Deliverable D3.1

Architecture and Design of the ContraCancrum Technical Environment

Project Number: FP7--IST-*223979*
Deliverable id: D 3.1
Deliverable name: Architecture and design of the ContraCancrum Technical Environment
Submission Date: 17/07/2009

Information Society Technologies

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	CONTRA CANCRUM
Project Full Name:	Clinically Oriented Translational Cancer Multilevel Modelling
Document id:	D3.1
Document name:	Architecture and Design of the ContraCancrum technical environment
Document type (PU, INT, RE)	RE
Version:	5
Submission date:	17/07/2009
Editor: Organisation: Email:	Steven Manos University College London s.manos@ucl.ac.uk

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

ABSTRACT:

This deliverable provides a high-level overview of the technical environment to be developed within ContraCancrum. The technical environment is ultimately intended to provide a multi-scale clinical decision making environment. Available technologies for storing various types of clinical data - patient, genomic, microarray, imaging as well as simulation data - are discussed, as well as work flow tools based on high-performance computing grid-infrastructure and web service interfaces.

KEYWORD LIST: Grid infrastructure, high-performance computing, medical imaging databases, oncology simulation, web services

MODIFICATION CONTROL			
Version	Date	Status	Author
1.0	1/5/2009	Template	K. Marias
2.0	6/5/2009	Draft	G. Clapworthy
3.0	26/6/2009	Draft	S. Manos
4.0	15/07/2009	Final	S. Manos

List of Contributors

- Steven Manos, UCL
- Stefan Zasada, UCL
- Gordon Clapworthy, BED
- Peter V. Coveney, UCL
- Xia Zhao, BED
- Tao Wang, BED
- Enjie Liu, BED
- Norbert Graf - USAAR
- Amos Folarin, UCL
- Konstantinos Maris, FORTH-ICS

Table of contents

1 Introduction

The ContraCancrum, i.e. the *Clinically Oriented Translational Cancer Multilevel Modelling*, project aims at developing a composite multilevel platform for simulating malignant tumour development and tumour and normal tissue response to therapeutic modalities and treatment schedules.

The project aims at having an impact primarily in providing (a) a better understanding of the natural phenomenon of cancer at different levels of biocomplexity and, most importantly, (b) a disease treatment optimization procedure in the patient's individualized context by simulating the response to various therapeutic regimens. The predictions of the simulators to be developed will rely on the imaging, histopathological, molecular and clinical data of the patient. Fundamental biological mechanisms involved in tumour development and tumour and normal tissue treatment response such as metabolism, cell cycle, tissue mechanics, cell survival following treatment etc. will be modelled. Stem cells will be addressed in the context of both tumour and normal tissue behaviour. From a mathematical point of view, the simulators will exploit several discrete and continuous mathematics methods such as cellular automata, Monte Carlo methods, finite elements, differential equations, novel dedicated algorithms and so on. A study of the analogies of tumour growth with embryological development is expected to provide insights into both mechanisms.

ContraCancrum endeavours to have two important outcomes. Firstly, the project will deploy two important clinical studies for validating the models, one on lung cancer and one on gliomas. The crucial validation work will be based on comparing the multi-level therapy simulation predictions with actual medical data (including medical images), acquired before and after therapy. Secondly, the project will provide a common technical environment where clinicians and researchers can view, study and manipulate patient-specific multi-level data - from genomics and molecular data up to imaging data of organs and tissues - to provide a *multi-scale decision support platform for oncology*. These two outcomes will pave the way for translating clinically validated multilevel cancer models into clinical practice.

This deliverable builds on deliverables D2.1 and D2.3, which identified and characterised the anticipated user population of the ContraCancrum integrated technological platform and the uses to which it will be put to produce a broad specification for the system. It views the system from the perspective of the individual workpackages and identifies needs that they anticipate having in the course of the project.

This document outlines the technical environment which will give clinicians and researchers access to the data generated throughout the project, integrating traditional patient medical record data and patient-specific medical simulation data into the same interface. It should be read alongside Deliverable D2.3, which deals with the requirements and specification for the ContraCancrum integrated platform and clinical validation studies.

Relationship to the VPH NoE, FP6 and FP7 projects

To help ensure interoperability on data and application levels across the VPH research sphere ContraCancrum will maintain a close relationship with NoE WP3 ToolKit members. Furthermore, ContraCancrum has established an ongoing collaboration with the FP6 ACGT[1] project.

[1] ACGT - Advancing Clinico Genomic Trials on Cancer - http://www.eu-acgt.org

2 Proposed solutions for computational and data requirements

The technical environment will firstly provide a central *integrated* repository of patient data for project clinicians and researchers. This data includes patient, clinical, histopathological, radiological medical imaging, genomic, microarray and simulation data. The storage of results from simulation is also central to this multi-scale patient-specific computational approach, where data from image analysis and segmentation (regions of interest, annotations, meshes), molecular dynamics and cell biomechanics simulations need to be stored.

Beyond a straightforward repository, many other factors need to be considered to ultimately provide a simulation and data environment for the integration of multi-scale data and a decision support environment for clinicians and research data environment for researchers. Simple intuitive interfaces for data collection, editing and display need to be developed. For the use of computational resources within such an environment, we need to consider appropriate access to resources (such as DEISA (EU) and the TeraGrid (USA)), user-settable reservations for clinical use and dedicated bandwidth allocations for the rapid transfer of simulation data. Wherever possible we plan to draw on existing solutions that have been developed within FP6, FP7 and other projects, as well as recognised standards and protocols, and provide the necessary APIs for (automated) insertion and extraction of data from simulation work flows (e.g. Imaging data via DICOM protocol & HTTP).

Patient Data

2.1.1 Pseudo-anonymisation of patient data

The requirements on patient clinical data are described in reports D2.2 and D9.1. All personal data will be pseudo-anonymised before being placed in the ContraCancrum technical environment. The patient data will be related to the hospital data via a (secret) table that is maintained by the clinicians. Only the treating physician will have the right to see personal data. It is up to the project clinicians to ensure that the correct pseudo-key is used for different types of patient data from within different hospital departments (pathology, imaging, central records, etc.).

2.1.2 Clinical data

ContraCancrum will make use of the ACGT[2] project Ontology[3] to store patient data and clinical data pertaining to patient presentation condition as well as treatment regimes. Using such standardised data has the advantage that data generated within ContraCancrum is fundamentally compatible with data from projects such as ACGT, as well as future projects that use this standardised ontology.

2.1.3 Radiological imaging data

Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in radiological medical imaging. The standard defines a file format as well as a network communications protocol (which uses TCP/IP). Medical imaging data acquired by the clinical partners are stored on centralised PACS (Picture Archiving and Communication Systems) servers at the University of Saarland, and this data must be made available to researchers within the project.

PACS and DICOM software used in the clinic is generally proprietary, however there are high-quality open-source solutions available. Here we will use *dcm4che* (http://www.dcm4che.org) to provide a central imaging repository (hosted at UCL). It is a robust open source collection of Java tools implementing the DICOM standard, at the core of *which* is *dcm4chee*, which is the PACS image manager and image archive element. Members of the project who are using DICOM viewers (such as OsiriX[4] on MAC OS X and ClearCanvas[5] on MS Windows) will be able to directly connect to this server securely using SSH tunnels.

---

[2] ACGT - Advancing Clinico Genomic Trials on Cancer - http://www.eu-acgt.org

[3] Brochhausen, M. *et. al.*, The ACGT Master Ontology on Cancer - A New Terminology Source for Oncological Practice. In Proceedings of the 2008 21st IEEE Int. Symposium on Computer-Based Medical Sys. - (June 17 - 19, 2008). pp. 324-329. DOI= http://dx.doi.org/10.1109/CBMS.2008.17

[4] OsiriX DICOM viewer - http://www.osirix-viewer.com

[5] ClearCanvas DICOM Viewer - http://www.clearcanvas.ca/dnn

In order to provide web-based access to DICOM we will use WADO, which is implemented by dcm4che. The WADO[6] standard specifies a web-based service for accessing and presenting DICOM persistent objects, providing a simple mechanism for accessing a DICOM persistent object from HTML pages or XML documents through HTTP/HTTPS, using DICOM UIDs. DICOM data then becomes available over the web in JPEG format or native DICOM format. This will be built into the web interface for the project.

2.1.4 Genomic, protein sequence and protein structure data

Genomic sequence data will be primarily used in WP5 for the patient-specific drug binding affinity simulations. Genomic assays of the sequences that code for the proteins of interest from patients will be acquired and appropriately pseudo-anonymised before insertion into the database. Gene sequence data are commonly held in flat file formats[7] such as EMBL, GenBank and FASTA formats. Each format differs in its human-readability and the specification of the region coded for. There are various scripts available for conversion between these different file formats.

Protein sequence data is also similarly held in a variety of flat file formats[8]. We intend to primarily use PDB formatted files for protein-structure data.

Metadata associated with the file (upload date, unique patient pseudo-key, etc.) will be stored in a central database (see Section 2.1.7 for more details).

2.1.5 Microarray data

D5.2 will focus on producing a model for predicting cell-survival probabilities in response to radio and chemotherapy regimes using gene expression profiles from patient microarray data.

The data used for WP5.1 consists of expression profiles (assay) and covariates (drug or radiation sensitivity profiles). Information on levels of gene expression is given by a microarray assay. Each microarray dataset consists of RMA normalised Affymetrix platform data HGU133 (A/B/plus2) which is formatted as a flat tab delimited file of rows: gene (ref: probe ids) and columns: samples (ref: sample ids). Information on drug/radiation sensitivity is given by drug screen data for the corresponding samples. Each covariate dataset consists of a flat tab delimited file of rows: samples (ref: sample ids) and columns: covariates (ref: covariate ids).

Metadata associated with the file (upload date, unique patient pseudo-key, etc.) will be stored in a central database (see Section 2.1.7 for more details).

Simulation Data

2.1.6 Molecular dynamics

In the short term, files related to genome sequences, protein structure files and other simulation data will be stored on a dedicated storage array along with a PostGreSQL database which will host the file metadata (outlined in Table 1). In the long-term we propose to use iRODS[9] for the storage, retrieval and archiving of simulation data.

