoadingCharacteristic_idDetailedLoadingCharacteristic	int(10) unsigned	☐	☐		
OriginalLoadingSignal_idOriginalLoadingSignal	int(10) unsigned	☐	☐		
loadingCoefficient	float	☐	☐		
direction	enum('trans_horiz1', 'trans_horiz2', 'trans_vert', 'rot_horiz1', 'rot_horiz2', 'rot_vert')	☐	☐		
EffectiveInputSignal	int(10) unsigned	☑	☐		

Table: device

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDevice	int(10) unsigned	☐	☑		
deviceType	int(10) unsigned	☐	☐		Device itself: actuator, servo-valve, etc
deviceSubtype	varchar(20)	☐	☐		Subtype of the device:For actuators: symmetric, telescopic... For servo-valves: proportional...For controllers: analogical, digital...
deviceProductName	varchar(40)	☐	☐		Name on the company manufacturer's catalogue.
deviceLabel	varchar(20)	☐	☐		Local inventory name at the facility.
strokeCapacityValue	double unsigned	☐	☐		
strokeCapacityUnit	varchar(10)	☐	☐		
forceCapacityValue	double	☐	☐		
forceCapacityUnit	varchar(10)	☐	☐		
calibration	varchar(10)	☐	☐		
notes	mediumtext	☑	☐		
inventoryReference	varchar(30)	☑	☐		Reference in the local inventory or inventory database

Table: device_document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
Device_idDevice	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		

Table: deviceconfiguration

Table: deviceconfiguration

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDeviceConfiguration	int(10) unsigned	☐	☑		
configurationName	varchar(50)	☐	☐		
configurationNotes	text	☑	☐		

Table: deviceconfiguration_device

Table: deviceconfiguration_device Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDeviceConfigurationDevice	int(10) unsigned	☐	☑		
DeviceConfiguration_idDeviceConfiguration	int(10) unsigned	☐	☐		
Device_idDevice	int(10) unsigned	☐	☐		

Table: devicerelation

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	To link devices each other

Columns

Name	Data Type	NULL	Identity	Default	Comment
DeviceConfiguration_DeviceID	int(10) unsigned	☐	☐		
Parent_DeviceID	int(10) unsigned	☐	☐		
Child_DeviceID	int(10) unsigned	☐	☐		

Table: devicetype — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	6
Avg row length:	2730
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	7
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDeviceType	int(10) unsigned	☐	☑		
deviceTypeName	varchar(50)	☐	☐		

Table: document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	81920
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Generic document or file.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idDocument	int(10) unsigned	☐	☑		
documentTitle	varchar(150)	☐	☐		Main title of the document.
documentAuthor	varchar(300)	☐	☐		Author or authors of the document
documentCreationDate	timestamp	☐	☐	CURRENT_TIMESTAMP	Creation date of the document.
documentFormat	enum('DOC', 'PDF', 'TXT', 'ZIP', 'RAR', 'WEB', 'OTH')	☐	☐		Format of the document: DOC, PDF, TXT, WEB (unknown-URL link), OTHER (other file or unknown)...
documentSize	double unsigned	☐	☐	0	Document size in KB (should admit not only whole numbers --from bytes to KB the size could be 1.34, for example).
abstract	text	☐	☐		Abstract or summary of the document.
scope	varchar(150)	☐	☐	Project document	If the document was used for a Journal or Conference, this field defines the name of the Journal/Conference.Otherwise it has the default value 'Project document'
documentLocation	enum('INTERNAL', 'EXTERNAL', 'ENCLOSED')	☐	☐	INTERNAL	INTERNAL:doc in local site (EX:/series/mydocument.pdf or http://www.ox.ac.uk/series/mydocument.pdf)EXTERNAL:doc in external site,access protocol (http://, ftp://, etc) must be specified.ENCLOSED:doc embedded in the

					db,col:"enclosedDocument".
documentURI	varchar(255)	☑	☐		Document location (Uniform Resource Identifier).RELATIVE or ABSOLUTE URI.Must be enough for the system to know where (and how) to access the document.If the document is enclosed in the DB, this field can be set to "seriesdb://".
originalDocumentURI	varchar(255)	☑	☐		Original document location (Uniform Resource Identifier). It can be in the local machine or in a remote one.
enclosedDocument	longblob	☑	☐		Optional, to store the document within the DB
enclosedOriginalDocument	longblob	☑	☐		Optional, to store the original document within the DB
downloadTimes	int(10) unsigned	☑	☐	0	This field can be used for statistics or to restrict downloads.
hashValue	tinyblob	☐	☐		File CRC to avoid duplicates.
downloadAllowed	enum('YES', 'NO')	☐	☐	YES	This field can be used to restrict downloads
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	
creator_idUser	int(10) unsigned	☐	☐		
lastModification_idUser	int(10) unsigned	☐	☐		
lastModification_time	timestamp	☐	☐	2000-01-01 00:00:00	

Table: experimentcomputation

Table: experimentcomputation Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	98304
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idExpComp	int(10) unsigned	☐	☑		
expCompName	varchar(50)	☐	☐		
class	enum('EXPERIMENT', 'COMPUTATION')	☐	☐	EXPERIMENT	Type of element: Experiment or Computation.
type	int(10) unsigned	☐	☐		Type of the experiment (Shaking table, centrifuge, in-situ, hammer, etc)
date	timestamp	☐	☐	CURRENT_TIMESTAMP	
numberOfRepetitions	smallint(5) unsigned	☐	☐	1	The first Experiment/Computation must have a value of 1. The first repetition will have a value of 2, etc
peakExcitationValue	double unsigned	☐	☐		
peakExcitationUnit	varchar(20)	☐	☐		
SpecimenID	int(10) unsigned	☐	☐		
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	

creator_idUser	int(10) unsigned	☐	☐	
lastModification_idUser	int(10) unsigned	☐	☐	
lastModification_time	timestamp	☐	☐	2000-01-01 00:00:00
DeviceConfigurationID	int(10) unsigned	☑	☐	
SensorConfigurationID	int(10) unsigned	☑	☐	

