JISC DEVELOPMENT PROGRAMMES

Project Document Cover Sheet

Integrating Service Oriented Architecture with a Virtual Research Environment

Project

Project Acronym	CORE	**Project ID**	
Project Title	Collaborative Orthopaedic Research Environment		
Start Date	01 November 2004	**End Date**	31 October 2006
Lead Institution	University of Southampton		
Project Manager & contact details	Dr Gary Wills Intelligence, Agents, Multimedia Group School of Electronics and Computer Science University of Southampton Southampton, SO17 1BJ		
Project Web URL	www.core.ecs.soton.ac.uk		
Programme Name (and number)	*Virtual Research Environments (05/04) Strand III*		
Programme Manager	Dr Maia Dimitrova		

Document

Document Title	Integrating Service Oriented Architecture with a Virtual Research Environment		
Reporting Period			
Author(s) & Project Role	Chu Wang (Co-Investigator), Yee Wai Sim (Co-Investigator), Lester Gilbert (Technical Manager), Gary Wills (Project Manager)		
Date	01/06/2005	**Filename**	ecstr_iam05_002.doc
ISBN	0854328246		
Technical Report Number	ECSTR-IAM05-002		
Access	☐ Project and JISC internal	■ General dissemination	

Document History

Version	Date	Comments
0a	1 June 2005	Draft version
1a	14 June 2005	Final Version

Integrating Service Oriented Architecture with a Virtual Research Environment

C. Wang, Y. W. Sim, L. Gilbert, G. B. Wills
School of Electronics and Computer Science
University of Southampton
E-mail: {cw2, yws01, lg3, gbw}@ecs.soton.ac.uk
Technical report Number: ECSTR-IAM05-002
ISBN: 0854328246

Abstract

A Virtual Research Environment using Service Oriented Architecture is designed to support surgeons and scientists in Bone Laboratory in carrying out trials and disseminating results. Based on the existing VRE system, the new system is reengineered as a loosely coupled web/grid services. The report describes the procedure of how to redesign and re-implement the VRE with focus on the work flows and processes within the server architecture and external web services.

1 Introduction

The Collaborative Orthopaedic Research Environment (CORE) [1] project is to develop and deploy a service enabling orthopaedic surgeons to collaborate in the design, analysis, and dissemination of experiments. CORE builds on the work carried out under the Virtual Orthopaedic European University (VOEU) [2] project. As part of the VOEU project -- a Dynamic Review Journal (DRJ), which is a tightly integrated system allowing researchers and clinicians working in bone laboratory to collaborate on clinical trials, was developed. The CORE project will enhance the DRJ by developing and deploying a Web services based Virtual Research Environment (VRE) demonstrator using Service Oriented Architecture (SOA) [4] approach, which will enable researchers to design experiments collaboratively, collect the results and disseminate the findings in a loosely coupled system.

This report describes how to integrate the CORE SOA with existing components in VOEU, at the early stage of SOA design. The work carried out initially includes a thorough study of current VOEU environment, in particular the work flow in DRJ, which is illustrated in detail in the following section. A redesign of VRE components based on a loosely coupled architecture is explained later in this report, which reflects the current status of this project.

2 Existing VOEU DRJ system

VOEU is dedicated to the ongoing professional education of Orthopedic Surgeons [3]. Within VOEU, DRJ is a tightly coupled system for aiding surgeons to write papers collaboratively. It is designed to meet two main functions: to aid surgeons in preparing findings for publication, and to support the educational process.

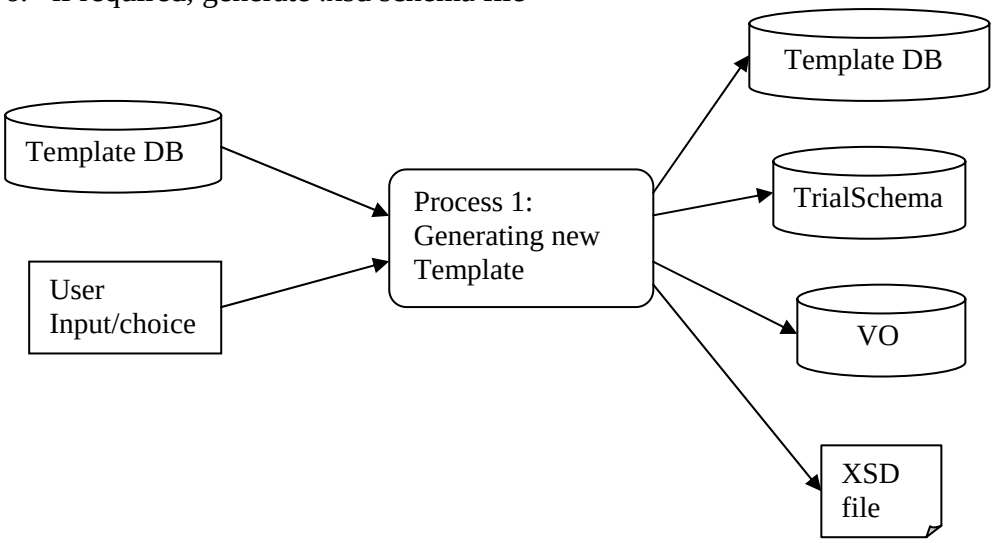
In order to design a web service based VRE on the foundation of VOEU, it is very important to decompose the working processes within DRJ to help us understand the system better.

2.1 Working Processes in DRJ

There are 15 processes which have been identified within DRJ. The work flow for each process is described as follows.

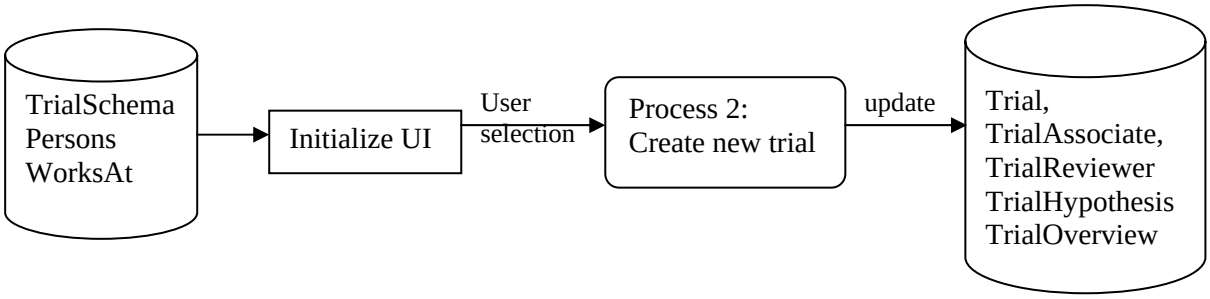
Process 1: Generating New Template

1. Retrieve current templates from Template Database (DB).
2. Record variables and mapping information for the new template
3. Get unique name from user input
4. Store the new template variables and mapping information to Template DB
5. If required, store the new template(schema) info into TrialSchema and DRJschema DB tables; and generate a new Virtual Observatory (VO) DB table (in the form of VO+schemaName)
6. if required, generate .xsd schema file