2.1.7 Biomechanics

The model used for the biomechanical simulation needs to be stored in a file once for each patient. This file includes the finite element mesh, mechanical properties and boundary conditions. Data will be stored in a text format, but can also be translated to an XML format if required. Each input file will be around 4 MB.

Regarding storage of the results of the simulation, the amount of space used by the WP6 will be strongly dependent on the simulation strategy chosen for the cellular modelling. As an initial estimate, the scans of the cellular level simulation take place every hour. A typical treatment course takes as long as two months. This implies about 1344 mesh scans per patient and per set of parameters used for the cellular simulation. Therefore, the amount of data that needs to be stored for the biomechanical simulation could reach up to 80 GB (binary files) for one patient and one set of parameters. In case the number of cellular simulations grows to very large numbers, simple strategies could

---

[6] WADO (Web Access to DICOM Persistent Object) standard - http://medical.nema.org/dicom/2004/04_18PU.PDF

[7] http://www.genomatix.de/online_help/help/sequence_formats.html

[8] http://www.ebi.ac.uk/help/formats.html

[9] iRODS:Data Grids, Digital Libraries, Persistent Archives, and Real-time Data Systems - https://www.irods.org

be used to decrease this amount of data. It is possible, for example, to store result information every *n* increments and/or to store the mechanical information in and around the tumor much more frequently than for distant tissues.

In order to build the biomechanical simulations, WP6 needs to access the medical images as well as the image segmentation. The data are collected by the clinician and processed by WP7. These data have to be stored in specific databases, which have to be accessible to our software to query the information for a specific patient. This connection will only occur once per patient. Once the model has been built, data exchange will be required with the cellular module. For this purpose, direct communication between the two modules will be developed allowing access to the required databases.

The model files will be initially stored in as text format. A file system database will be developed to store the detailed file access information such as file URL. This database will also have a relationship with the patient database, which will make the model files searchable for a specific patient. The proposed file database schema is shown in Table 1.

A text-to-XML translator will be offered to allow users to translate models from text format to XML format. This function will be web services compliant, which allow the translator to be invoked by different applications.

The biomechanical simulation results will be stored as binary files in the data storage and will be searchable via patient and parameter set information. Similarly, these data will be accessed from file system database. UCL will host a large data array, initially 7TB in size, to support the necessary data storage requirements.

2.1.8 Image processing

The storage needs for WP7 (tumour imaging and visualisation) are strongly dependent on the number of clinical images to be processed. An estimate of the requirements is 60 patients, 5 studies per patient with approximately 20 images per study and 0.5 MB per image (3 GB in total).

Registration results of this radiology data can be stored either as transformation matrices (rigid registration) or as point sets (non-rigid registration). In both cases, the storage needed is much less than for the original images. Segmentation results can be stored either as surface meshes or as label images. Label images typically have a bit depth smaller than the original images. Therefore, on the DICOM server, there should be storage allowed for twice the size of the original clinical images. On the database for the non-image data, storage for image processing results should be allowed equivalent to the size of the original clinical images.

The DICOM server for clinical images and processed images will use the DICOM protocol for data transfer. Whenever non-DICOM data must be stored, this should be done in the database for non-image data. The VTK[10] native format is the preferred file format for surface meshes, rigid and non-rigid transformations.

In order to ensure the quality of the processing, information on the processing steps must be stored in the database. For each image processing result stored in the database, it must have stored with it an indication of which original data has been used to create the result. Since image-processing results depend on the tools used, information on which tools have been used to create the result and which person has done it should also be stored. Details of the schema used are outlined in Table 2. The image segmentation results will be made available to WP6 (Simulating cancer biomechanics) as initial inputs for biomechanical simulation at the tissue level.

Further details of requirements can be found in delivery report D2.3 - *Requirements and specifications for the ContraCancrum integrated platform and clinical validation studies.*

Table1. File database schema

Table	Column	Datatype	Description
file	file_id	INTEGER	The primary key
	patient_id	INTEGER	The foreign key reference to patient
	file_path	VARCHAR	Path to the file including file name
	file_size	BIGINT	Size of the file in bytes
	created_time	DATETIME	When the record was created in the DB

[10] VTK - Visualization ToolKit - http://www.vtk.org

Table 2. Non-image database schema

Table	Column	Datatype	Description
Non-image data	file_id	INTEGER	The primary key
	fk_data_source	INTEGER	The foreign key reference to original data
	fk_tool	INTEGER	The foreign key reference to tool applied
	fk_creator	INTEGER	The foreign key reference to creator
	file_path	VARCHAR	Path to the file including file name
	file_size	BIGINT	Size of the file in bytes
	created_time	DATETIME	When the record was created in the DB

2.1.9 Cellular and higher-level data

The data requirements for tumour growth simulation (WP4/ICCS) will be typically large (GBs per simulation). In order to help eliminate errors, each geometrical cell of the discretisation mesh should contain a *single* biological cell. This implies that, for a large tumour and adjacent anatomical region of interest of dimensions 100 mm x 100 mm x 100 mm = 10^6 mm^3, we would need: 10^6 biological cells included in each mm^3 x 10^6 mm^3 = 10^{12} biological cells = 10^{12} geometrical cells. Each biological cell (corresponding to a single geometrical cell) would be described by at least the following parameters, which would need to be stored:

- Metabolic layer (well vascularised tumour region, poorly vascularised tumour region, edema, cyst, adjacent normal tissue of type "1", adjacent normal tissue of type "2" etc.) – type: character
- Cell category [stem, progenitor (of stage 1, stage 2, stage 3, stage 4, stage 5), differentiated, apoptotic, necrotic cell] - type: character
- Phase within or out of the cell cycle (e.g. G1, S, G2, M, G0, ...) – type: character
- Treatment 1 hit or non hit cell – type: Boolean [Note: treatment of type "1" may denote a particular drug or radiotherapy]
- Treatment 2 hit or non hit cell – type: Boolean
- Time spent within the current phase – type: integer

Initially a cell database will be provided to store biological cell information. A proposed database schema is shown in Table 3. A more realistic scenario is listed in Appendix A and fast memory management mechanisms will be investigated.

Table 3. Cell database schema

Table	Column	Datatype	Description
Cell	cell_id	INTEGER	The primary key
	metabolic_layer	VARCHAR	IN ('well vascularized tumour region', 'poorly vascularized tumour region', 'edema', 'cyst', 'adjacent normal tissue of type 1', 'adjacent normal tissue of type 2')
	cell_category	VARCHAR	IN ('stem', 'progenitor', 'differentiated', 'apoptotic', 'necrotic cell')
	phase	VARCHAR	IN ('G1', 'S', 'G2', 'M', 'G0', 'N', 'A')
	treatment1	BOOLEAN	Hit or non hit cell
	treatment2	BOOLEAN	Hit or non hit cell
	time_spent	INTEGER	Time spent with current phase

Near-term and long-term proposed solutions

In the near term the aforementioned data schemas and repositories will be hosted at UCL.

In the long term we propose to use the data hosting environment (DHE) which is being developed within the VPH NoE WP3 ToolKit. The DHE looks beyond the short term data hosting needs of the NoE and proposes an infrastructure that is VPH-compliant, and offers a clear development path to full VPH implementation. However, as a particular infrastructure implementation, it necessarily chooses to resolve numerous architectural issues in a particular manner, and this has certain philosophical implications for its implementation. One example is data centralisation; should the VPH infrastructure be based around a centralised data warehouse model or is it more appropriate to be based around a data mediation service? The former involves a centralised, curated data repository, whilst the latter favours distributed archives and provides a means to access them. At its core, the DHE is an implementation of the distributed solution, although elements of the centralised philosophy are integrated where necessary. For the purposes of the VPH, the distributed model is considered to be a more robust solution (no single point of failure; no storage capacity limit; emphasises that data ownership resides with the data provider and all that that entails in terms of looking after the data; costs for data maintenance are naturally distributed across the data providers; data users operate in an environment that consists of live copies of the data; inherently extensible; etc.). The proposed implementation is framed around the response to four questions that are central to interaction with the VPH, namely:

- VPH data - Where is it?
- VPH data - What kind of data is it?
- VPH data - Can I use it?
- VPH data - How can I contribute my data to it?

The first item is an exercise in searching and cataloguing (e.g. data content/formats), with cataloguing being particularly relevant to the second item on the list (description of the data - ontologies, metadata, etc.). The third issue involves practicalities such as data formats (can the data be read by my particular application?) and authorisation (am I permitted to use the data in the way I want to use it? - security, regulatory control, ethics, copyright, validation, etc.), whilst the final question is concerned with transfer protocols, storage, curation, integration, etc. The solutions to these issues as characterised by the DHE, is provided by support for three fundamental operations that define the architecture:

i. Expose
ii. Query
iii. Download

(i) Expose - Reflects a conscious choice by the data provider to expose a selection of his/her data to the biomedical community via the VPH. This requires a software environment configured to accept and place the data in the VPH domain, supported by a host of infrastructure-related items that include secure transfer protocols, user authentication and authorisation, and facilities for cataloguing the data as a metadata file that is registered with a central data controller.

(ii) Query - Permits searching of the metadata registry (held by the central data controller) as a means of exploring the data content within the VPH domain. This step is necessary if the source of the metadata of interest is to be identified and a link provided for subsequent download.

(iii) Download - The operation by which a user obtains VPH data (and naturally includes aspects such as secure transfer protocols, user authentication and authorisation, etc.)

In this model, the VPH participants wishing to share data (and indeed models) will expose their information to a central VPH controller/registry (a server) for cataloguing (as metadata), but the data itself will remain on the users' own machines. Particular tools are needed to implement such a strategy, to support client communication with the central controller in a robust and secure manner. A variety of VPH services will be run automatically on the exposed data, carrying out a range of categorisation and validation activities intended to result in the controller being able to catalogue and offer the data to the remainder of the VPH community. Download is a simple operation in which the address of the requested data is passed to the requesting user, leading to subsequent file transfer.

Such an implementation would provide a generic data access mechanism within ContraCancrum which would also be standards compliant and compatible with other projects also using the DHE.

3 Workflow requirements and proposed solutions

Workflows will be used extensively within the ContraCancrum project. Usability of the technical environment - particularly from the viewpoint of clinicians - necessitates the use of distributed workflows that hide many of the details in relocating data files, running simulations (on local resources or HPC-scale resources) as well as generating and reporting results.