Table: experimentcomputation_com putersystem

Table: experimentcomputation_com putersystem Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
ComputerSystem_idComputerSystem	int(10) unsigned	☐	☐		

Table: experimentcomputation_det ailedloadingcharacteristi c

Table: experimentcomputation_det ailedloadingcharacteristi c Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
DetailedLoadingCharacteristic_idDetailedLoadingCharacteristic	int(10) unsigned	☐	☐		
loadingCoefficient	float	☐	☐		

Table: experimentcomputation_document

Table: experimentcomputation_document Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For Experiment's input files, Boundary conditions...

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		
documentType	enum('BOUNDARYCONDITION', 'EXPLOGOBSERVATION', 'MMOBSERVATION')	☐	☐		- Experiment Log Observation field- Multi-media Observations

Table: experimentcomputation_file

Table: experimentcomputation_file

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
File_idFile	int(10) unsigned	☐	☐		
fileType	enum('INPUTFILE', 'RAWDATA', 'POSTPROCESSING')	☐	☐		

Table: experimentcomputation_image

Table: experimentcomputation_image

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For Boundary conditions

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
Image_idImage	int(10) unsigned	☐	☐		
imageType	enum('BOUNDARYCONDITION', 'MMGRAPHIC', 'MMPHOTO', 'MM3DPLOT')	☐	☐		

Table: experimentcomputation_person

Table: experimentcomputation_person Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Test agents in an experiment.

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
Person_idPerson	int(10) unsigned	☐	☐		
role	enum('TEST AGENT', 'OTHER')	☐	☐		

Table: experimentcomputation_video

Table: experimentcomputation_video Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
ExperimentComputation_idExpComp	int(10) unsigned	☐	☐		
Video_idVideo	int(10) unsigned	☐	☐		

Table: experimentcomputationtype

Table: experimentcomputationtype Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	27
Avg row length:	606
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	28
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For the type of experiments

Columns

Name	Data Type	NULL	Identity	Default	Comment
idExperimentComputationType	int(10) unsigned	☐	☑		
typeName	varchar(40)	☐	☐		
type	enum('EXPERIMENT', 'COMPUTATION')	☐	☐		

Table: file

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idFile	int(10) unsigned	☐	☑		
fileName	varchar(100)	☐	☐		
fileDate	timestamp	☐	☐	CURRENT_TIMESTAMP	
fileFormat	varchar(50)	☐	☐		
Description	mediumtext	☑	☐		
fileLocation	enum('INTERNAL', 'EXTERNAL', 'ENCLOSED')	☐	☐	INTERNAL	INTERNAL:file in local site (EX:/series/mydocument.pdf or http://www.ox.ac.uk/series/mydocument.pdf)EXTERNAL:file in external site,access protocol (http://, ftp://, etc) must be specified.ENCLOSED:file embedded in the db,col:"enclosedFile".
fileURI	varchar(255)	☑	☐		File location (Uniform Resource Identifier). It can be in the local machine or in a remote one.
enclosedFile	longblob	☑	☐		
hashValue	tinyblob	☐	☐		File CRC to avoid duplicates.

Table: image

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	81920
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Graphic image or photo.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idImage	int(10) unsigned	☐	☑		
imageName	varchar(50)	☐	☐		Name for the image
imageDate	timestamp	☑	☐		Date when the image was taken.
imageFormat	enum('JPG', 'BMP', 'TIFF', 'PNG', 'GIF', 'ZIP', 'RAR', 'WEB', 'OTH')	☐	☐		
imageLocation	enum('INTERNAL', 'EXTERNAL', 'ENCLOSED')	☐	☐	INTERNAL	INTERNAL:image in local site (EX:/series/img.jpg or http://www.ox.ac.uk/series/img.jpg)EXTERNAL:image in external site,access protocol (http://, ftp://, etc) must be specified.ENCLOSED:image embedded in the db,col:"enclosedImage".
imageAuthor	varchar(100)	☑	☐		Author or copyright owner (person, company...) of the image
imageURI	varchar(255)	☑	☐		Image location (Uniform Resource Identifier). It can be in the local machine or in a remote one.
enclosedImage	longblob	☑	☐		Optional to store the image in the DB
imageSize	double unsigned	☐	☐		Image size in KB (should admit not only whole numbers).

summary	tinytext	☑	☐		Brief description of the image.
hashValue	tinyblob	☐	☐		File CRC to avoid duplicates.
downloadAllowed	enum('YES', 'NO')	☐	☐	YES	
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	

Table: infrastructure

Table: infrastructure — Properties

Engine: InnoDB
Version: 10
Row format: Compact
Rows: 3
Avg row length: 5461
Data length: 16384
Max data length: 0
Index length: 32768
Data free: 8388608
Auto increment: 4
Create time: 01/31/2012
Update time:
Check time:
Collation: latin1_swedish_ci
Checksum:
Create options:
Comment: Location and name of the resources used in the project.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idInfrastructure	int(10) unsigned	☐	☑		Infrastructure ID.
infrastructureName	varchar(40)	☐	☐		Name and location of the infastructure. Ex: CEA SACLAY, ELSA ISPRA...
facilityName	varchar(40)	☐	☐		Name of the resource used within the infrastructure. Ex: AZALEE, CABLE FACILITY, COMPUTER...
location	varchar(20)	☐	☐		
InfrastructureImage	int(10) unsigned	☑	☐		

