Transatlantic TUmour MOdel Repositories

D4.1.2
Finalized version of web services for TUMOR

Project Number:	FP7--IST-247754
Deliverable id:	D4.1.2
Deliverable name:	Finalized version of web services for TUMOR
Submission Date:	13/9/2012

Information Society Technologies

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	TUMOR
Project Full Name:	Transatlantic TUmour MOdel Repositories
Document id:	D4.1.2
Document name:	Finalized version of web services for TUMOR
Document type (PU, INT, RE)	PU
Version:	1.0
Submission date:	
Editor: Organisation: Email:	Thomas S. Deisboeck Massachusetts General Hospital deisboec@helix.mgh.harvard.edu

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

ABSTRACT: This deliverable documents the US (CViT) and EU web services to support cancer model interoperability and TUMOR clinical scenarios for integrating EU-US cancer models. The web services provide a standard communication mechanism through which EU and US model repositories can interoperate. Descriptions of the EU and US repositories and their respective web service APIs are provided.

KEYWORD LIST: cancer modelling, models, web service

MODIFICATION CONTROL			
Version	Date	Status	Author
0.1	14 July 2012	Initial draft	Thomas Taylor
0.2	15 August 2012	Draft	Thomas Deisboeck
0.3	13 September 2012	Final	Thomas Deisboeck

List of Contributors

- Thomas Taylor (INFOTECH Soft)
- Thomas Deisboeck (MGH)
- Fay Misichroni (ICCS)
- Vangelis Sakkalis (FORTH)
- Stelios Sfakianakis (FORTH)

Contents

1 EXECUTIVE SUMMARY 5
2 INTRODUCTION 6
3 EU REPOSITORY 7
TUMOR EU-REPOSITORY GUI 7
TUMOR EU-REPOSITORY WEB SERVICE 8
4 US (MGH-CVIT) REPOSITORY 15
MGH CENTER FOR THE DEVELOPMENT OF A VIRTUAL TUMOR (CVIT) REPOSITORY 15
MGH-CVIT DMR DOMAIN MODEL 16
MGH-CVIT DMR WEBSITE 27
MGH-CVIT DMR TUMOR WEB SERVICE 28
5 APPENDIX I - ABBREVIATIONS AND ACRONYMS 34

1 Executive Summary

The TUMOR project aims at developing a European *clinically oriented* semantic-layered cancer digital model repository from existing EC projects that will be interoperable with the US grid-enabled semantic-layered digital model repository platform at CViT.org (Center for the Development of a Virtual Tumor, Massachusetts General Hospital (MGH), Boston, USA; currently IT-operated by InfoTech Soft Inc.) which is NIH/NCI-caBIG compatible. This interoperable, CViT interfaced, environment will offer a range of services to international cancer modellers, bio-researchers and eventually clinicians aimed at supporting both basic cancer quantitative research and individualized optimization of cancer treatment. This 'Transatlantic' project will therefore be the starting point for an international validation environment that will support joint applications, verification and validation of the clinical relevance of cancer models.

This deliverable documents the US (CViT) and EU web services to support cancer model interoperability and TUMOR clinical scenarios for integrating EU-US cancer models. The web services provide a standard communication mechanism through which EU and US model repositories can interoperate. Descriptions of the EU and US repositories and their respective web service APIs are provided.

In deliverable D4.1.1, a review of the state-of-the-art web service technologies and analysis of interoperability design directions tailored to the TUMOR project was completed. The work described in D4.1.2 applies the technologies described in D4.1.1 to implement the TUMOR web services.

2 Introduction

The main aim of the TUMOR project is to create a European-based digital repository for clinically oriented cancer models. The repository will store models provided by other EC projects such as the Advancing Clinico Genomic Trials on Cancer (ACGT) and the Clinically Oriented Translational Cancer Multilevel Modeling (ContraCancrum) projects. Biological model repositories are not novel, as demonstrated by existing services provided by the CellML repository, E-Cell, and biomodels.net to name but a few. However, one of the key aims of the TUMOR project is to enable the European cancer model repository to seamlessly interoperate with its US equivalent service that was developed by the Center for the Development of a Virtual Tumor (CViT) project led by the Massachusetts General Hospital (MGH) in Boston, USA (CViT.org and CViT DMR are currently IT-operated by InfoTech Soft Inc.). This 'transatlantic' link may ultimately allow the US and EU cancer research communities to pool their resources through effective model sharing, and act as a bridge between the two communities to foster further research advancements in cancer.