3.1.1 Molecular dynamics

The workflow typically used in molecular dynamics binding affinity simulations look like this:

1. Acquisition of raw x-ray structures (PDB file format) from protein data bank.
2. Manual cleaning of x-ray files: removal of duplicated sidechains and unwanted residues.
3. Building of protein loops if they are missing in the x-ray structure. Missing regions are usually very flexible and their electron densities can't be defined.
4. Generate force field parameters and topology for inhibitors using Gaussian[11] (using part of the BAC - *Binding Affinity Calculator* - on local resources).
5. Generate the protein-inhibitor complex, add counterions and water molecules; generate script files for job submission (using BAC on local resources).
6. Copy the structural files and job submission scripts (from step 5 above) to DEISA or TeraGrid HPC resources and manually submit jobs (or automatically from within BAC). The NAMD[12] package is used for simulation.
7. Atom trajectory files (from step 6, typically files large) are copied back to the local machine
8. Preparing for post-process analysis using BAC (on local machine):
   8.1 Generate topology file for protein-inhibitor (remove water, counterions)
   8.2 Generate scripts for binding free energy calculation
   8.3 Convert Charmm format trajectory (DCD format file) to Amber format trajectory (TRAJ format file)
9. Transfer files from step 8 to UK NGS[13] (located at the University of Leeds or Rutherford Appleton Laboratory (RAL)) resources.
10. Run scripts for the MMPBSA and NMODE stages on Leeds/RAL using manual job submission. Amber[14] is used for post-processing analysis
11. When finished, transfer back to local machine.

In order to automate this workflow of simulation execution and data movement between remote and local computational resources, the GridSpace Engine (GSEngine[15]) developed in the FP6 ViroLab[16] project will be used along with the Application Hosting Environment[17]. Work is already underway in establishing the workflow tools such as the Binding Affinity Calculator on DEISA grid resources.

3.1.2 Microarray analysis

Using microarray assays from a patient, an estimate can be derived for the cell survival probability under certain radiotherapeutic and chemotherapeutic conditions. A flat file that contains the microarray assay data can be uploaded to the online decision support system via a web interface. The R[18] statistical analysis package is used for processing of the array data. Here we plan to use the Rweb[19] package for integration in the online environment.

If storing large amounts of microarray data becomes a necessity within the project, we will consider open source microarray database environments such as BASE[20].

---

[11] Gaussian - http://www.gaussian.com

[12] NAMD - Scalable Molecular Dynamics - http://www.ks.uiuc.edu/Research/namd

[13] UK NGS - National Grid Service - http://www.grid-support.ac.uk

[14] Amber molecular dynamics package - http://ambermd.org

[15] GSEngine - http://virolab.cyfronet.pl/trac/vlruntime

[16] FP6 ViroLab - http://www.virolab.org

[17] Application Hosting Environment - http://www.realitygrid.org/AHE

[18] The R Project for Statistical Computing - http://www.r-project.org

[19] Rweb - http://www.math.montana.edu/Rweb

[20] BASE - BioArray Software Environment - http://base.thep.lu.se

3.1.3 Biomechanical simulation

The biomechanical simulations (WP6) will be performed in close collaboration with the biological analysis done at the cellular and lower levels (WP4). An iterative coupling approach will be used to simulate the interactions across these two levels of bio-complexity. From this point of view, the biomechanical simulations will be executed on requests according to the input needed by WP4.

When started for a specific patient, the biomechanical simulation will connect to the image database and retrieve patient CT/MRI data and the segmentation of the tissues performed by the image analysis work-package (WP7). Therefore, availability of all the required input data needs to be checked in order to ensure the correct initialization before simulation is started. Based on the input images, a finite element mesh will be automatically generated for the specific patient. The mesh and boundary conditions constitute the input model for the biomechanical simulation and need to be stored for future verification. At each calculation step, mechanical information will be provided to the "cellular" algorithm. This information should be stored as well in order to analyze tumour growth. Figure 1 summarises the data flow between the different work packages.

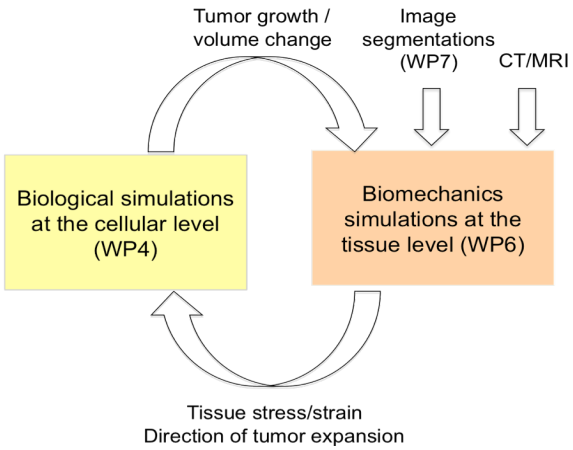


Figure 1: Data flow between the different work packages.

3.1.4 Image processing

This section describes an example scenario for image processing in chronological order. Detailed information can be found in D2.3.

Figure 2

1. A clinical partner (CP1) decides to create a new patient case in the database, because complete patient data is available, e.g. from file. The data includes PET/CT images of the lung for different points in time.
2. CP1 creates a pseudonym Ps for the patient P.
3. CP1 creates a patient entry on ContraCancrum's DICOM server for P
4. CP1 copies the image data for patient P under the pseudonym Ps from the hospitals PACS system to ContraCancrum's DICOM server.
5. CP1 uploads a case description for Ps to the database.
6. CP1 submits the image data for processing.

Figure 3

7. A notification is sent to the responsible image processing partner (IP1). This partner is responsible for registration of multi-modal PET/CT images of the lung.
8. IP1 receives the notification.
9. IP1 downloads the complete patient case of Ps (images and non-image data) to a local database.

10. IP1 uses the tools available at his organisation to register each CT image of Ps to the corresponding PET image.
11. IP1 uploads the resulting registration information to the database.
12. IP1 submits this data for further processing.
13. A notification is sent to the responsible image processing partner (IP2). This partner is responsible for registration of CT images of the same patient at different points in time.
14. IP2 receives the notification.
15. IP2 downloads the complete patient case of Ps (images and non-image data) to a local database. If IP2 works at the same partner as IP1, he may want to use the data which has been downloaded by IP1 already.
16. IP2 uses the tools available at his organisation to register the time series of CT image of Ps.
17. IP2 uploads the resulting registration information to the database.
18. IP2 submits this data for further clinical approval.
19. A notification is sent to the responsible clinical partner (CP2). This partner is responsible for approving the registration results.
20. CP2 receives the notification.
21. CP2 uses a verification and visualisation tool to download the registration information. The tool downloads the original images from the local DICOM server and presents them overlaid to CP2.
22. CP2 checks the registration result and eventually approves it with the help of the verification and visualisation tool.
23. The status of the registration in the database is set to "approved" and a notification is sent to the responsible image processing partner (IP3). This partner is responsible for segmentation of lung tumours in PET/CT images.
24. IP3 receives the notification.
25. IP3 downloads the complete patient case of Ps (images and non-image data) to a local database. If IP3 works at the same partner as IP2, he may want to use the data which has been downloaded by IP1 already.
26. IP3 uses the tools available at his organisation to segment the tumour and the normal tissue. Register for each of Ps' PET/CT images.
27. IP3 uploads the resulting segmentation information to the database.
28. IP3 submits this data for further clinical approval.
29. A notification is sent to the responsible clinical partner (CP3). This partner is responsible for approving the segmentation results.
30. CP3 receives the notification.
31. CP3 uses a verification and visualisation tool to download the registration and segmentation information. The tool downloads the original images from the local DICOM server and presents them overlaid to CP3.
32. CP3 checks the segmentation result and eventually approves it with the help of the verification and visualisation tool.
33. The status of the whole patient case Ps in the database is set to "image processing approved". Now the image processing tasks are completed and and a notification is sent to the responsible modelling partners in WP4 (MP3) and the responsible biomechanical modelling partner in WP6.

The requirements are represented in the following use case and sequence diagrams.

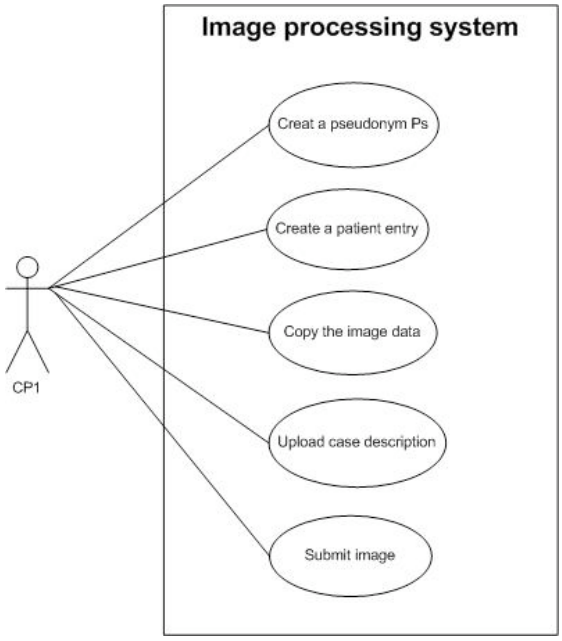


Figure 2 Use case diagram of Clinical Partner 1 (CP1)