Table: institution

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	24
Avg row length:	682
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	25
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	List with all the participants institutions available

Columns

Name	Data Type	NULL	Identity	Default	Comment
idInstitution	int(10) unsigned	☐	☑		
institutionName	varchar(100)	☐	☐		Formal name of the institution. Ex: University of Oxford
institutionAcronym	varchar(15)	☑	☐		Acronym or short name for the institution. Ex: UOXF
institutionContact	varchar(20)	☑	☐		Contact Details, such as a generic phone number.
location	varchar(30)	☑	☐		
institutionWebsite	varchar(255)	☑	☐		

Table: material

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	7
Avg row length:	2340
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	8
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Represent a material.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idMaterial	int(10) unsigned	☐	☑		
materialName	varchar(45)	☐	☐		
hasDocument	enum('YES', 'NO')	☐	☐	NO	This col helps to improve performance in tables that are less likely to have associated Documents.

Table: material_document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Additional documentation

Columns

Name	Data Type	NULL	Identity	Default	Comment
Material_idMaterial	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		

Table: meshmodel

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idMeshModel	int(10) unsigned	☐	☑		
meshModelName	varchar(50)	☐	☐		
materialSymmetryType	enum('ISOTROPIC', 'ORTHOTROPIC', 'ANISOTROPIC', 'UNIAXIAL')	☐	☐		
materialNonlinearity	varchar(20)	☐	☐		
hasDocument	enum('YES', 'NO')	☐	☐	NO	This col helps to improve performance in tables that are less likely to have associated Documents.
hasImage	enum('YES', 'NO')	☐	☐	NO	This col helps to improve performance in tables that are less likely to have associated Images.
expCompID	int(10) unsigned	☐	☐		

Table: meshmodel_document Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For Additional Documentation

Columns

Name	Data Type	NULL	Identity	Default	Comment
MeshModel_idMeshModel	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		

Table: meshmodel_image Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For Undeformed shape picture

Columns

Name	Data Type	NULL	Identity	Default	Comment
MeshModel_idMeshModel	int(10) unsigned	☐	☐		
Image_idImage	int(10) unsigned	☐	☐		

Table: meshmodel_material

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
MeshModel_idMeshModel	int(10) unsigned	☐	☐		
Material_idMaterial	int(10) unsigned	☐	☐		

Table: nominalproperty

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Properties for materials

Columns

Name	Data Type	NULL	Identity	Default	Comment
idNominalProperty	int(10) unsigned	☐	☑		
nominalPropertyName	varchar(40)	☐	☐		
nominalPropertyValue	double	☐	☐		
nominalPropertyUnit	varchar(20)	☐	☐		
observations	varchar(150)	☑	☐		Observations, comments, conditions for the property or the value.
valueVectorX	longblob	☑	☐		
valueVectorY	longblob	☑	☐		
hasDocument	enum('YES', 'NO')	☐	☐	NO	This col helps to improve performance in tables that are less likely to have associated Documents.
materialID	int(10) unsigned	☐	☐		

Table: nominalproperty_document

Table: nominalproperty_document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
NominalProperty_idNominalProperty	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		
documentType	enum('MATERIALCURVE', 'ADDITIONALDOC')	☐	☐		

Table: originalloadingsignal

Table: originalloadingsignal

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Original inputs time-histories and related info.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idOriginalLoadingSignal	int(10) unsigned	☐	☑		
originalLoadingName	varchar(75)	☐	☐		
nature	enum('NATURAL', 'ARTIFICIAL', 'NATURAL-MODIFIED', 'GENERATED')	☐	☐		
source	varchar(60)	☐	☐		Source of the input signal, like for example:- a station and direction- a reference to a code of practice- the name of a finite element package or simulation- the name of some software (e.g. matlab)- the name of a signal generator
peakExcitationValue	double	☐	☐		
peakExcitationUnit	varchar(20)	☐	☐		
hasDocument	enum('YES', 'NO')	☐	☐	NO	This col helps to improve performance in tables that are less likely to have associated Documents.
signalID	int(10) unsigned	☐	☐		Signal Element

Table: originalloadingsignal_document

Table: originalloadingsignal_document Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
OriginalLoadingSignal_idOriginalLoadingSignal	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		

Table: partner

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idPartner	int(10) unsigned	☐	☑		
addrInformation	int(11)	☐	☐		IP information
NSInformation	varchar(50)	☑	☐		Name Server information. (Suggested to try to use this before the addrInformation).
notes	text	☑	☐		
institutionID	int(10) unsigned	☑	☐		

Table: person — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	3
Avg row length:	5461
Data length:	16384
Max data length:	0
Index length:	65536
Data free:	8388608
Auto increment:	4
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Human resources working in a Project.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idPerson	int(10) unsigned	☐	☑		ID of the Person
foreName	varchar(25)	☐	☐		Name of the person. Ex: Ignacio
familyName	varchar(80)	☐	☐		Surname of the person. Ex: Lamata Martinez
contactEmail	varchar(70)	☐	☐		Contact E-mail. Ex: ignacio.lamata@eng.ox.ac.uk
contactPhone	varchar(20)	☑	☐		Usual work contact phone (landline or mobile).
userImage	int(10) unsigned	☑	☐		
institutionID	int(10) unsigned	☐	☐		Institution ID of the Person.Institution in which the person works for.
idUser	int(10) unsigned	☑	☐		User ID for existing user interface tables (ex. UKNEES Website).
userID	int(10) unsigned	☑	☐		User ID for the interface.