WP4 of the TUMOR project focuses on developing interoperable interfaces between the two repositories. This will be achieved by the development of a set of Web services to allow the two repositories to communicate with each other, a task led by MGH/Infotech. Secondly, and the focus of this deliverable, the TUMOR project is to develop a simulation markup language specifically targeted at the cancer modelling domain that will act as the standard communication format between elements of the TUMOR infrastructure and eventually for exporting models to external services.

To address the specific domain of cancer modelling, we have developed a markup language, *TumorML*, to describe computational cancer models within TUMOR. The specification of TumorML is fully described in deliverable D4.2.2.

This deliverable documents the US (CViT) and EU web services to support cancer model interoperability and TUMOR clinical scenarios for integrating EU-US cancer models. The web services provide a standard communication mechanism through which EU and US model repositories can interoperate. Descriptions of the EU and US repositories and their respective web service APIs are provided.

3 EU Repository

The contents of the TUMOR EU-Repository can be accessed through two different interfaces:

- A Graphical User Interface (GUI), using a Model-View-Controller design pattern and the REST architectural style, designed to be used by humans,
- Two REST Web services, designed for machine-to-machine communication.

The aforementioned interfaces use different authentication and authorization mechanisms. However, these mechanisms use the same authentication and authorization resources. More information concerning those mechanisms will be provided in the following sections.

TUMOR EU-Repository GUI

The GUI of TUMOR EU-Repository can be used to interact with the database.

Access to TUMOR EU-Repository is limited to authenticated users. Each user must provide his username and password in order to login to the TUMOR EU-Repository GUI. Access to resources and application of actions are limited to the users that have the appropriate rights.

Users can easily manipulate the resources stored in the database, by accessing specific URLs. Some of the actions that a user can perform on a resource, according to the resource and his rights, are:

- view a list of entries,
- view the details of an entry,
- add a new entry,
- delete an existing entry,
- edit an existing entry,
- download an entry, if the resource represent the content of a file,
- upload an entry, if the resource represent the content of a file,
- login,
- logout, and
- search.

In the implementation of TUMOR EU-Repository Front End we have used the CakePHP application framework[1]. The aforemented actions are either already available through the framework or are extentions to it. Further description of these actions is out of the scope of the present document.

[1] http://cakephp.org

TUMOR EU-Repository Web Service

The TUMOR EU-Repository Web Service API exposes part of the content of the TUMOR EU-repository in TumorML form as described in *"D4.2.2: The TUMOR Markup Language (TumorML) Version 1.0 Specification"*. The TUMOR EU-Repository Web Service API is initially intended to be used by the workflow environment. The implemented web service methods can be divided in two categories:

- The ones that are used to access the content of the TUMOR-EU Repository (listentries, gettool, userinfo, listpatients, download) and
- The ones that are used in the machine-to-machine authentication mechanism (OAuth2).

The API for the EU repository is implemented following the principles of REST architectural style, as described in Deliverable 4.1.1. In particular, the API uses the HTTP application level protocol to implement, in a request – response message exchange pattern, the repository's functionality.

On the security front, there is the need for authenticating the users with the minimal possible distraction (Single Sign On) and also supporting authorization and access control. To address both of these concerns, the EU repository uses the OAuth 2.0 (Open Authorization, version 2.0 - http://oauth.net/2/) protocol, a well-known specification supported by major web companies such as Google, Microsoft, and Facebook. Using OAuth the client application that require access to the EU repository can contact it on the users' behalf without prior knowledge of users' passwords or other authenticating information. The OAuth communication needs to take place before the "normal" access to the repository's contents. After this communication happens the clients are provided with an "access token" that can subsequently use in the EU repository API.

In the following tables we describe the REST based API of the EU repository. According to the HTTP based realization of the REST architectural style we present the the HTTP(S) URL endpoint, the HTTP method used (normally either GET or POST), the parameters passed, and the response status and reply message format. All of the domain specific API requests demand the submission of the OAuth access token so that the EU repository can validate the request and make proper authentication and authorization checks.