Process 2: Create New Trial

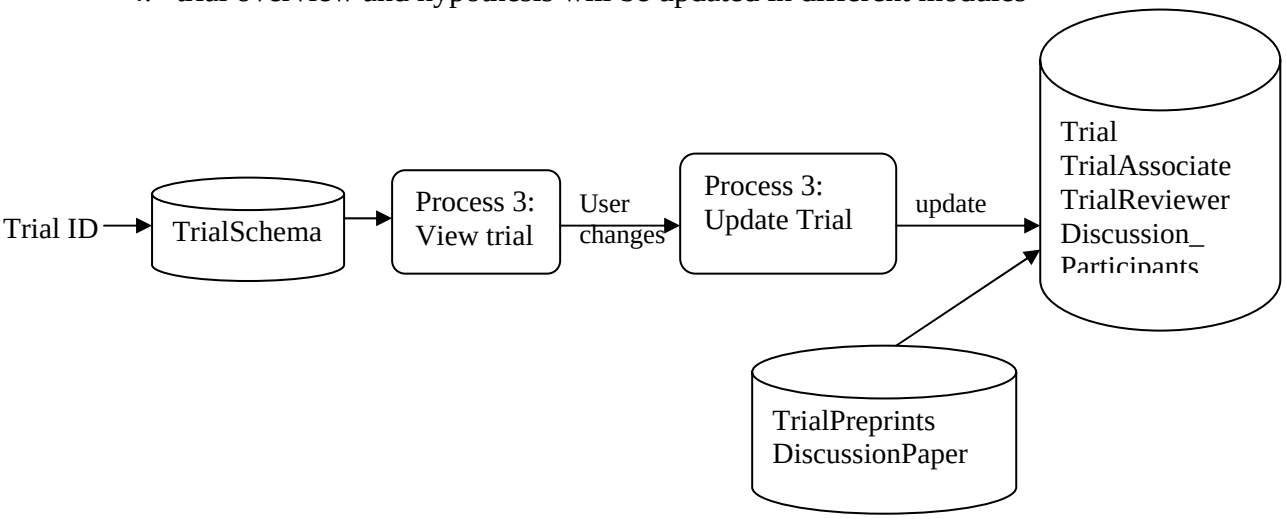
1. Initialize User Interface (UI) by displaying schema list and edit-person panel (TrialSchema, Person, WorksAt DB tables)
2. Upon user's selection of a schema and investigator/reviewer, add into Trial, TrialAssociate, TrialReviewer
3. Create empty hypothesis and overview for the trial and insert into DB tables TrialHypothesis and TrialOverview.



- Meta data associated with a new trial:
  Trial Identifier, Date commenced, Planned completion date, Principal investigator, Trial schema, Trial Associate, and Trial reviewer.

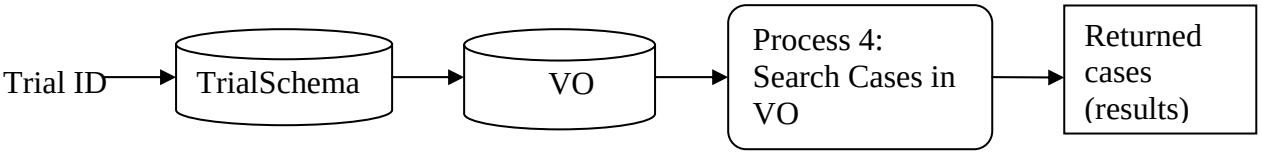
Process 3: Update Trial (View Trial) [edittrial.ascx]

1. Display current trial information, including schema, date, people (principle, associate, reviewer) (in module viewtrial.ascx, edittrial.ascx)
2. Store the changes user made into DB (trial, trialAssociate, trialReviewer)
3. Update the people information associated with discussion (new associates/reviewer instead of old ones); need to access trialPreprints (get preprintID) and discussionPapers, to find the discussion associated with a paper
4. trial overview and hypothesis will be updated in different modules



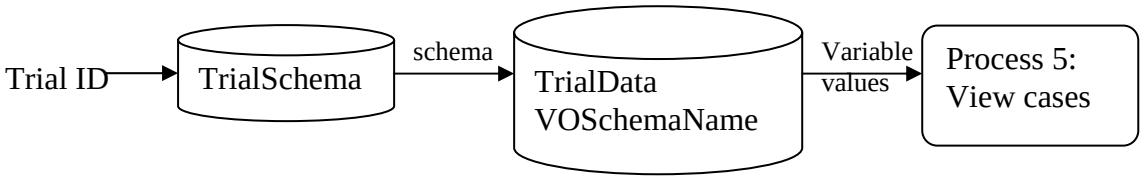
Process 4: Search for Cases in VO [module vodata.ascx]

1. Locate associated trial ID and trial schema
2. Generate search form dynamically based on schema variables
3. Locate VO data table using the name VO+shemaName
4. Search cases within this VO DB table
5. Return resulting cases to user



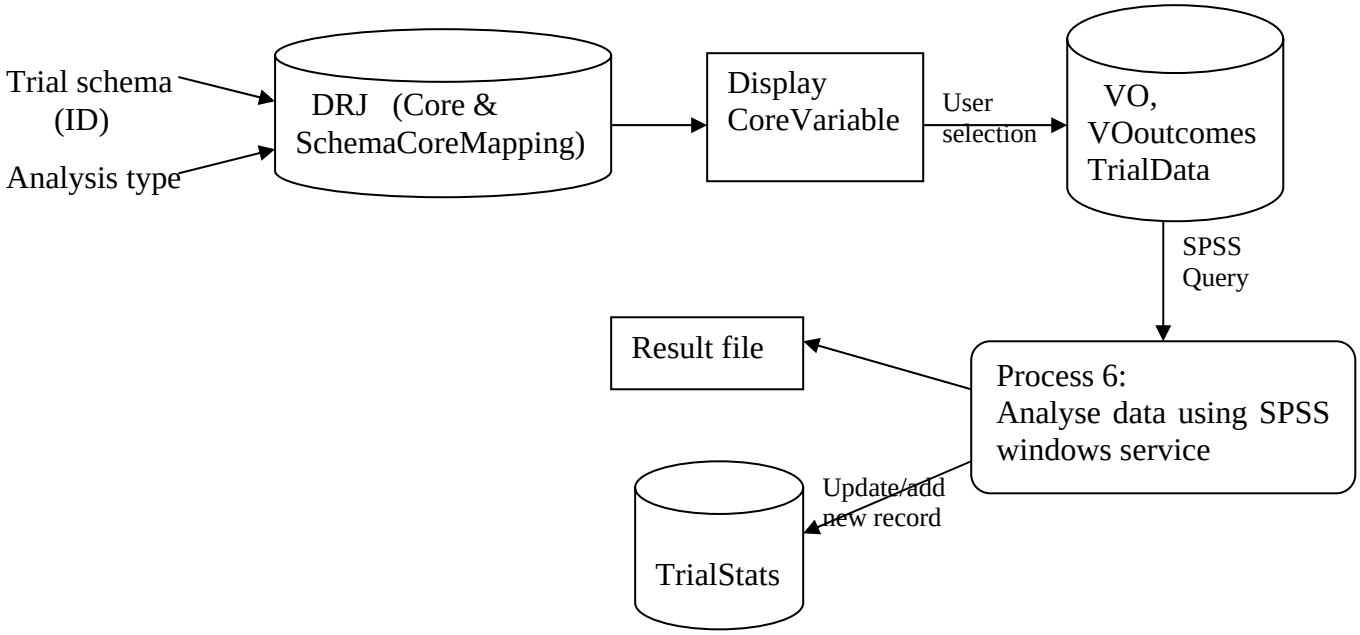
Process 5: View Cases
[viewcases (view all cases), viewdata (view individual case), cases]

1. Get trial schema based on trial (id)
2. Retrieve information (core variables) from TrialData and VOschemaName DB.
3. Present all cases in table based on schema core variables
4. Present individual case from hyperlink