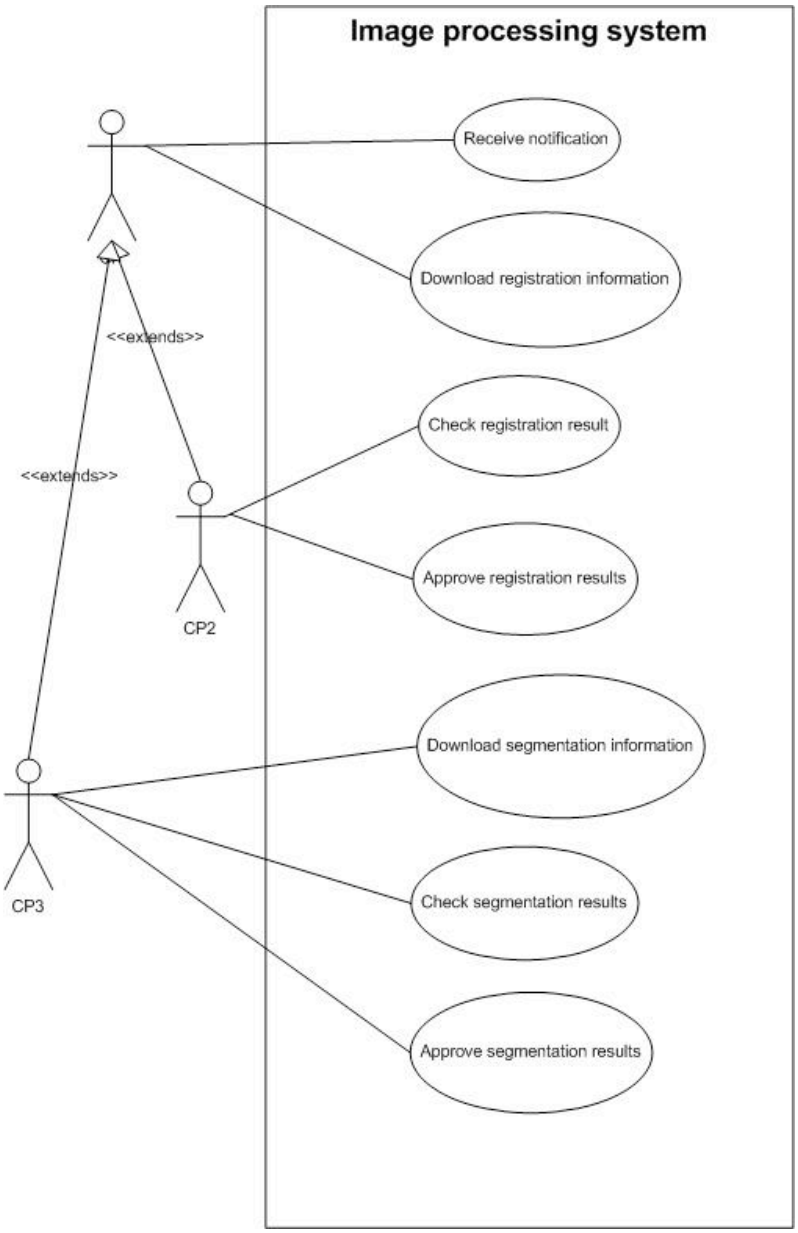


Figure 3 Use case diagram of Clinical Partner 2 & 3 (CP2 & CP3)

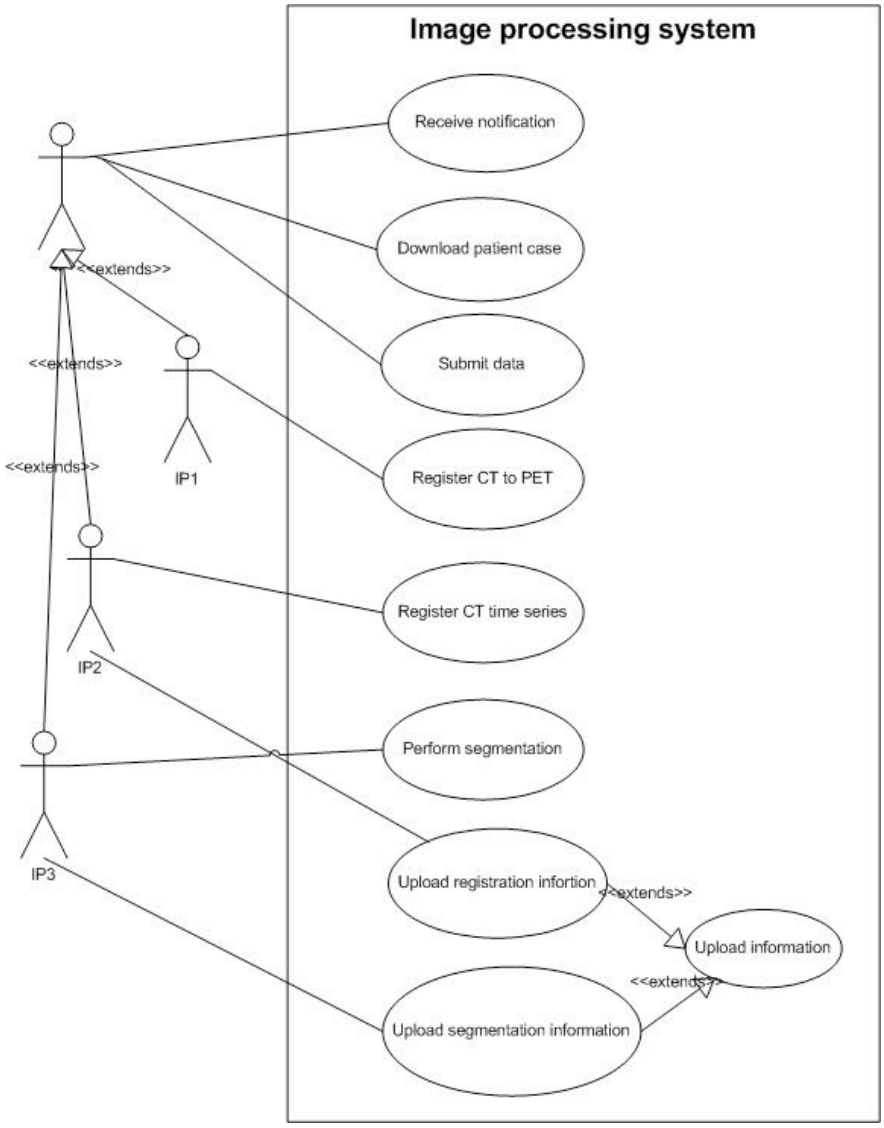


Figure 1 Workflow and use case diagram from the perspective of the image processing partner (IP1)

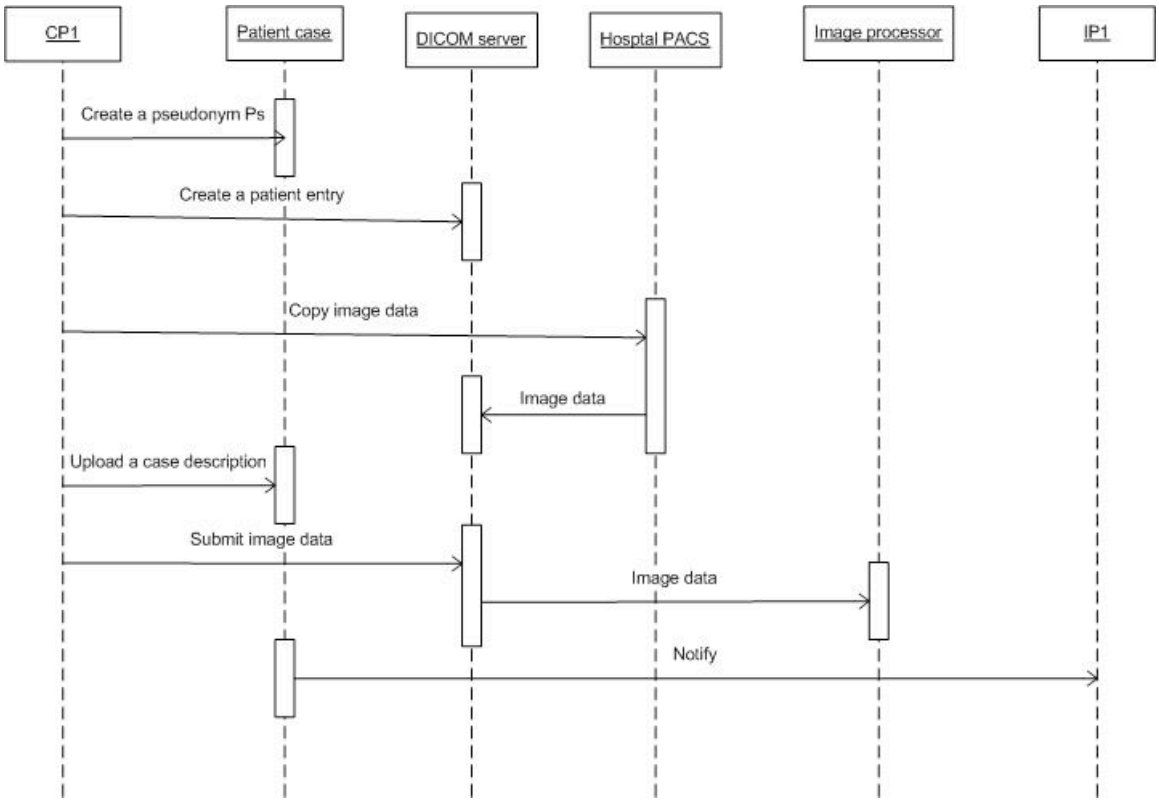


Figure 2 Data flow and task sequence diagram between clinical partner 1 and clinical partner 2