Table: project

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	65536
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idProject	int(10) unsigned	☐	☑		ID of the Project
projectTitle	varchar(130)	☐	☐		Official title of the Project. Ex: Seismic Engineering Research Infrastructures for European Synergies
acronym	varchar(15)	☑	☐		Acronim or common name of the Project. Ex: SERIES, UKNEES...
projectStartDate	timestamp	☑	☐		Date when the project starts officially.
projectEndDate	timestamp	☑	☐		Date when the project is finished.
projectDescription	text	☑	☐		Brief description of the Project. This can be the project's official description.
projectCreationDate	timestamp	☐	☐	CURRENT_TIMESTAMP	Project creation date in the Database. It should hold the date that the project was added in the Database.
projectStatus	enum('NEW', 'FORESEEN', 'IN PROGRESS', 'FINISHED')	☐	☐	NEW	

fundingOrganization	varchar(100)	☑	☐		Organization funding the project
projectMainFocus	int(10) unsigned	☐	☐		
projectUpdate	enum('OPEN', 'CLOSED')	☐	☐	OPEN	
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	
creator_idUser	int(10) unsigned	☐	☐		ID of the user who added the Project into the Database.
lastModification_idUser	int(10) unsigned	☐	☐		
lastModification_time	timestamp	☐	☐	2000-01-01 00:00:00	

Table: project_document

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Reports in a Project.

Columns

Name	Data Type	NULL	Identity	Default	Comment
Project_idProject	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		
documentType	enum('PRELIMINARY', 'ONGOING', 'FINAL', 'JOURNAL', 'CONFERENCE')	☐	☐		

Table: project_infrastructure Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Main infrastructures and facilities used for the project.

Columns

Name	Data Type	NULL	Identity	Default	Comment
Project_idProject	int(10) unsigned	☐	☐		
Infrastructure_idInfrastructure	int(10) unsigned	☐	☐		

Table: project_person

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Relationship between Projects and Persons

Columns

Name	Data Type	NULL	Identity	Default	Comment
Project_idProject	int(10) unsigned	☐	☐		
Person_idPerson	int(10) unsigned	☐	☐		
role	enum('PRINCIPAL INVESTIGATOR', 'LOCAL COINVESTIGATOR', 'OTHER')	☐	☐		- Principal Investigator: People in highest level in charge of the project. Principal investigator at site infrastructure.- Local CoInvestigator- Test Agent

Table: project_projectkeywords

Table: project_projectkeywords Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
Project_idProject	int(10) unsigned	☐	☐		
ProjectKeywords_idProjectKeywords	int(10) unsigned	☐	☐		

Table: projectkeywords Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	39
Avg row length:	420
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	40
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Keywords to define a proj

Columns

Name	Data Type	NULL	Identity	Default	Comment
idProjectKeywords	int(10) unsigned	☐	☑		
keywordName	varchar(50)	☐	☐		

Table: projectmainfocus

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	11
Avg row length:	1489
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	12
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idProjectMainFocus	int(10) unsigned	☐	☑		
projectMainFocus	varchar(100)	☐	☐		

Table: scaling

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Scale factor in a scaled specimen

Columns

Name	Data Type	NULL	Identity	Default	Comment
idScaling	int(10) unsigned	☐	☑		
scaledProperty	int(10) unsigned	☐	☐		Lenght and Time "should" appear
PrototypeModel_ratio	double unsigned	☐	☐		Ratio of scalation. Same units as originally.
specimenID	int(10) unsigned	☐	☐		Specimen that has the scalation.

Table: scaling_props

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Properties for Scalation. This table is not essential

Columns

Name	Data Type	NULL	Identity	Default	Comment
idScaling_props	int(10) unsigned	☐	☑		
scaledPropertyName	varchar(25)	☐	☐		

Table: sensor — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSensor	int(10) unsigned	☐	☑		
sensorType	varchar(50)	☐	☐		Sensor type itself.
sensorSubtype	varchar(20)	☐	☐		Subtype of the sensor
sensorProductName	varchar(40)	☐	☐		Name on the company manufacturer's catalogue.
sensorLabel	varchar(20)	☐	☐		Local inventory name at the facility.
sensorAccuracyValue	double unsigned	☑	☐		
sensorAccuracyUnit	varchar(10)	☑	☐		
sensorRangeValue	varchar(10)	☑	☐		Informative value --thus not a number
sensorRangeUnit	varchar(10)	☑	☐		Not essential
calibration	varchar(10)	☑	☐		
inventoryReference	varchar(30)	☑	☐		Reference in the local inventory or inventory database

Table: sensorconfiguration

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSensorConfiguration	int(10) unsigned	☐	☑		
configurationName	varchar(50)	☐	☐		
configurationNotes	text	☑	☐		