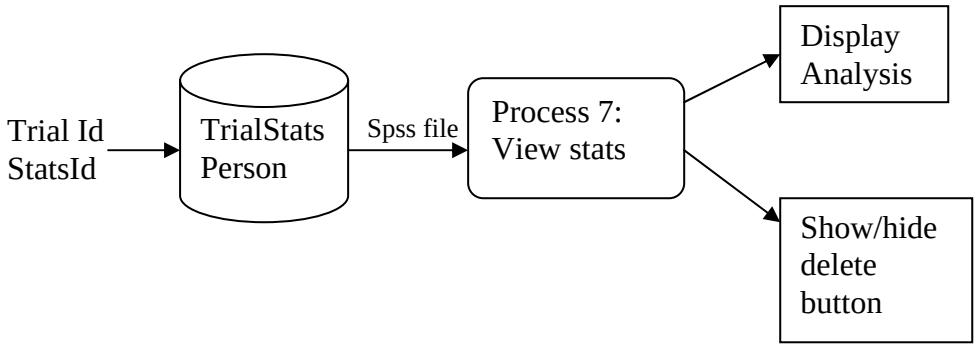
Process 6: Analyse Data (generate new statistics)

1. Get user's choice of analysis type
2. Select relevant variables (DRJCore elements) from DB, based on trial schema, analysis type (decide necessary variable types).
3. Display selected variables to the UI
4. Get user's selection and generate SQL query and SPSS query
5. SPSS windows service will perform the query after detecting the query file in the directory, then generate result file (sps.htm). (SPSS.processor to generate query file)
6. Locate result file and display



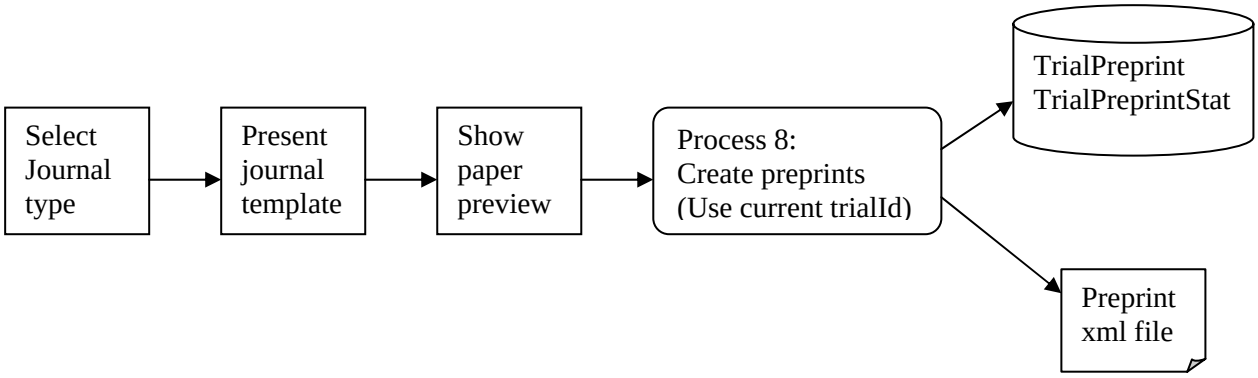
Process 7: View Statistics [viewStats.ascx]

1. Retrieve stats information (based on trialId, statsId) from Person (for displaying creator information), TrialStats.
2. Find SPSS analysis result file (sps.htm) to present the statistics.
3. Show/hide delete button according to roles (principle/associate, reviewer)
4. Upon user's request for deleting, remove record from TrialStats, TrialPreprintstats tables



Process 8: Create Preprints (new preprint) [newpreprint.ascx]

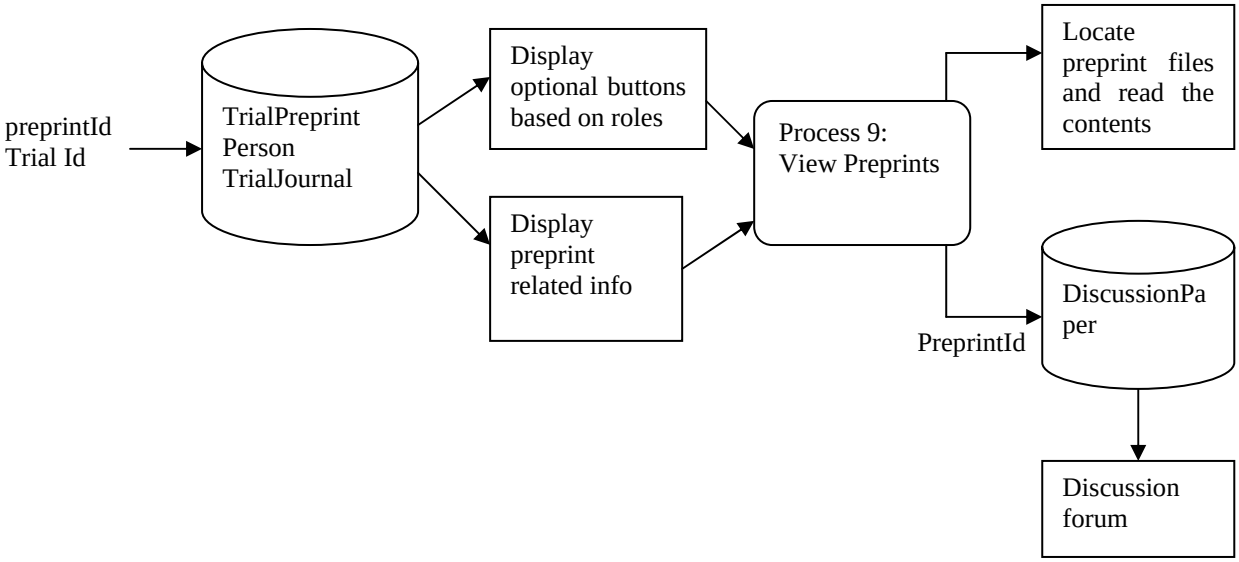
1. Select journal type
2. Present template based on journal type and trial information (with relevant analysis, etc)
3. Show preview of the paper
4. Upon user clicking submit button, create new record in TrialPreprint
5. Update TrialPreprintStats table with selected (associated with paper) statistics.
6. Create xml file in the trialPreprint directory



Process 9: View Preprints [viewpreprint.ascx]

1. Retrieve preprints related data from TrialPreprint, person (creator), TrialJournal.
2. Locate preprint file and previous versions in the file directory (by trialId)
3. Link to discussion forum from current paper, using DiscussionPaper DB table to present relevant discussions to a paper (via preprinted)
4. Present relevant buttons based on person's role (principle, associate or reviewer)
5. Delete preprint files (including previous versions) and database entries (from trialPreprints table) upon user's request