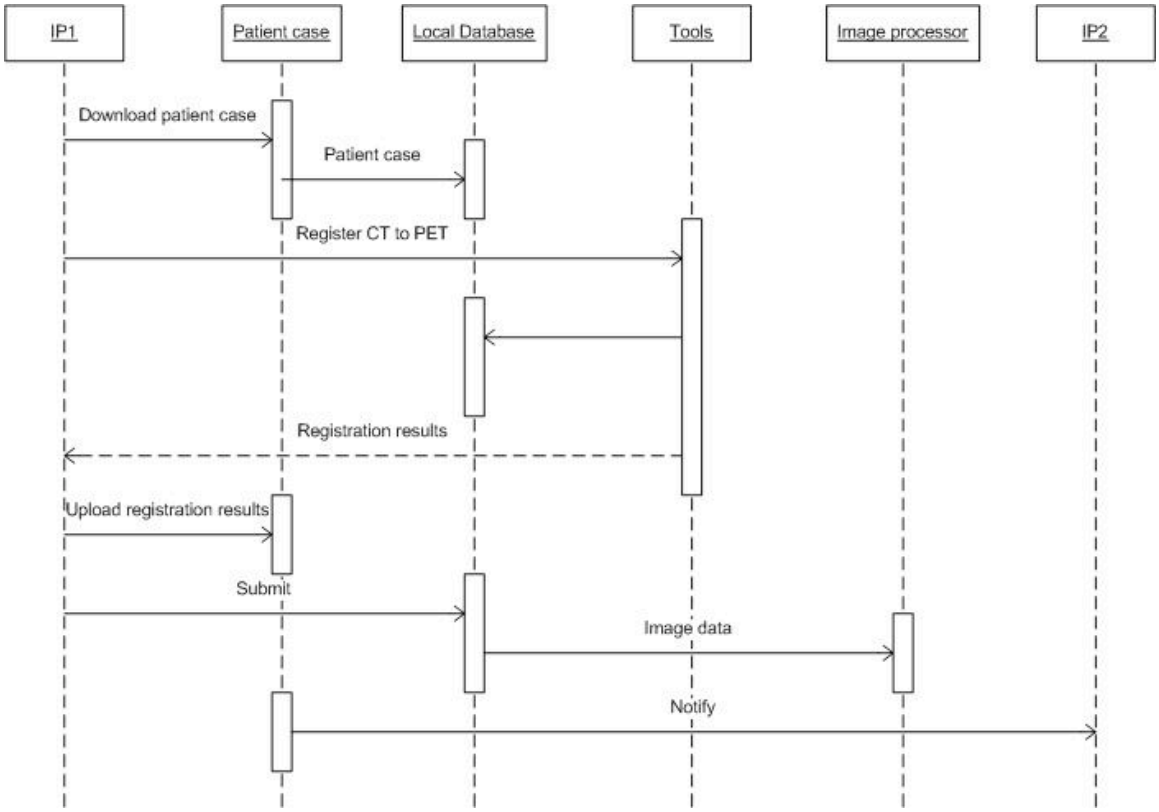
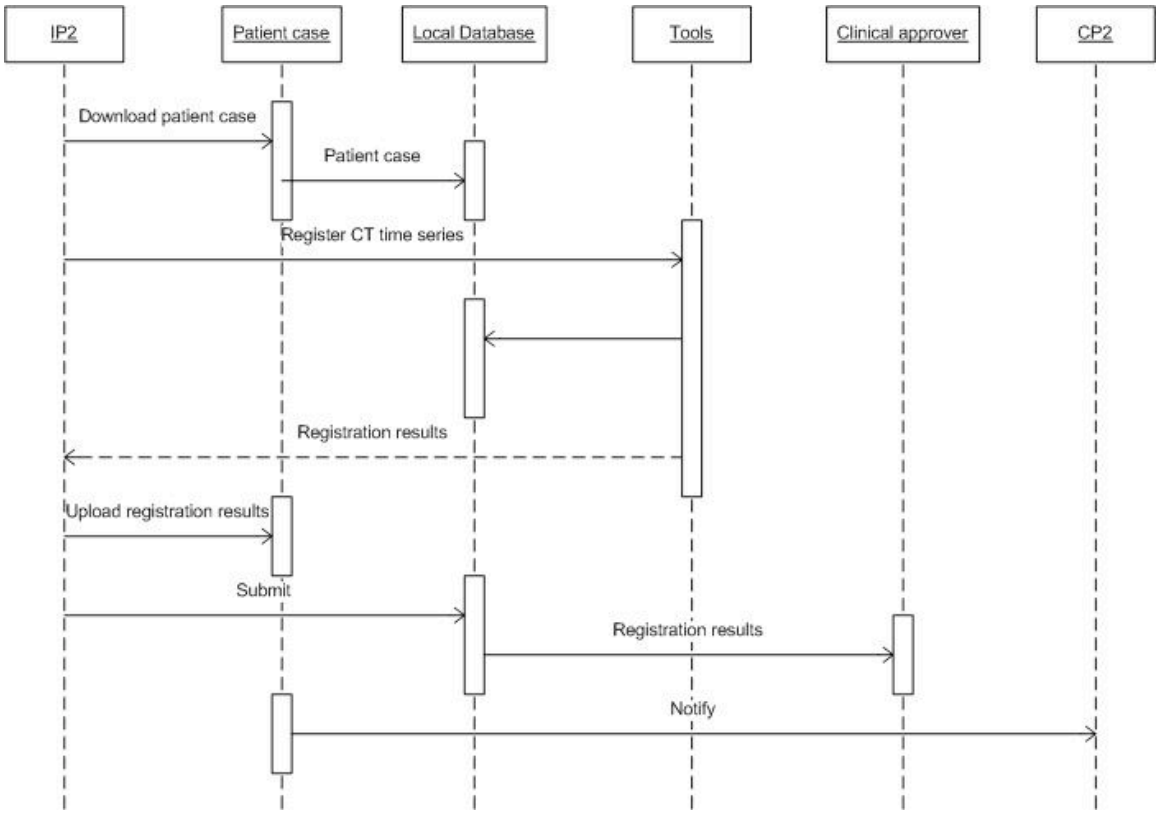


Figure 3 Data flow and task sequence diagram between image processing partner 1 and image processing partner 2



.

Figure 4 Data flow and task sequence diagram between image processing partner 2 and clinical partner 2

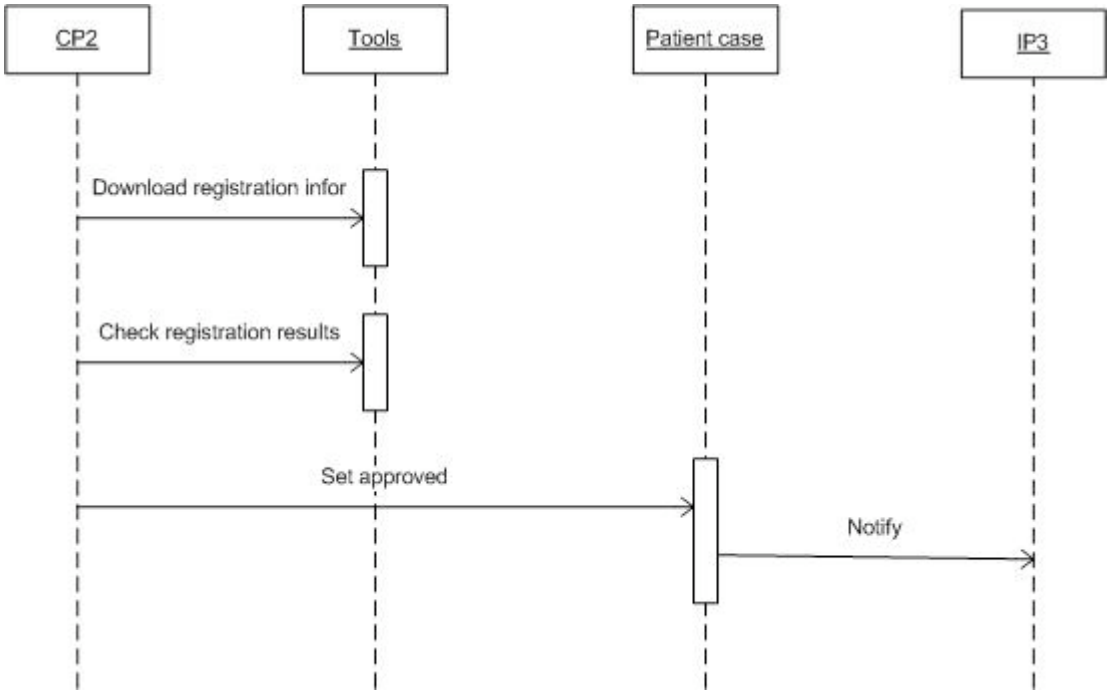


Figure 5 Data flow and task sequence diagram between clinical partner 2 and image processing partner 3

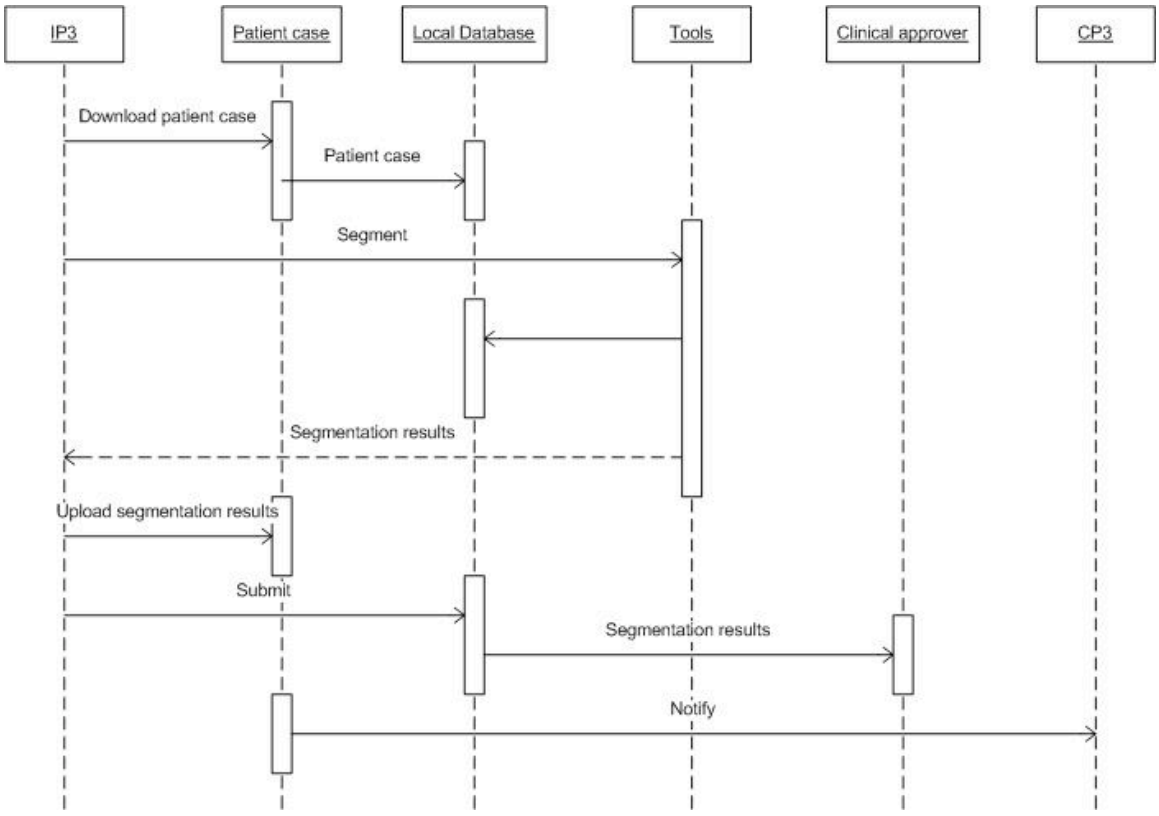


Figure 6 Data flow and task sequence diagram between image processing partner 3 and clinical partner 3