Table: sensorconfiguration_sensor

Table: sensorconfiguration_sensor Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSensorConfiguration_Sensor	int(10) unsigned	☐	☑		
SensorConfiguration_idSensorConfiguration	int(10) unsigned	☐	☐		
Sensor_idSensor	int(10) unsigned	☐	☐		
sensorLocation	varchar(35)	☐	☐		
sensorLocationFinal	varchar(35)	☑	☐		
sensorDirection	varchar(45)	☐	☐		
sensorDirectionFinal	varchar(45)	☑	☐		
sensorCoordA_horiz1	double	☐	☐		
sensorFinalCoordA_horiz1	double	☑	☐		
sensorCoordA_horiz2	double	☐	☐		
sensorFinalCoordA_horiz2	double	☑	☐		

sensorCoordA_vert	double	☐	☐	
sensorFinalCoordA_vert	double	☑	☐	
sensorCoordB_horiz1	double	☑	☐	
sensorFinalCoordB_horiz1	double	☑	☐	
sensorCoordB_horiz2	double	☑	☐	
sensorFinalCoordB_horiz2	double	☑	☐	
sensorCoordB_vert	double	☑	☐	
sensorFinalCoordB_vert	double	☑	☐	
notes	mediumtext	☑	☐	
sensorPosition	int(10) unsigned	☑	☐	Document with a drawing: sketch of the specimen and the position of the sensors (sensor position field)

Table: sensorconfigurationsensor _image

Table: sensorconfigurationsensor _image

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Images of a sensor

Columns

Name	Data Type	NULL	Identity	Default	Comment
SensorConfiguration_Sensor	int(10) unsigned	☐	☐		
Image_idImage	int(10) unsigned	☐	☐		

Table: servicerecord

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Record of WS activity-Clear this table every 2months

Columns

Name	Data Type	NULL	Identity	Default	Comment
idServiceRecord	int(10) unsigned	☐	☑		
source	varchar(40)	☐	☐		Source computer to call the WS (IP/NS of computer accessing the WS)
startOpTime	timestamp	☐	☐	2000-01-01 00:00:00	
operationName	varchar(100)	☐	☐		
operationCompleted	enum('YES', 'NO')	☐	☐	NO	
endOpTime	timestamp	☑	☐		

Table: signal

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	81920
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSignal	int(10) unsigned	☐	☑		
signalLabel	varchar(25)	☐	☐		
attribute	varchar(30)	☐	☐		Characterizes the magnitude of the signal
physicalQuantity	varchar(20)	☐	☐		Possible ENUM field
type	enum('COMPUTED', 'MEASURED')	☐	☐		
unit	varchar(10)	☐	☐		
location	varchar(35)	☑	☐		
additionalParameter	varchar(60)	☑	☐		
signalValuesURI	varchar(255)	☐	☐		File with the signal values
signalValuesFormat	enum('SERIESBINARY', 'MATLAB', 'OTH')	☐	☐	SERIESBINARY	Format of the file with the signal values
valueVector	longblob	☐	☐		Vector for the signal valuesOLD: Use the format: VALUE,VALUE,VALUE,...UPDATE: Use binary format

complexValue	longblob	☑	☐		Optional if the signal is a complex number.OLD: Use the format: VALUE,VALUE,VALUE,...UPDATE : Use binary format
labTimeVectorID	int(10) unsigned	☐	☐		
accTimeVectorID	int(10) unsigned	☐	☐		
sensorConfig_sensorID	int(10) unsigned	☑	☐		Related Sensor, used for Experiments (computations have no associated sensor)
expCompID	int(10) unsigned	☑	☐		For OUTPUT signals: Related Experiment. "NULL" means it is not an output signal.
repetitionNumber	smallint(5) unsigned	☑	☐		For OUTPUT signals: repetition number for the Experiment/Computation "expCompID"
originalSignalFileLocation	enum('INTERNAL', 'EXTERNAL', 'ENCLOSED')	☑	☐	INTERNAL	
originalSignalFileURI	varchar(255)	☑	☐		
enclosedOriginalSignalFile	longblob	☑	☐		
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	
creator_idUser	int(10) unsigned	☐	☐		
lastModification_idUser	int(10) unsigned	☐	☐		
lastModification_time	timestamp	☐	☐	2000-01-01 00:00:00	

Table: signaltimes

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Times for signals. Normally reused for signals.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSignalTimes	int(10) unsigned	☐	☑		
timeLabel	varchar(25)	☐	☐		
timeVector	longblob	☐	☐		Vector for the signal times.OLD: Use the format: VALUE,VALUE,VALUE,...UPDATE: Use binary format

Table: software

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Software used in a Test.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSoftware	int(10) unsigned	☐	☑		
softwareName	varchar(45)	☐	☐		Name of the program.
softwareURI	varchar(255)	☐	☐		Software location (Uniform Resource Identifier). It can be in the local machine or in a remote one.
softwareVersion	varchar(15)	☐	☐	0	Version of the program.
hashValue	tinyblob	☐	☐		CRC of the program. It allows to avoid storing the same program several times.

Table: specimen — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Specimens tested in a project.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSpecimen	int(10) unsigned	☐	☑		
specimenName	varchar(45)	☐	☐		
projectID	int(10) unsigned	☐	☐		Project ID that holds the Specimen.
max_Length	double unsigned	☐	☐	0	Value in meters. "0" means unknown or N/A
max_Width	double unsigned	☐	☐	0	Value in meters. "0" means unknown or N/A
max_Height	double unsigned	☐	☐	0	Value in meters. "0" means unknown or N/A
max_Depth	double	☐	☐	0	Value in meters. "0" means unknown or N/A
specimenMass	double unsigned	☐	☐	0	Value in Kgr. "0" means unknown or N/A
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	
creator_idUser	int(10) unsigned	☐	☐		
lastModification_idUser	int(10) unsigned	☐	☐		

lastModification_time	timestamp	☐	☐	2000-01-01 00:00:00

Table: specimen_document

Table: specimen_document — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Documents for a Specimen.