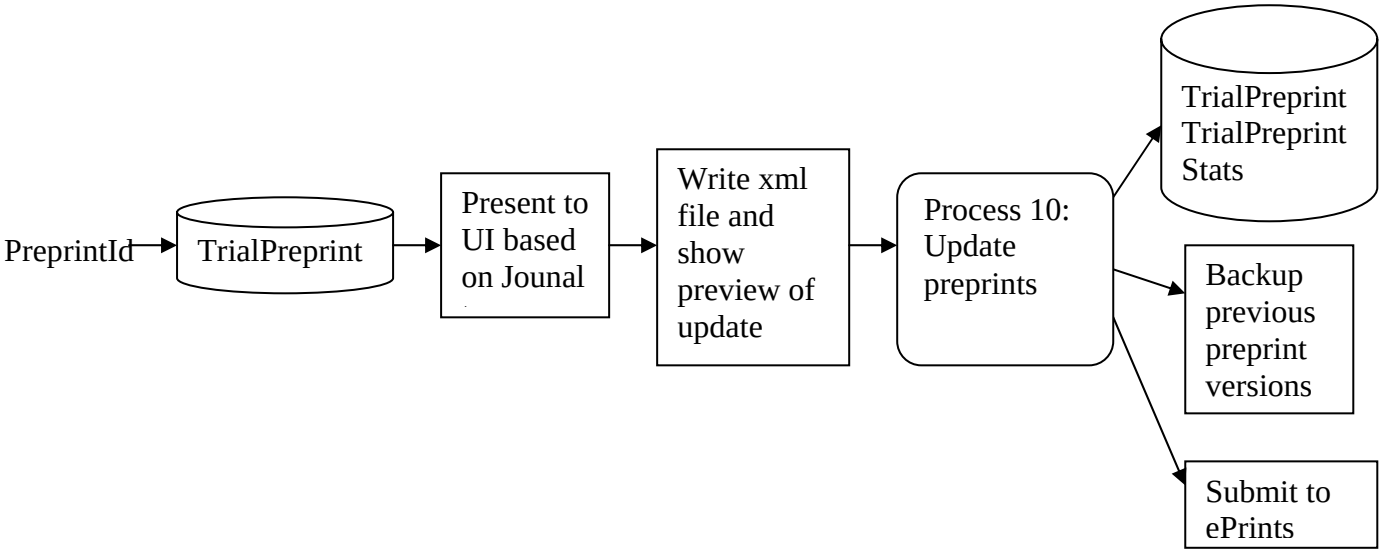
- differentiate between preprints (trialId and preprintId) and paper (paperId)



Process 10: Update Preprints

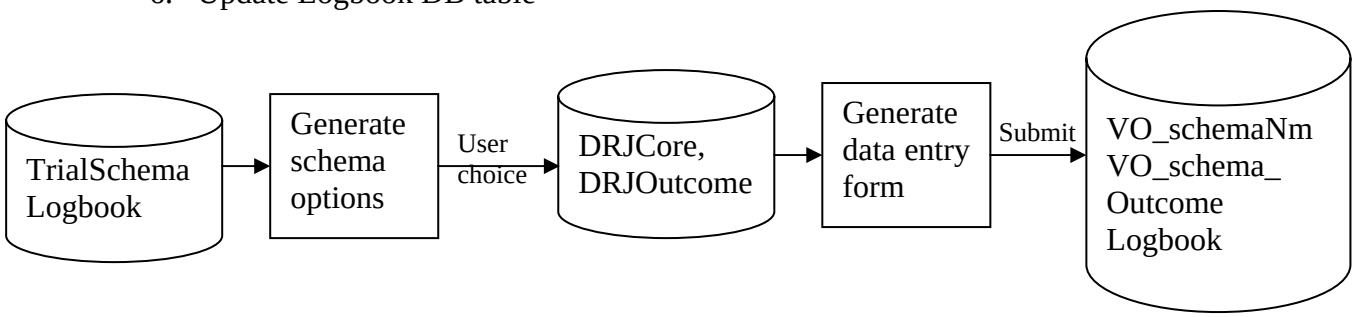
1. Retrieve information for current preprint version from TrialPreprint DB, read information from xml file (meta data like version), and present to UI (via preprintId, trialId)

2. Show preview to user upon request
3. Update DB TrialPreprint, TrialPreprintStats upon submission
4. Back up previous version of preprints
5. Submit preprint id to EPrints service, add EPrints ID to DB TrialPreprints



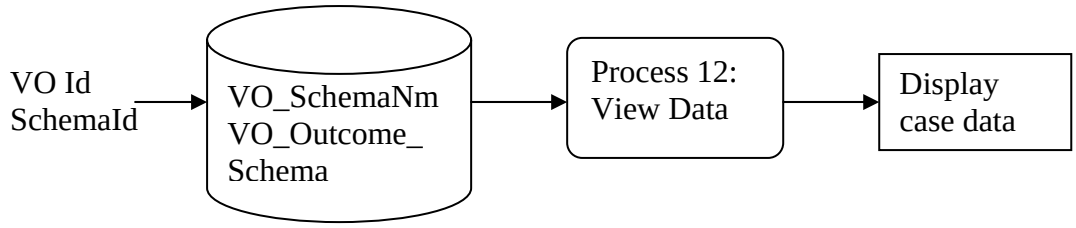
Process 11: Add Data (to My data / Logbook) [newlogbook.ascx]

1. Generate an ordered list of schema options (based on a current user's schema generation)
2. Upon catching user's schema choice, dynamically produce an entry form using DRJCore and DRJOutcomes database tables.
3. The entry cell can be either textbox or dropdown list based on the variable type and range.
4. On user's submission, generate/add into the VO_schemaName table with the entries.
5. Update VO SchemaOutcome table as well, if the outcome variables are available.
6. Update Logbook DB table



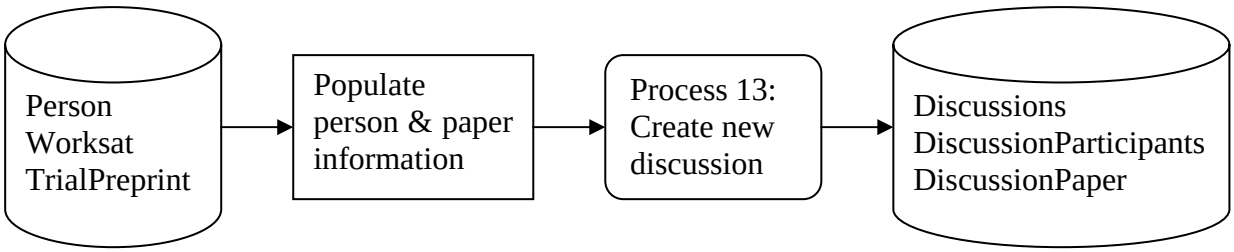
Process 12: View Data (in Logbook) [module viewdata.ascx]

1. Display case data based on caseId (VOId) and schemaId
2. Retrieve data from VO_SchemaName and VO_OutcomeSchema if outcome is available



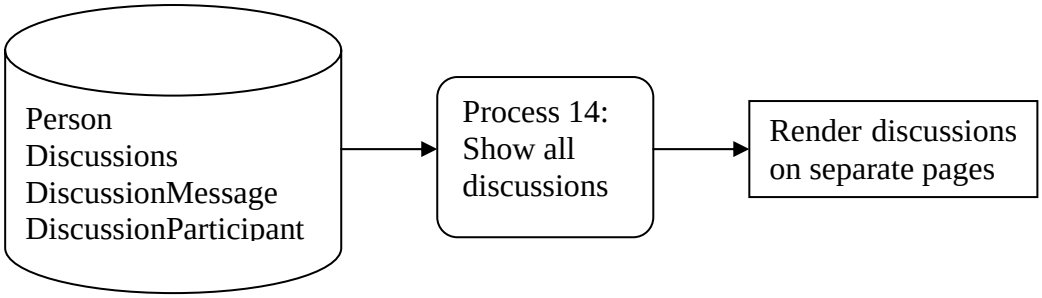
Process 13: Create New Discussion [forums, creatediscussions]

1. Populate lists of people and papers from Person, Worksat and TrialPreprint tables.
2. Upon user's choice of creating a new discussion, update discussion related DB tables with topic, owner, member and paper information.
3. Display information about the discussion on a new page