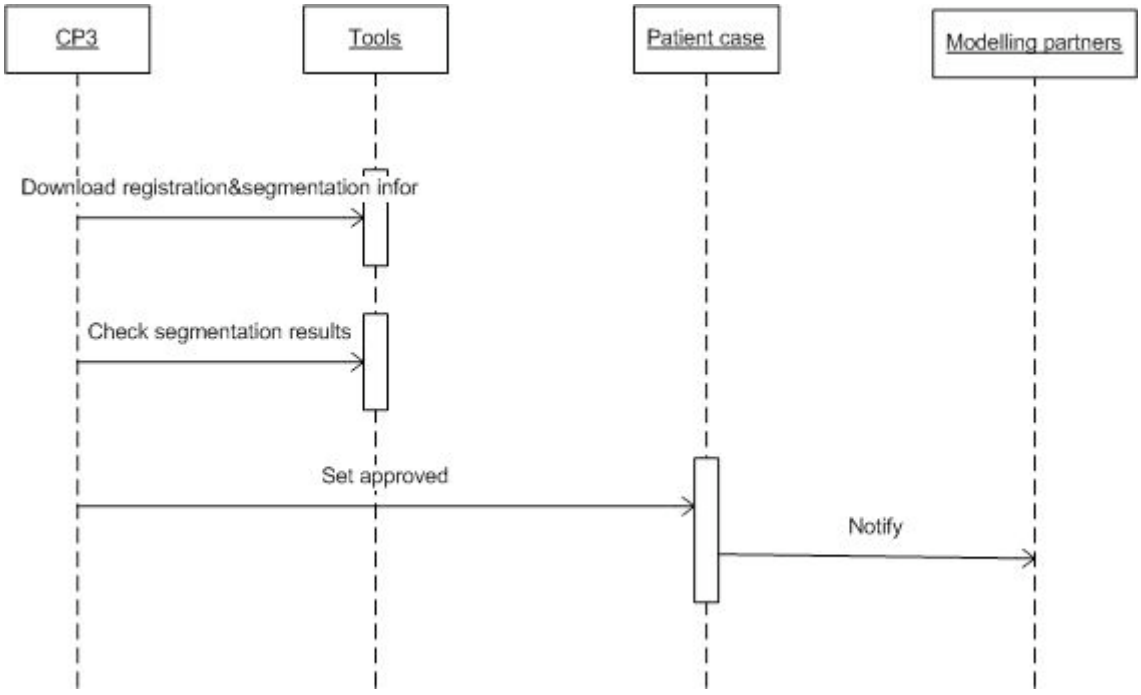


Figure 7 Data flow and task sequence diagram between clincal partner 3 and tumour growth modelling partners

Near-term and long-term proposed solutions

In the near-term automating the process of linking discrete simulations will be done through scripting of AHE commands, as is used in the binding affinity calculations in work package 5, whereby a series of scripts have been developed which automate the process of setting up an MD simulation and creating the necessary input data sets, running a series of chained MD simulations on a powerful grid resource, and then transferring the data back to local machines for post processing.

The longer-term solution will be to use the GSEngine workflow system developed in the FP6 ViroLab (http://www.virolab.org) project, to either construct workflows graphically, or through scripting, and share them between a group of users.

GSEngine[21], developed by the FP6 ViroLab project, consists of two important parts to automatically run and manage ViroLab virtual experiments. The first is an interpreter of the JRuby language in which experiments are written. The second part is the runtime library that provides all the specific functionalities of the virtual laboratory which fall beyond the capabilities of the interpreter's standard library. The runtime library provides a defined set of routines that allow the developer to perform the following actions:

- Instantiate Grid Objects (in general, or ViroLab Gems in the case of tools of ViroLab origin) on a suitable computing element in the ViroLab infrastructure
- Destroy previously created Grid Object instances
- Execute remote operations on those instances with required input parameters and receiving specified output results
- Store any important data outside the experiment execution machine in a safe Laboratory Data Store (not yet supported)
- Access remote data sources in a unified way, especially the ViroLab Data Access system that integrates crucial ViroLab databases
- Interact with experiement executor asking him to provide some input
- Freely use any kind of functionality that the JRuby standard library provides

The library is loaded into the interpreter every time execution of a ViroLab experiment plan is commended. The Runtime is responsible for holding and managing the entire experiment execution context. This may involve taking care of unique execution identification, passing authorization tokens on the user's behalf and maintaining information regarding the state of the experiment run.

Furthermore, the GSEngine interpreter with its runtime library may be referred from within any arbitrary Java application via its API. The API provided is intended to be compiled by several implementations. One can perform script interpretation within the JVM of calling application, while the other is allowed to call dedicated remotely staged service, namely GSEngineServer.

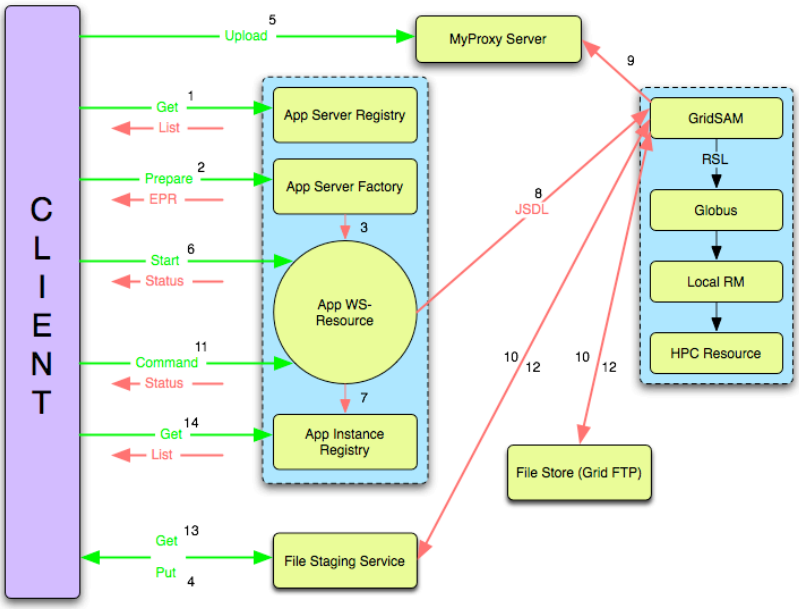
[21] GSEngine - http://virolab.cyfronet.pl/trac/vlruntime

4 Web services and solutions

Application Hosting Environment

The Application Hosting Environment (AHE[22]) developed at University College London, provides simple desktop and command line interfaces, to run applications on resources provided by national and international grids, in addition to local departmental and institutional clusters, while hiding from the user the details of the underlying middleware in use by the grid. In addition, a mobile interface for Windows Mobile based PDAs is available, and an iPhone interface is in development. The AHE is able to run applications on both UNICORE and Globus grids, meaning that a user can use a single AHE installation to access resources from the UK NGS and DEISA for example. Development of an EGEE connector for AHE is currently underway. The recent version 2.0 release[23] of the AHE also includes support for HARC (the Highly-Available Resource Co-allocator) advance reservations[24]. The version 3.0 release currently under development will also include SPRUCE[25] (Special PRiority and Urgent Computing Environment) functionality for urgent computing scenarios.

The AHE is designed to allow scientists to quickly and easily run unmodified, legacy applications on grid resources, manage the transfer of files to and from the grid resource and monitor the status of the application. The philosophy of the AHE is based on the fact that very often a group of researchers will all want to access the same application, but not all of them will possess the skill or inclination to install the application on remote grid resources. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application. This community model draws a parallel with the modus operandi of numerous scientific application communities such as ContraCancrum A diagram of the AHE's architecture is given in the figure below.



[22] Application Hosting Environment - http://www.realitygrid.org/AHE

[23] S. J. Zasada and P. V. Coveney, *Virtualizing Access to Scientific Applications with the Application Hosting Environment*, Computer Physics Communications 2009, DOI: 10.1016/j.cpc.2009.06.008

[24] HARC - http://www.cct.lsu.edu/site54.php

[25] SPRUCE - http://spruce.teragrid.org

The AHE client is easily installed on an end user's machine, requiring only that they have a Java installation and an X.509 certificate for the grid, which they want to access. The client package contains both GUI and command line clients which interoperate, allowing jobs launched with the GUI client to be manipulated with the command line tools and vice versa, and also application workflows to be easily constructed.

A current EPSRC funded project in the UK, UF-Security[26], is developing the AHE further to remove the need for users to obtain individual grid certificates in order to use resources. Instead an audited credential delegation (ACD) is stored in a central repository that users can access through local username/password authorisation. This ACD can then be used to access grid resources. The technology is built in to the AHE client portal interface, meaning that access to their certificate is seamless to the user. It is expected that an initial implementation will be available in November 2009.

Web services

Web services provide a standard means of interoperating between different software applications that may be running on a variety of platforms and/or frameworks. Many applications nowadays are on the Internet, and the users are using different systems, platforms and software languages. Web services are a solution that allows them to interact.

In the ContraCancrum project, the following proposed functions will be implemented as Web Services:

- A data format transforming service, for instance a text-to-XML translator that allows biomechanical models to be transformed between text format and XML format;
- A notification service that provides process notification during the image processing flows;
- Any applications that needed to be implemented as Web Services. Most of the applications are in theory can be deployed as Web Service, however, the feasibility in terms of, such as needs and performance, may result in different decisions.

Web Services will bring at least the following benefits to the project:

- Providing a service layer that bridges the underlying Grid facilities and the front-end portal;
- Serving as an inter-media to allow users to access certain functionalities via the Internet;
- Allowing standard functionalities to be implemented in open-standard, which thus can be invoked by different applications;
- Enabling better co-operativity when using the functions in or with other projects.