Columns

Name	Data Type	NULL	Identity	Default	Comment
Specimen_idSpecimen	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		
documentType	enum('COORDREFSYSTEM', 'SCALING', 'GEOMETRY', 'CONSTRUCTION', 'TRANSPORT', 'DEMOLITION', 'PRELIMINARY', 'ONGOING', 'FINAL')	☐	☐		

Table: specimen_image

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Images for a Specimen.

Columns

Name	Data Type	NULL	Identity	Default	Comment
Specimen_idSpecimen	int(10) unsigned	☐	☐		
Image_idImage	int(10) unsigned	☐	☐		
imageType	enum('CONSTRUCTION', 'TRANSPORT', 'DEMOLITION', 'MAIN')	☐	☐		

Table: structuralcomponent

Table: structuralcomponent Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	65536
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Structural Components in a Specimen.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idStructuralComponent	int(10) unsigned	☐	☑		
structuralComponentName	varchar(70)	☐	☐		Custom name
specimenID	int(10) unsigned	☐	☐		
SCtypeID	int(10) unsigned	☐	☐		Name (type) of the Structural Element. Ex: 2D frame,tunnels,reinforced soil, etc
materialDescription	varchar(100)	☑	☐		

Table: structuralcomponent_document

Table: structuralcomponent_document Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	32768
Data free:	8388608
Auto increment:	
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Additional Documentation

Columns

Name	Data Type	NULL	Identity	Default	Comment
StructuralComponent_idStructuralComponent	int(10) unsigned	☐	☐		
Document_idDocument	int(10) unsigned	☐	☐		

Table: structuralcomponent_material

Table: structuralcomponent_material Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Materials in a Struct Comp

Columns

Name	Data Type	NULL	Identity	Default	Comment
idSCMAT	int(10) unsigned	☐	☑		
StructuralComponent_idStructuralComponent	int(10) unsigned	☐	☐		
Material_idMaterial	int(10) unsigned	☐	☐		

Table: structuralcomponenttype

Table: structuralcomponenttype Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	51
Avg row length:	321
Data length:	16384
Max data length:	0
Index length:	16384
Data free:	8388608
Auto increment:	52
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Structural Systems or Elements (just names)

Columns

Name	Data Type	NULL	Identity	Default	Comment
idStructuralComponentType	int(10) unsigned	☐	☑		
typeName	varchar(90)	☐	☐		Ex: 2D frame,tunnels,reinforced soil, etc

Table: testing — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	1
Avg row length:	16384
Data length:	16384
Max data length:	0
Index length:	0
Data free:	8388608
Auto increment:	2
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For testing purpose.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idTesting	int(10) unsigned	☐	☑		
text	varchar(20)	☐	☐		

Table: updaterecord

Table: updaterecord — Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	49152
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	For updating process

Columns

Name	Data Type	NULL	Identity	Default	Comment
idUpdateRecord	int(10) unsigned	☐	☑		
level	enum('PROJECT', 'SPECIMEN', 'EXPERIMENTCOMPUTATION', 'SIGNAL', 'DOCUMENT')	☐	☐		
objectName	varchar(25)	☐	☐		
objectID	int(10) unsigned	☐	☐		
auxID	int(10) unsigned	☑	☐		Some objects might need to store an auxiliar ID (parent ID, etc)
operation	enum('NEW', 'UPDATE', 'DELETE')	☐	☐		
updateDate	timestamp	☐	☐	CURRENT_TIMESTAMP	
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE	

Table: video

Properties

Engine:	InnoDB
Version:	10
Row format:	Compact
Rows:	0
Avg row length:	0
Data length:	16384
Max data length:	0
Index length:	65536
Data free:	8388608
Auto increment:	1
Create time:	01/31/2012
Update time:	
Check time:	
Collation:	latin1_swedish_ci
Checksum:	
Create options:	
Comment:	Videos.

Columns

Name	Data Type	NULL	Identity	Default	Comment
idVideo	int(10) unsigned	☐	☑		
videoName	varchar(50)	☐	☐		Name for the video.
videoDate	timestamp	☑	☐		Date when the video was taken.
videoFormat	enum('MPG', 'AVI', 'MOV', 'FLV', 'ZIP', 'RAR', 'WEB', 'OTH')	☐	☐		
videoLocation	enum('INTERNAL', 'EXTERNAL', 'ENCLOSED')	☐	☐	INTERNAL	INTERNAL:vid in local site (EX:/series/myvid.mpg or http://www.ox.ac.uk/series/myvid.mpg)EXTERNAL:vid in external site,access protocol (http://, ftp://, etc) must be specified.ENCLOSED:vid embedded in the db,col:"enclosedVideo".
videoURI	varchar(255)	☑	☐		Video location (Uniform Resource Identifier). It can be in the local machine or in a remote one.
enclosedVideo	longblob	☑	☐		Optional to store the video in the DB (warning: size)
videoSize	double unsigned	☐	☐		Video size in KB (should admit not only whole numbers).
summary	tinytext	☑	☐		Brief description of the video.
hashValue	tinyblob	☐	☐		File CRC to avoid duplicates.

downloadAllowed	enum('YES', 'NO')	☐	☐	YES
privacy	enum('PRIVATE', 'PARTNER', 'PUBLIC')	☐	☐	PRIVATE