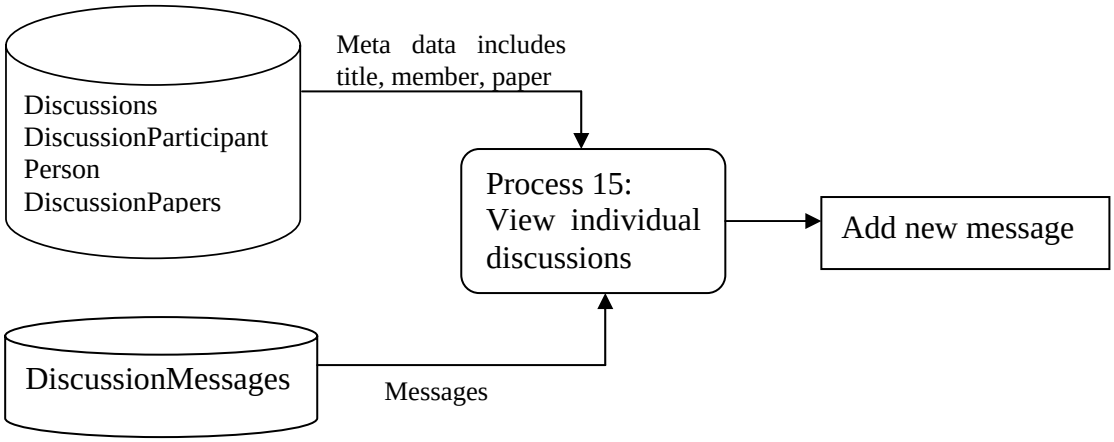
Process 14: Show All Discussions [forums.aspx]

1. Retrieve discussion related information (title, creation date, and people involved etc) from DB tables Person, Discussions, DiscussionMessage, DiscussionParticipant.
2. Display each discussion as a record with relevant information retrieved.
3. Render discussions over separate pages



Process 15: View Individual Discussion [discussion.aspx]

1. Display the discussion title based on id from table Discussions
2. Check the user permission to view the discussion (DiscussionParticipants)
3. Display member and paper information under stretchable links (from person, DiscussionParticipants, DiscussionPapers)
4. Display all the messages under the same discussion (DiscussionsMessages)
5. Add new message into DB table DiscussionsMessages upon users' submission



The above working processes and databases are tightly integrated in VOEU, so it is difficult to expand as the user requirements change. CORE is taking the modules and developing them into Web services and supporting these services in a SOA where flexible granular functional components expose service behaviours accessible to other applications via loosely coupled standards-based interfaces. The current design of CORE SOA is described in the later section.