Near-term and long-term proposed solutions

In the near term the AHE, which is deployed at UCL, will allow job submission of individual codes such as NAMD on a variety of UK, EU and US resources, and made available to ContraCancrum users project wide. Other codes as required by project members can be hosted from within AHE. Resources are currently available through a VPH NoE DEISA allocation, supported by the DEISA VPH Virtual Community. In the longer term, efforts involving the interoperability of different grids using SAGA[27] (a Simple API for Grid Applications) will be implemented and launched from within the AHE platform.

[26] UF-Security project - http://www.realitygrid.org/uf-security

[27] SAGA - A Simple API for Grid Applications - http://saga.cct.lsu.edu

5 Overall architecture

WP3 focuses on developing a software environment to support clinicians and researchers within the ContraCancrum project. WP3 will develop these tools through several subtasks, which cover software environment design, remote access to data, service oriented environment development, development of workflow engine, system integration. We will do this in close collaboration with Work Packages 4-10, and tailor the software developed to the user needs identified in WP2 - user requirements. Where appropriate, we shall work with other VPH projects such as the Network of Excellence - in particular the VPH ToolKit (WP3) to ensure that software developed will interoperate with other projects, and integrate with the wider multi-scale VPH effort.

Researchers and clinicians need to be able seamlessly to access suitably pseudoanonymized clinical data, and use it to run both simulation and informatics' applications, and then visualize, or otherwise analyze, the results. WP3 will provide software tools to address data storage, access and management.

Access methods

Access to DICOM data will be conducted in a number of ways. For clinicians and other users where their PACS workstations have static IP addresses direct DICOM connectivity will be used.

The online data repository will be initially hosted as a Drupal[28] content management system (CMS), allowing rapid testing of PHP code for database access. Drupal also has built-in user authentication and authorisation facilities.

As the integrated environment develops further, particularly as web services and other computational services can be access through the web portal, more sophisticated authorisation will be used. We plan to use Shibboleth[29] that supports X.509 grid certificates in conjunction with audited credential delegation (ACD) to support shared certificates and online access to HPC grid resources.

Architecture design

5.1.1 Requirements and functionalities summary of WP6

Requirements of Files & Data

- WP6_FD_1: This file includes the finite element mesh, mechanical properties and boundary conditions.
- WP6_FD_2: Data will be stored in a text format.
- WP6_FD_3: The amount of data could reach up to 80 Gb (binary files) for one patient and one set of parameters.
- WP6_FD_4: It is possible to store result information every n increments, and/or to store the mechanical information in and around the tumor much more frequently than for distant tissues.

Functionalities:

From the technological perspective, WP3 will provide means for WP6 to download necessary data from the database for the simulation, and upload the simulation results to the server. The portal services to achieve will be described later; below are listed the main functionalities that WP3 needs to provide for WP6.

[28] Drupal CMS - http://drupal.org

[29] Shibboleth - http://shibboleth.internet2.edu

- WP6_FN_1: In case the number of cellular simulations grows to very large numbers, simplifying could be used to decrease this amount of data.
- WP6_FN_2: Data can be translated to an XML format if required.
- WP6_FN_3: Availability of all the required input data needs to be checked in order to ensure the correct initialization before simulation is started.
- WP6_FN_4: Based on the input images, a finite element mesh will be automatically generated for the specific patient. The mesh and boundary conditions constitute the input model for the biomechanical simulation and need to be stored for future verification.
- WP6_FN_5: At each calculation step, mechanical information will be provided to the "cellular" algorithm. This information should be stored as well in order to analyze tumor extension.

5.1.2 Requirements and functionalities summary of WP7

Requirements of Files & Data

- WP7_FD_1: This database must be able to store any format (binary or text).
- WP7_FD_2: Registration results can be stored either as transformation matrices (rigid registration) or as point sets (non-rigid registration).
- WP7_FD_3: Segmentation results can be stored either as surface meshes or as label images. Label images typically have a bit depth smaller than the original images.
- WP7_FD_4: The non-image data storage for image processing results should be allowed for maximally the size of the original clinical images.
- WP7_FD_5: Non-DICOM data must be stored in the database for non-image data.
- WP7_FD_6: Information on the processing steps must be stored in the database. These information includes original data has been used to create the result, the tools used to create the result and which person has done it.
- WP7_FD_7: Image processing results must be stored together with the pseudonym of the patient either on the DICOM server or in the non-image database.

Functionalities:

Based on the above descriptions, the system required by WP7 can be divided into three components: user interface, database and functional module. The user interface will be a web-based portal which is used for user instruction and viewing results; the database is data storage, which covers XML data, non-image data, DICOM data, segmented data, etc; function module are system functions, like image processing, reality and virtuality comparison, etc.

- WP7_FN_1: The DICOM server for clinical images and processed images should use the DICOM protocol for data transfer.
- WP7_FN_2: The users of the database are responsible for creating the necessary interfaces to read and process the data. VTK is the preferred file format for surface meshes, rigid and non-rigid transformations.
- WP7_FN_3: All clinical image data should be pseudonymised by the clinicians before being stored on the DICOM server.
- WP7_FN_4: The pseudonymisation should be reversible by the clinical partner in order to be able to store new data for the same patient in the database.
- WP7_FN_5: A notification mechanism should be required. This mechanism should inform responsible groups/users on new data are available, uploading processing results and changing the workflow.
- WP7_FN_6: A mechanism to change the workflow.
- Downloading & Uploading data

- WP7_DU_1: It should be possible to download all data for a certain patient (identified by a pseudonym) onto a local disk for further local processing. This includes original clinical images as well as processed images and non-image data.
- WP7_DU_2: It should be possible to select certain data or to download everything.
- WP7_DU_3: Original clinical image should be accessible via the DICOM protocol to be copied to a local DICOM server. Furthermore, it should be possible to download them to a local directory.
- WP7_DU_4: Uploading complete patient data at the same time is not necessary. Only single results will be uploaded.
- WP7_DU_5: Uploading and downloading data should be done via the web browser.

5.1.3 Requirements and functionalities summary of WP4

- WP4_FD_1: Each geometrical cell of the discretization should contain a single biological cell. This implies that for a large tumour and adjacent anatomical region of interest of dimensions 100 mm x 100 mm x 100 mm = 10^6 mm^3
- WP4_FD_2: we would need: 10^6 biological cells included in each mm^3 x 10^6 mm^3 = 10^{12} biological cells = 10^{12} geometrical cells.
- WP4_FD_3: Each biological cell (corresponding to a single geometrical cell) would be have parameter: Metabolic layer (char), Cell category (char), Phase (char), Treatment 1 hit or non hit cell (boolean) Treatment 2 hit or not hit cell (boolean), time_spend (integer).

5.1.4 Architecture Design

From the requirements of WP4, WP6 & WP7, the system can be divided the system into 4 components: web portal, database, functional module and simulation modules.

The **database component** covers five types of data: modelling result, DICOM image, XML Meta data, segmented images, as well as text and binary data. We are currently in discussions with the EU FP6 ACGT project (http://www.eu-acgt.org/) and seeking to use their master intology in ContraCancrum.

At this stage, the **functional components** required by WP4, WP6 and WP7 are: DecreaseAmountofData, Text2XML, IfImageAvailable4Simu, ImagedBaseMesh, DataTransferWithDicom, VtkBasedTransformations, ImagePseudonumised, ImageRePseudounmised, NotificationOnUpdate, ChangeWorkFlow.

Front-end design

The front-end is based on a web portal. According to JSR168, a portal is a web-based application that provides personalization, single sign on, content aggregation from different sources and hosts the presentation layer of Information Systems. Aggregation is the action of integrating content from different sources within a web page. A portal may have sophisticated personalization features to provide customized content to users. Portal pages may have different set of portlets creating content for different users. Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems. Based on the user requirement, user would need to see a group of result. With a portal/portlet-based interface, user can customize their interface.

Figure 8 summaries the requirements from WP4, WP6, and WP7 for the portal services. It was drafted at the Hamburg workshop.

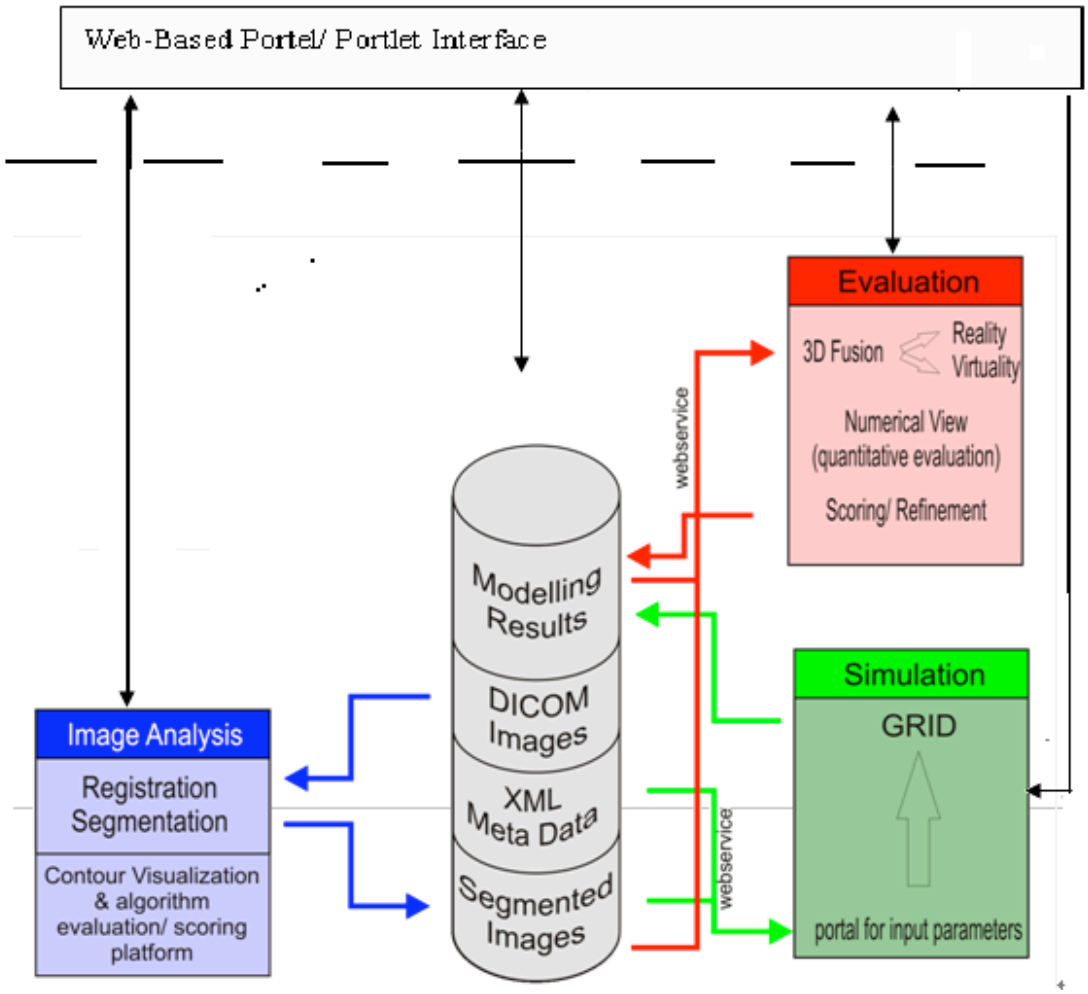


Figure 8 Portal

Workflow information related to WP5 (genetic, protein and microarray data is not included in this diagram as it pertains to WP4, 6 and 7 only.