Laboratory Web Services Description Language

```
<definitions xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"xmlns:wsp="http://www.w3.org/ns/ws-
policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/poli
cy"xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xm
lns:soap="http://schemas.xmlsoap.org/wsdl/soap/"xmlns:tns="http:
//ilm.eng.ox.ac.uk/" xmlns:xsd="http://www.w3.org/2001/XMLSchema
" xmlns="http://schemas.xmlsoap.org/wsdl/"targetNamespace="http:
//ilm.eng.ox.ac.uk/" name="DOS">
<types>
<xsd:schema>
<xsd:import namespace="http://ilm.eng.ox.ac.uk/" schemaLocation=
"http://lab/DatOX/DOS?xsd=1"/>
</xsd:schema>
</types>
<message name="getVersion">
<part name="parameters" element="tns:getVersion"/>
</message>
<message name="getVersionResponse">
<part name="parameters" element="tns:getVersionResponse"/>
</message>
<message name="getCacheInfo">
<part name="parameters" element="tns:getCacheInfo"/>
</message>
<message name="getCacheInfoResponse">
<part name="parameters" element="tns:getCacheInfoResponse"/>
</message>
<message name="testMe">
<part name="parameters" element="tns:testMe"/>
</message>
<message name="testMeResponse">
<part name="parameters" element="tns:testMeResponse"/>
</message>
<message name="getProjectIDs">
<part name="parameters" element="tns:getProjectIDs"/>
</message>
<message name="getProjectIDsResponse">
<part name="parameters" element="tns:getProjectIDsResponse"/>
</message>
<message name="getProjectData">
<part name="parameters" element="tns:getProjectData"/>
</message>
<message name="getProjectDataResponse">
<part name="parameters" element="tns:getProjectDataResponse"/>
```

```
</message>
<message name="getSpecimenData">
<part name="parameters" element="tns:getSpecimenData"/>
</message>
<message name="getSpecimenDataResponse">
<part name="parameters" element="tns:getSpecimenDataResponse"/>
</message>
<message name="getProjectDocuments">
<part name="parameters" element="tns:getProjectDocuments"/>
</message>
<message name="getProjectDocumentsResponse">
<part name="parameters" element="tns:getProjectDocumentsResponse
"/>
</message>
<message name="getSpecimenDocuments">
<part name="parameters" element="tns:getSpecimenDocuments"/>
</message>
<message name="getSpecimenDocumentsResponse">
<part name="parameters" element="tns:getSpecimenDocumentsRespons
e"/>
</message>
<message name="getSpecimenImages">
<part name="parameters" element="tns:getSpecimenImages"/>
</message>
<message name="getSpecimenImagesResponse">
<part name="parameters" element="tns:getSpecimenImagesResponse"/
>
</message>
<message name="testMyKR">
<part name="parameters" element="tns:testMyKR"/>
</message>
<message name="testMyKRResponse">
<part name="parameters" element="tns:testMyKRResponse"/>
</message>
<message name="getStructuralComponentData">
<part name="parameters" element="tns:getStructuralComponentData"
/>
</message>
<message name="getStructuralComponentDataResponse">
<part name="parameters" element="tns:getStructuralComponentDataR
esponse"/>
</message>
<message name="getExperimentData">
<part name="parameters" element="tns:getExperimentData"/>
</message>
<message name="getExperimentDataResponse">
<part name="parameters" element="tns:getExperimentDataResponse"/
>
```

```
</message>
<message name="getComputationData">
<part name="parameters" element="tns:getComputationData"/>
</message>
<message name="getComputationDataResponse">
<part name="parameters" element="tns:getComputationDataResponse"
/>
</message>
<message name="getExperimentLoadingData">
<part name="parameters" element="tns:getExperimentLoadingData"/>
</message>
<message name="getExperimentLoadingDataResponse">
<part name="parameters" element="tns:getExperimentLoadingDataRes
ponse"/>
</message>
<message name="getComputationLoadingData">
<part name="parameters" element="tns:getComputationLoadingData"/
>
</message>
<message name="getComputationLoadingDataResponse">
<part name="parameters" element="tns:getComputationLoadingDataRe
sponse"/>
</message>
<message name="getExperimentDocuments">
<part name="parameters" element="tns:getExperimentDocuments"/>
</message>
<message name="getExperimentDocumentsResponse">
<part name="parameters" element="tns:getExperimentDocumentsRespo
nse"/>
</message>
<message name="getComputationDocuments">
<part name="parameters" element="tns:getComputationDocuments"/>
</message>
<message name="getComputationDocumentsResponse">
<part name="parameters" element="tns:getComputationDocumentsResp
onse"/>
</message>
<message name="getExperimentImages">
<part name="parameters" element="tns:getExperimentImages"/>
</message>
<message name="getExperimentImagesResponse">
<part name="parameters" element="tns:getExperimentImagesResponse
"/>
</message>
<message name="getComputationImages">
<part name="parameters" element="tns:getComputationImages"/>
</message>
<message name="getComputationImagesResponse">
```

```
<part name="parameters" element="tns:getComputationImagesRespons
e"/>
</message>
<message name="getExperimentVideos">
<part name="parameters" element="tns:getExperimentVideos"/>
</message>
<message name="getExperimentVideosResponse">
<part name="parameters" element="tns:getExperimentVideosResponse
"/>
</message>
<message name="getComputationVideos">
<part name="parameters" element="tns:getComputationVideos"/>
</message>
<message name="getComputationVideosResponse">
<part name="parameters" element="tns:getComputationVideosRespons
e"/>
</message>
<message name="operationCompleted">
<part name="parameters" element="tns:operationCompleted"/>
</message>
<message name="operationCompletedResponse">
<part name="parameters" element="tns:operationCompletedResponse"
/>
</message>
<portType name="DOS">
<operation name="getVersion">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getVersionReques
t" message="tns:getVersion"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getVersionRespo
nse" message="tns:getVersionResponse"/>
</operation>
<operation name="getCacheInfo">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getCacheInfoRequ
est" message="tns:getCacheInfo"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getCacheInfoRes
ponse" message="tns:getCacheInfoResponse"/>
</operation>
<operation name="testMe">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/testMeRequest" m
essage="tns:testMe"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/testMeResponse"
 message="tns:testMeResponse"/>
</operation>
<operation name="getProjectIDs">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getProjectIDsReq
uest" message="tns:getProjectIDs"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getProjectIDsRe
sponse" message="tns:getProjectIDsResponse"/>
```

```
</operation>
<operation name="getProjectData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getProjectDataRe
quest" message="tns:getProjectData"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getProjectDataR
esponse" message="tns:getProjectDataResponse"/>