2.2 Databases in DRJ

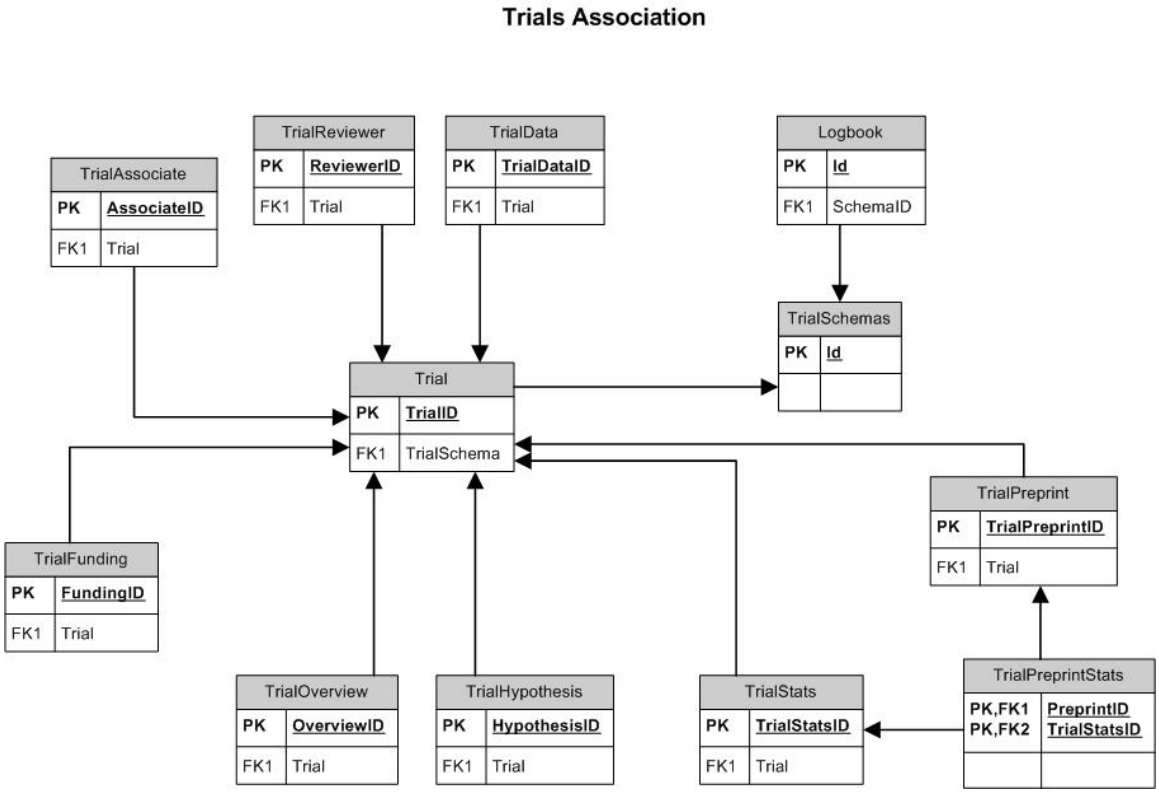


Figure 1 Trials database ER diagram

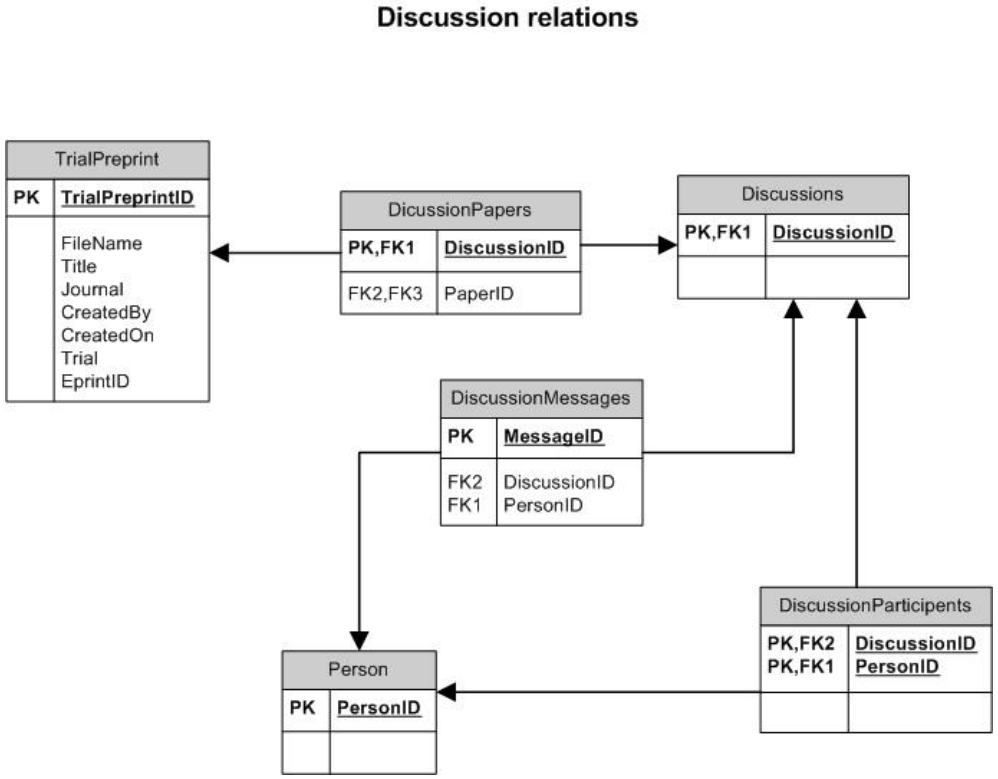


Figure 2 Discussion database ER diagram

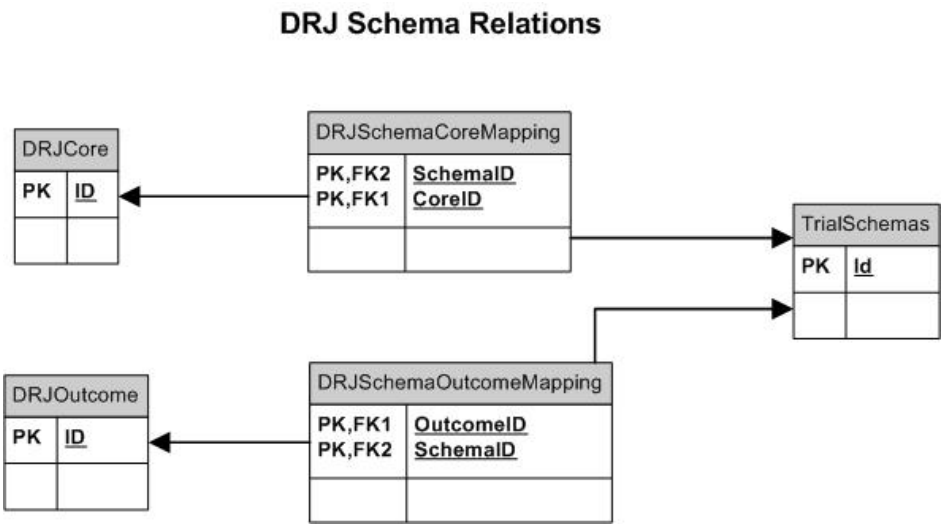


Figure 3 DRJ Schema ER diagram

In the tightly coupled DRJ system, the database tables are closely related in a way that each process updates a number of relevant database tables when it is being triggered, especially the tables holding user profiles and status. The major databases involved in DRJ are those storing trials, schema, and case information. Their relationships are illustrated in the ER diagrams above (Figure1, Figure 2, Figure 3). From the table in Appendix 1, it is easy to identify how the database tables are closely linked with each working process. It is our aim to break up the rigid associations in the current work flow and redesign them into a system with pluggable components.

3 Integrating CORE SOA

The CORE project is being implemented as a toolkit of generic components. CORE architecture has been designed using the concept of portal and SOA (Figure 4).

On the server side, a portal will be used to facilitate the sharing of research resources in the CORE VRE. Portal is defined as an online facility that provides a personalised, single point of access to resources that supports the end user in one or more tasks (i.e. discovery, learning and research) [5]. The main purpose of a portal in CORE infrastructure is to act as a presentation layer which aggregates, integrates, personalises and presents information, transactions and applications to the user according to their role and preferences. As portal mainly acts as a gateway between clients and a range of services/components, portlet is the technology that provides specific service/component within a portal. In the context of CORE, several components are designed as portlets, including Trial Manager, User Manager, Authentication, and so on. As a key component in the portal, Trial Manger and its work flow are explained in depth in the following section.

In order to achieve certain tasks, the portal needs to invoke some grid/web services as depicted in Figure 4. These services are designed as reusable components, being decoupled from the VOEU system. By using these services, CORE users can perform tasks such as formalising trial protocol, storing and analysing data, submitting and

reviewing articles, and discussing research findings in a forum. Another Grid service, named *Grid Simulation* is also included in the VRE infrastructure. It will provide users with functionalities, such as job submission, file transfer, and credential management, in running their simulations. The portal needs a middleware called OMII so that end users of the CORE toolkit can access Grid / Web Services in a trusted and secure environment.

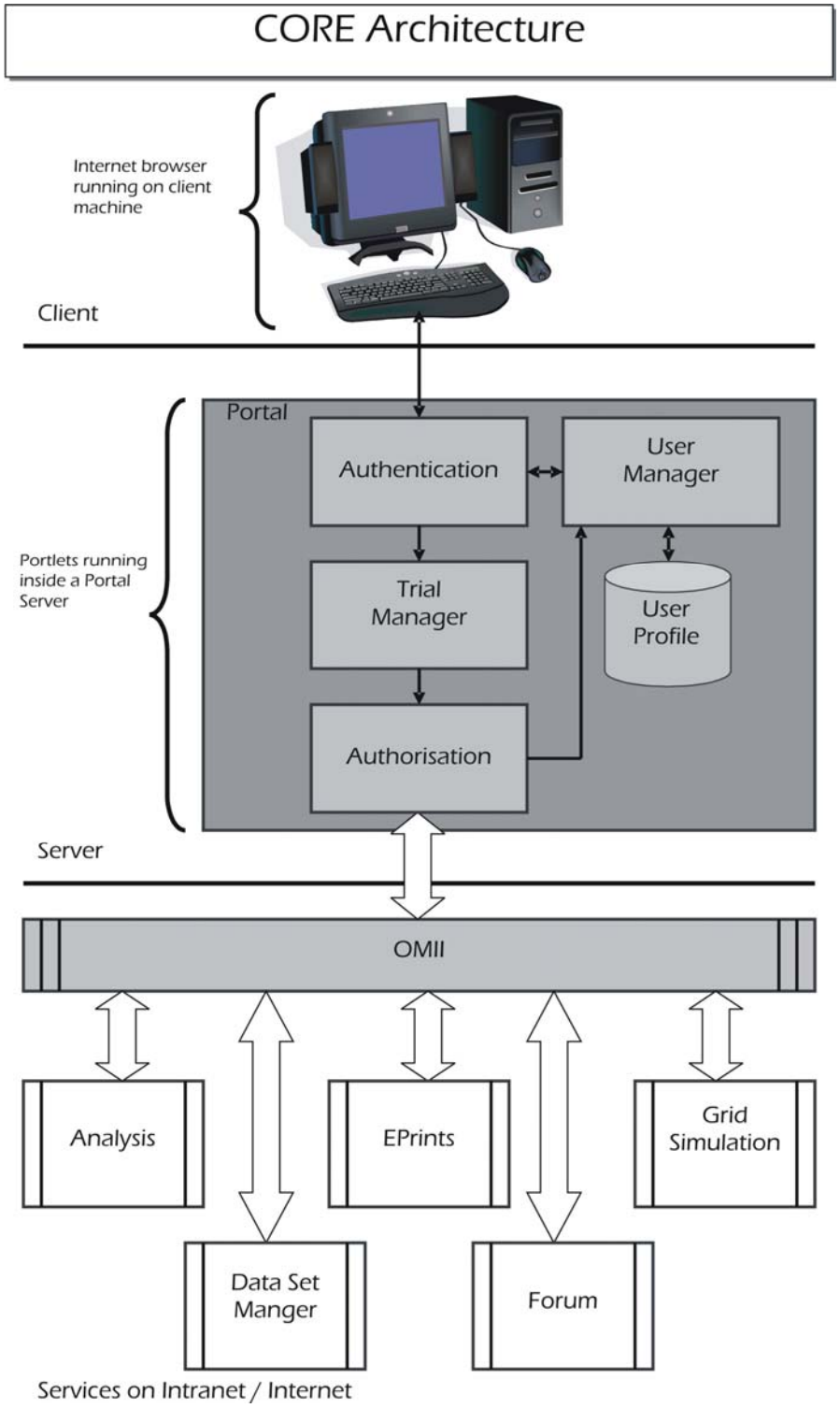


Figure 4 CORE Architecture

3.1 Trial Manager

Within the portal architecture, Trial Manager plays a vital role as it controls the work flows for setting up experiments and submitting papers. Similar to those processes in VOEU, Trial Manager performs a set of tasks as described below. From the user point of view, it might have similar functionalities as VOEU from the interface level. But behind the scene, it works in a different way compared to VOEU which provides more flexibility and personalisation. The Trial Manager will carry out these processes by using the web services located on different servers. There are four web services identified and designed for CORE: *Analysis, Data Set Manager, EPrints and Forum.*