The simulation results and imaging processing results will be stored in a central database. Appropriate email notification services will be developed to notify users of simulation result completion.

To ensure that the simulation can be launched when needed, advance reservation facilities through HARC will be used via the AHE, as well as SPRUCE for urgent computing requirements.

The requirements on the ContraCancrum portal from WP4, WP6 and WP7 include:

a. Welcome GUI

- Login & Logout
- Different content for user with different right.

b. GUI with Database

- Database view and edit.
- Data downloading and uploading.

c. GUI for Operations

- Possible operation for authorised user.
- Image processing information.

d. GUI for Simulation

- Allow input parameters for simulation.
- Result visualization.
- Result comparison.

1. Simulation flow

a. Receives input parameters from the user (researchers)
b. Launch the simulation, and simulation will be run using Grid resources
c. Save the simulation result to the database
d. Check the simulation result
e. Change parameters
f. Launch another simulation
g. Repeat c-f to get a satisfactory result

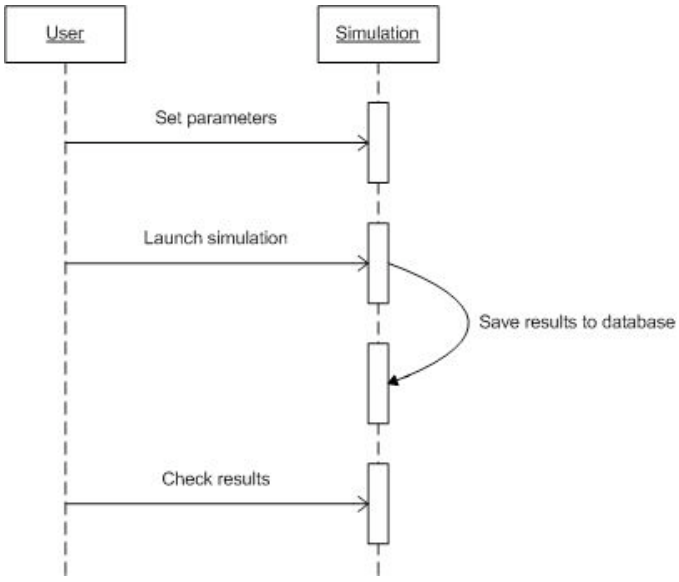


Figure 12. Diagram illustrating data flow during a user-instantiated simulation

2. Evaluation flow

a. Receives input data in order for users (clinician or researcher) to search for the database
b. Access database to retrieve simulation result and imaging processing result
c. Use visualisation tool to post process simulation result and imaging processing result for evaluation, 1D (chat), 2D, 3D visualisation should be provided
d. At this stage, user may decide to run the simulation again, refer to 1 for detail steps
e. Provide scores for simulation prediction and segmentation, save the scores in the central database

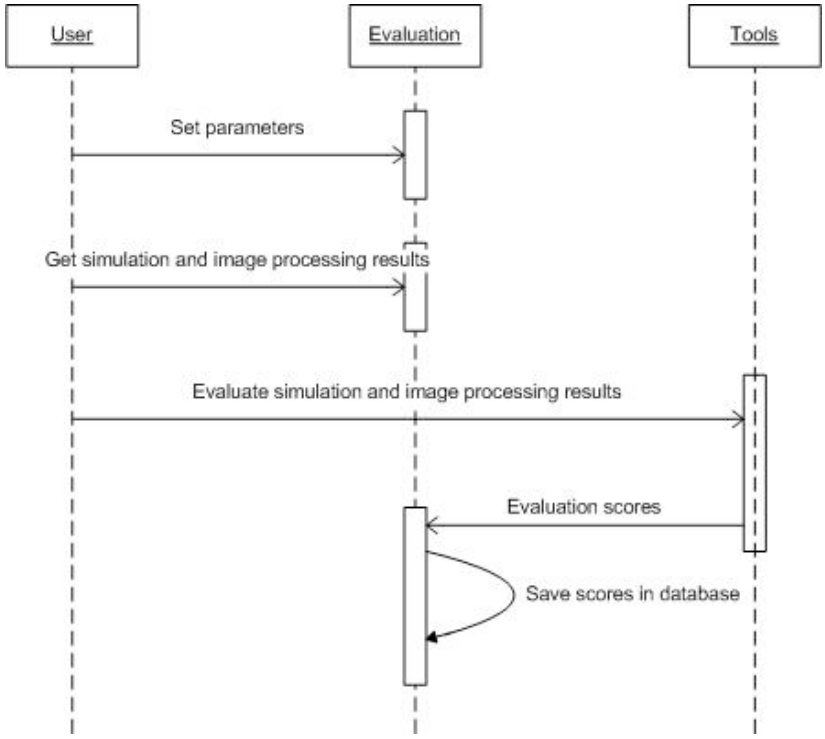


Figure 13. Diagram illustrating data flow and user interactivity for clinician scoring of image processing results

6 Conclusion

In this report we have provided a high level overview of the ContraCancrum technical environment. In developing this environment, two main themes lie at the forefront. Firstly, wherever possible we intend to re-use software solutions developed by other researchers in the VPH research sphere. Secondly, we will use standard protocols and formats, as well as standardised data structures (ontologies) to help ensure the reuse of tools developed within this project. The sharing of technologies and techniques with other VPH researchers via the VPH NoE is considered paramount to the wider success of this project. In ensuring access to the necessary computational resources, the online data environment will be integrated with work-flow tools and web services to give clinicians and researches access to high-performance computing resources distributed throughout Europe, as well as access to local resources. Finally, in evaluating the technical environment for clinical use, at the end of the project it will be validated as a multi-level clinical decision making tool and a research tool by oncologists and scientists.

7 Appendix

Detailed scenario for the cellular and higher levels discrete simulator

Each (cubic) geometrical cell of volume 1 mm^3 would contain 10^6 biological cells, and a large tumour and adjacent anatomical region of size 100mm x 100mm x 100mm requires 10^6 geometrical cells. Each geometrical cell needs at least the following registers:

1. 1 character register to denote the metabolic layer within which the geometrical cell is lying (well vascularized tumour region, poorly vascularized tumour region, edema, cyst, adjacent normal tissue of type "1", of type "2" etc.)
2. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the stem cells. The content of each register would be the number of biological cells residing in the given hour of the G0 phase. The contents of the rest of the registers below would have analogous meanings.
3. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the stem cells
4. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 1 hit stem cells
5. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 1 hit stem cells
6. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 2 hit stem cells
7. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 2 hit stem cells
8. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the progenitor cells having undergone 1 mitosis
9. 100 integer registers, each one each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the progenitor cells having undergone 1 mitosis
10. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 1 hit progenitor cells having undergone 1 mitosis
11. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 1 hit progenitor cells having undergone 1 mitosis
12. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 2 hit progenitor cells having undergone 1 mitosis
13. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 2 hit progenitor cells having undergone 1 mitosis
14. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the progenitor cells having undergone 2 mitoses
15. 100 integer registers, each one each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the progenitor cells having undergone 2 mitoses
16. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 1 hit progenitor cells having undergone 2 mitoses
17. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 1 hit progenitor cells having undergone 2 mitoses
18. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 2 hit progenitor cells having undergone 2 mitoses
19. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 2 hit progenitor cells having undergone 2 mitoses
20. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the progenitor cells having undergone 3 mitoses
21. 100 integer registers, each one each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the progenitor cells having undergone 3 mitoses

22. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 1 hit progenitor cells having undergone 3 mitoses
23. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 1 hit progenitor cells having undergone 3 mitoses
24. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 2 hit progenitor cells having undergone 3 mitoses
25. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 2 hit progenitor cells having undergone 3 mitoses
26. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the progenitor cells having undergone 4 mitoses
27. 100 integer registers, each one each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the progenitor cells having undergone 4 mitoses
28. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 1 hit progenitor cells having undergone 4 mitoses
29. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 1 hit progenitor cells having undergone 4 mitoses
30. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 2 hit progenitor cells having undergone 4 mitoses
31. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 2 hit progenitor cells having undergone 4 mitoses
32. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the progenitor cells having undergone 5 mitoses
33. 100 integer registers, each one each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the progenitor cells having undergone 5 mitoses
34. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 1 hit progenitor cells having undergone 5 mitoses
35. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 1 hit progenitor cells having undergone 4 mitoses
36. 100 integer registers, each one corresponding to each hour of the duration of the G0 phase of the treatment 2 hit progenitor cells having undergone 5 mitoses
37. 100 integer registers, each one corresponding to each hour of the total duration of the G1, S, G2, M phases of the treatment 2 hit progenitor cells having undergone 5 mitoses
38. 1 real register for the mean time of the differentiated cells
39. 1 real register for the mean time of the apoptotic cells
40. 1 real register for the mean time of the necrotic cells