</operation>
<operation name="getSpecimenData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getSpecimenDataR
equest" message="tns:getSpecimenData"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getSpecimenData
Response" message="tns:getSpecimenDataResponse"/>
</operation>
<operation name="getProjectDocuments">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getProjectDocume
ntsRequest" message="tns:getProjectDocuments"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getProjectDocum
entsResponse"message="tns:getProjectDocumentsResponse"/>
</operation>
<operation name="getSpecimenDocuments">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getSpecimenDocum
entsRequest" message="tns:getSpecimenDocuments"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getSpecimenDocu
mentsResponse"message="tns:getSpecimenDocumentsResponse"/>
</operation>
<operation name="getSpecimenImages">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getSpecimenImage
sRequest" message="tns:getSpecimenImages"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getSpecimenImag
esResponse" message="tns:getSpecimenImagesResponse"/>
</operation>
<operation name="testMyKR">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/testMyKRRequest"
 message="tns:testMyKR"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/testMyKRRespons
e" message="tns:testMyKRResponse"/>
</operation>
<operation name="getStructuralComponentData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getStructuralCom
ponentDataRequest"message="tns:getStructuralComponentData"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getStructuralCo
mponentDataResponse"message="tns:getStructuralComponentDataRespo
nse"/>
</operation>
<operation name="getExperimentData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentDat
aRequest" message="tns:getExperimentData"/>
```

```
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentDa
taResponse" message="tns:getExperimentDataResponse"/>
</operation>
<operation name="getComputationData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationDa
taRequest" message="tns:getComputationData"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationD
ataResponse"message="tns:getComputationDataResponse"/>
</operation>
<operation name="getExperimentLoadingData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentLoa
dingDataRequest"message="tns:getExperimentLoadingData"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentLo
adingDataResponse"message="tns:getExperimentLoadingDataResponse"
/>
</operation>
<operation name="getComputationLoadingData">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationLo
adingDataRequest"message="tns:getComputationLoadingData"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationL
oadingDataResponse"message="tns:getComputationLoadingDataRespons
e"/>
</operation>
<operation name="getExperimentDocuments">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentDoc
umentsRequest" message="tns:getExperimentDocuments"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentDo
cumentsResponse"message="tns:getExperimentDocumentsResponse"/>
</operation>
<operation name="getComputationDocuments">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationDo
cumentsRequest"message="tns:getComputationDocuments"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationD
ocumentsResponse"message="tns:getComputationDocumentsResponse"/>
</operation>
<operation name="getExperimentImages">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentIma
gesRequest" message="tns:getExperimentImages"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentIm
agesResponse"message="tns:getExperimentImagesResponse"/>
</operation>
<operation name="getComputationImages">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationIm
agesRequest" message="tns:getComputationImages"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationI
magesResponse"message="tns:getComputationImagesResponse"/>
</operation>
<operation name="getExperimentVideos">
```

```
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentVid
eosRequest" message="tns:getExperimentVideos"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getExperimentVi
deosResponse"message="tns:getExperimentVideosResponse"/>
</operation>
<operation name="getComputationVideos">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationVi
deosRequest" message="tns:getComputationVideos"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/getComputationV
ideosResponse"message="tns:getComputationVideosResponse"/>
</operation>
<operation name="operationCompleted">
<input wsam:Action="http://ilm.eng.ox.ac.uk/DOS/operationComplet
edRequest" message="tns:operationCompleted"/>
<output wsam:Action="http://ilm.eng.ox.ac.uk/DOS/operationComple
tedResponse"message="tns:operationCompletedResponse"/>
</operation>
</portType>
<binding name="DOSPortBinding" type="tns:DOS">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" s
tyle="document"/>
<operation name="getVersion">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getCacheInfo">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="testMe">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
```

```
<operation name="getProjectIDs">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getProjectData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getSpecimenData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getProjectDocuments">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getSpecimenDocuments">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getSpecimenImages">
<soap:operation soapAction=""/>
<input>
```

```
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="testMyKR">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getStructuralComponentData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getExperimentData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getComputationData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getExperimentLoadingData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
```

```
<soap:body use="literal"/>
</output>
</operation>
<operation name="getComputationLoadingData">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getExperimentDocuments">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getComputationDocuments">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getExperimentImages">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getComputationImages">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
```

```
<operation name="getExperimentVideos">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getComputationVideos">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="operationCompleted">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="DOS">
<port name="DOSPort" binding="tns:DOSPortBinding">
<soap:address location="http://lab/DatOX/DOS"/>
</port>
</service>
</definitions>
```