1. Create new trial

This is the first step of research collaboration based on trials. A new trial needs to be set up to enable the further tasks such as searching for cases, analysing data and creating preprint based on trials. This process requests a set of current trial schemas from *Data Set Manger* and presents a list to users. It subsequently updates a trial database within Trial Manger with trial metadata including identifier, data, investigator, schema, etc.

2. Update trial

This process is to update relevant data for an existing trial. It requests current schema list from *Data Set Manager* and current data from Trial database. Then it updates Trial database with users' entry. It also accesses *Forum* service to update participants' information associated with discussions.

3. Add data

This process allows users to enter data based on a certain schema. It retrieves schema sets and associated variables (core and outcome) from *Data Set Manager*. Then it updates Virtual Observatory (VO) database in *Data Set Manager* with the user entries. The process can occur at any point.

3. a. View data

This process can be carried out after the add data process. It obtains data requested from the VO database in *Data Set Manager.*

4. Search for case

The process searches for data in the VO based on schema type and user-supplied criteria. It needs to access VO databases to retrieve relevant data sets which match the search conditions.

4. a. View cases

The process displays cases selected from the search result (in process 4). It retrieves data from VO database in *Data Set Manager* and a TrialData database (holding selected cases) within Trial Manager.

5. Analyse data

This data analysis process can be performed after cases are selected. It retrieves data from VO in *Data Set Manager* and TrialData database. Then the process generates queries and sends to *Analysis* service. *Analysis* service produces a result file and TrialStats database is updated.

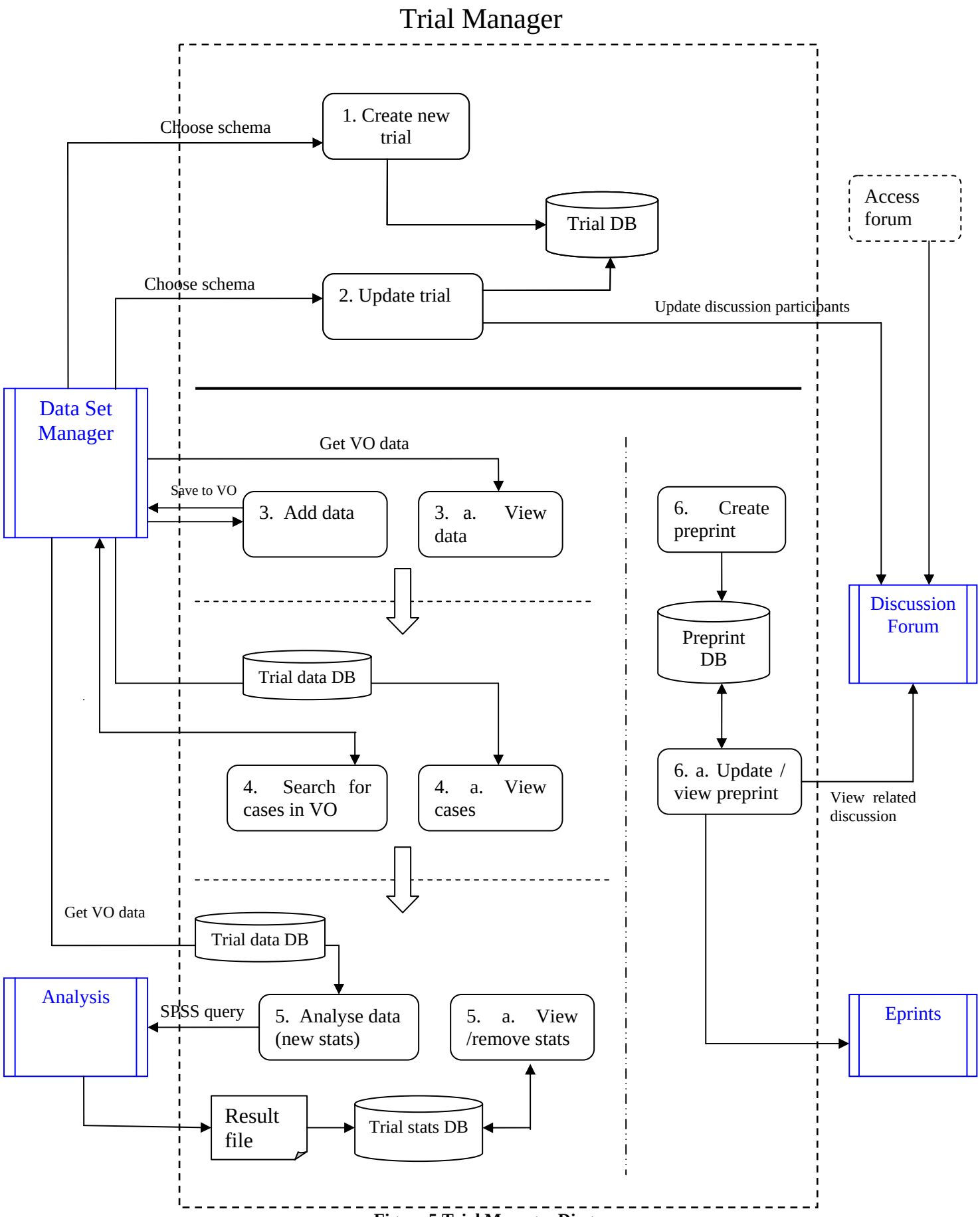


Figure 5 Trial Manager Diagram

5. a. View stats
Once any analysis is created, the process of View stats can be carried out. The process accesses the TrialStats and retrieves the file name for analysis result. Then the analysis can be viewed and removed based on users' privilege.

6. Create preprint
After a trial is created, a new preprint can be made associated with a specific trial. A preprint xml file will be generated and TrialPreprint and TrialPreprintStats will be updated subsequently.

6. a. Update/view preprints
For any current preprint in the system, it can be viewed and updated by relevant users. The process accesses TrialPreprint database and retrieves/updates information (metadata and content) from/to the xml file. It also accesses *Forum* service to retrieve related discussions associated with the preprint. Finally the preprint can be submitted to *EPrints* service for wider communities.

The numbering of the above processes doesn't indicate the order in which the processes should occur. For example, process 6 – *Create Preprint* can occur at any time as long as a trial is created; process 3 *Add Data* could occur at any point when users want to add new data sets; a user can also access forum at any time. But the hollow arrow does suggest the sequence of some of the processes. For instance, a user can only obtain a search result after data is added into databases; a new data statistics is generated based on the data set selected from a search result.

Trial Manager conducts the above processes by making use of the functions from the web services. Trial Manager is mainly in charge of the overall work flow and interface while web services handle the actual implementation of most functionalities. Among these web services, *Data Set Manager* is slightly complicated and consists of several components. It is described in more detail in the next section.

3.2 Data Set Manager

Data Set Manager comprises two types of data and their associated work flow. One type is the trial or data template information, which is the formal specification of trial procedure or the nature of the trial data; the other type is the data held in Virtual Observatory (VO), which are the actual case records/data under different templates.

As illustrated in Figure 6, template and VO data are stored in databases. A template organizer and a data organizer handle all the processes related to template and data respectively. The general processes are described as below:

- Add Template:
Generate new template based on user specification or current template; new template is stored in template database with a unique identifier. A new table under the name of the newly-generated template is created in VO database in order to input data records for this template.

- Delete Template

Select a template and delete it from the database; although the record is just marked as deleted, not actually removed from the database. The relevant records in VO database are also labelled as deleted.

- Modify Template

Select a template and modify its content; then the altered template is stored in the database under the same identifier. The VO data under the previous template is to be migrated into the current template (could result in problems in database?)

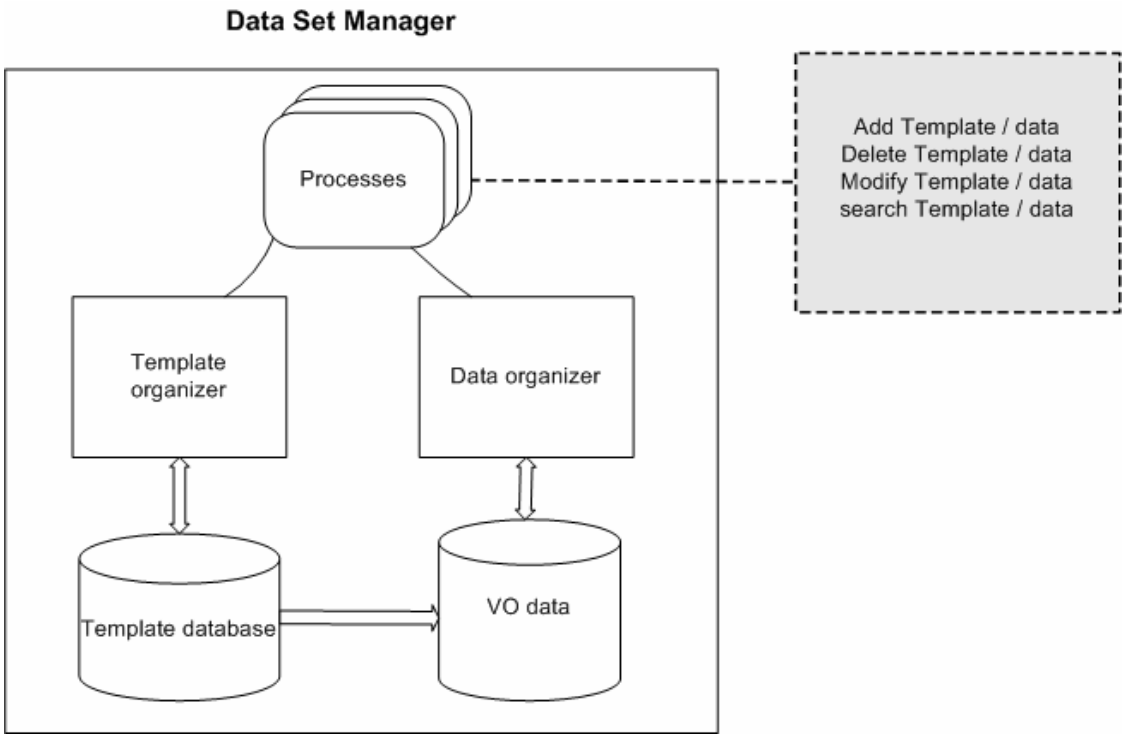


Figure 6 Data Set Manager Diagram

- Add data

Choose a data template and enter data accordingly; the new data record is stored in the table identified by the template name in the VO database. The data will be available to users for further analysis or search tasks.

- Delete data

Select a data record and delete it; the data is still kept in database but labelled as "deleted". So it won't be available for search or analysis.

- Modify data

Select a data record and update relevant data fields; save the amended record in the database.

- Search data

Generate a search form based on a template; according to user supplied searching criteria, look for matching data records under a particular template/schema in VO database; return the records to user.

As designed, not every user has the permission to carry out each of the above tasks. For example, only a data creator or a user in the same project or an administrator has the privilege to delete a data record; other users have restricted right to view and search data only.

4 Conclusion

The CORE VRE described in this report is to be implemented as a Grid/Web-based environment for supporting a critical subset of the e-science cycle: the collation and analysis of experimental results, the organisation of internal project discussions, and the production of appropriate outline documents depending upon the requirements of conferences and journals selected for dissemination [2]. The system provides a distributed loose-coupled architecture to accomplish the tasks in e-research for medical scientists. The aim of this project is to provide a toolkit which is generic, applicable across surgical and medical training.

With the focus on this specific VRE framework, the report describes the design of the CORE system. It starts from the previous VOEU system and decouple its working processes. Then it uses SOA and Grid services to redesign the architecture and work flows in a new concept. At the time of the report is being written, the main components in the CORE system have been designed and are to be implemented. However, the authors realize that user requirements might change over time; therefore a VRE must be designed to evolve in accordance with the changing needs of its users.

Reference

1. Collaboration Orthopaedics Research Environment (CORE) Project Proposal, University of Southampton, Information available from: http://www.core.ecs.soton.ac.uk
2. Carr, L., Miles-Board, T., Wills, G., Power, G., Bailey, C., Hall, W., and Grange, S. (2004) Extending the Role of Digital Library: Computer Support for Creating Articles. In *Proceedings of the Fifteenth ACM Conference on Hypertext and Hypermedia*, University of California, Santa Cruz, USA, pp. 12-21.
3. Grange, S., Wills, G., Power, G., Miles Board, T., Carr, L. and Hall, W. (2003) Building a dynamic review journal (DRJ) - Extending the role of the Virtual Orthopaedic University. 3rd Annual Meeting of the International Society for Computer Assisted Surgery (CAOS International), Marbella, Spain June 18-21, pp 122-123.
4. Smythe, C., Evdemon, J., Sim, S., & Thorne, S. (2004). Basic Architectural Principles for Learning Technology Systems. from http://www.imsglobal.org/
5. Allan, R., Awre, C., Baker, M. and Fish, A. (2004). Portals and Portlets 2003. Available from National e-Science Centre, UK. http://www.nesc.ac.uk/technical_papers/UKeS-2004-06.pdf

Appendix 1 Processes and databases relationship table

Processes / Tables	Create new trial	Update trial	Search for cases	View cases	Analyse data	View/update stats	Create preprint	View preprint	Update preprint	Add data	View data	Generate new template	Create new Discussion	Show all discussions	View individual discussion
Trial schema	√	√	√	√						√					
VO			√	√	√					√	√				
Person	√					√		√					√	√	√
Trial	√	√													
TrialData				√	√										
TrialAssociate	√	√													
TrialReviewer	√	√													
TrialHypo, TrialOverview	√														
Trialpreprints		√					√	√	√				√		
TrialpreprintStats						√	√		√						
TrialStatistics					√	√									
TrialJournal								√							
DiscussionPaper		√						√					√		√
DiscussionParticipants		√											√	√	√
Discussions													√	√	√
DiscussionMessage														√	√
DRJ					√					√					
Logbook										√					
Template												√			
WorkAt													√		