OAuth protocol message exchanges

According to the OAuth terminology[2] the EU Repository has the role of the "resource owner/server" and the "authorization server".

[2] The OAuth 2.0 Authorization Framework, draft-ietf-oauth-v2-31, http://tools.ietf.org/html/draft-ietf-oauth-v2-31

authorize		
Description	The authorization request according to OAuth 2.0 Authorization Framework (http://tools.ietf.org/html/draft-ietf-oauth-v2-31#section-4.1.1)	
URL	…/oauth/authorize	
Method	GET	
Querystring	client_id=	REQUIRED. The client identifier
	response_type=	REQUIRED. Value MUST be set to "code".
	redirect_uri=	OPTIONAL. The redirection endpoint of the client.
Returns	200 OK 400 Bad Request	

token		
Description	The Access Token Request according to OAuth 2.0 Authorization Framework (http://tools.ietf.org/html/draft-ietf-oauth-v2-31#section-4.1.3)	
URL	…/oauth/token	
Method	POST	
Querystring	client_id=	REQUIRED. The client identifier
	client_secret=	REQUIRED if the client identifier has a matching secret.
	grant_type=	REQUIRED. Value MUST be set to "authorization_code".
	code=	REQUIRED. The authorization code received from the authorization server
Returns	200 OK & JSON 400 Bad Request 401 Authorization Required	

EU Repository API

listentries		
Description	Enumerates the set of Tools/Models in the TUMOR EU-Repository that the user is authorized to read.	
URL	…/services/listenties	
Method	GET	
Querystring	access_token=	REQUIRED. The access token used for authorization.
Returns	200 OK & XML 401 Authorization Required	

gettool		
Description	Returns the Tool/Model details specified by the provided Tool/Model UID.	
URL	…/services/gettool/[UID]	
Method	GET	
Querystring	access_token=	REQUIRED. The access token used for authorization.
Returns	200 OK & XML 401 Authorization Required	

download (tool/model file)		
Description	Returns the file (associated with a Tool/Model) specified by the provided file UID.	
URL	…/tfiledata/download/[UID]	
Method	GET	
Querystring	access_token=	REQUIRED. The access token used for authorization.
Returns	200 OK & XML 401 Authorization Required	

listpatients		
Description	Enumerates the set of Patients in the TUMOR EU-Repository that the user has authorization to read.	
URL	…/services/listpatients	
Method	GET	
Querystring	access_token=	REQUIRED. The access token used for authorization.
Returns	200 OK & XML 401 Authorization Required	

download (patient data)		
Description	Returns the file (associated with a Patient) specified by the provided file UID.	
URL	…/dfiledata/download/[UID]	
Method	GET	
Querystring	access_token=	REQUIRED. The access token used for authorization.
Returns	200 OK & XML 401 Authorization Required	

userinfo		
Description	Returns information concerning the authenticated user.	
URL	…/services/userinfo	
Method	GET	
Querystring	access_token=	REQUIRED. The access token used for authorization.
Returns	200 OK & XML 401 Authorization Required	

4 US (MGH-CViT) Repository

Here we provide a description of the US (MGH-CViT) cancer model repository and the US web service.

MGH Center for the Development of a Virtual Tumor (CViT) Repository

The MGH-CViT DMR uses Resource Description Framework (RDF) as the foundational means to model and link entities corresponding to real-world concepts, for the DMR, it uses RDF to represent computational models of cancer and links these models to publications, experiments, and data built into a provenance structure. RDF is a standard model for data interchange on the Web. RDF facilitates data merging even when the underlying schemas differ and supports the evolution of schemas over time without requiring changes to the persistent storage and APIs accessing the data. In general, RDF is more robust against schema evolution than relational and XML approaches In addition, RDF allows structured and semi-structured data to be mixed, exposed, and shared across different applications and linked with external websites. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (referred to as a "triple"). MGH-CViT provides the following services.

Digital Model Repository (DMR)

- *RDF-based Semantic database to store models and model metadata*
- *RDF links to other web-based resources produces a semantic graph*

CViT.org Web Application

- *Provides the graphical user interface to the repository*
- *eLicensing Workflow protects intellectual property*
- *Scientists can add new models, share models, and discuss model simulations*
- *Allows for execution of the models in the repository*

caBIG DMR Data Service

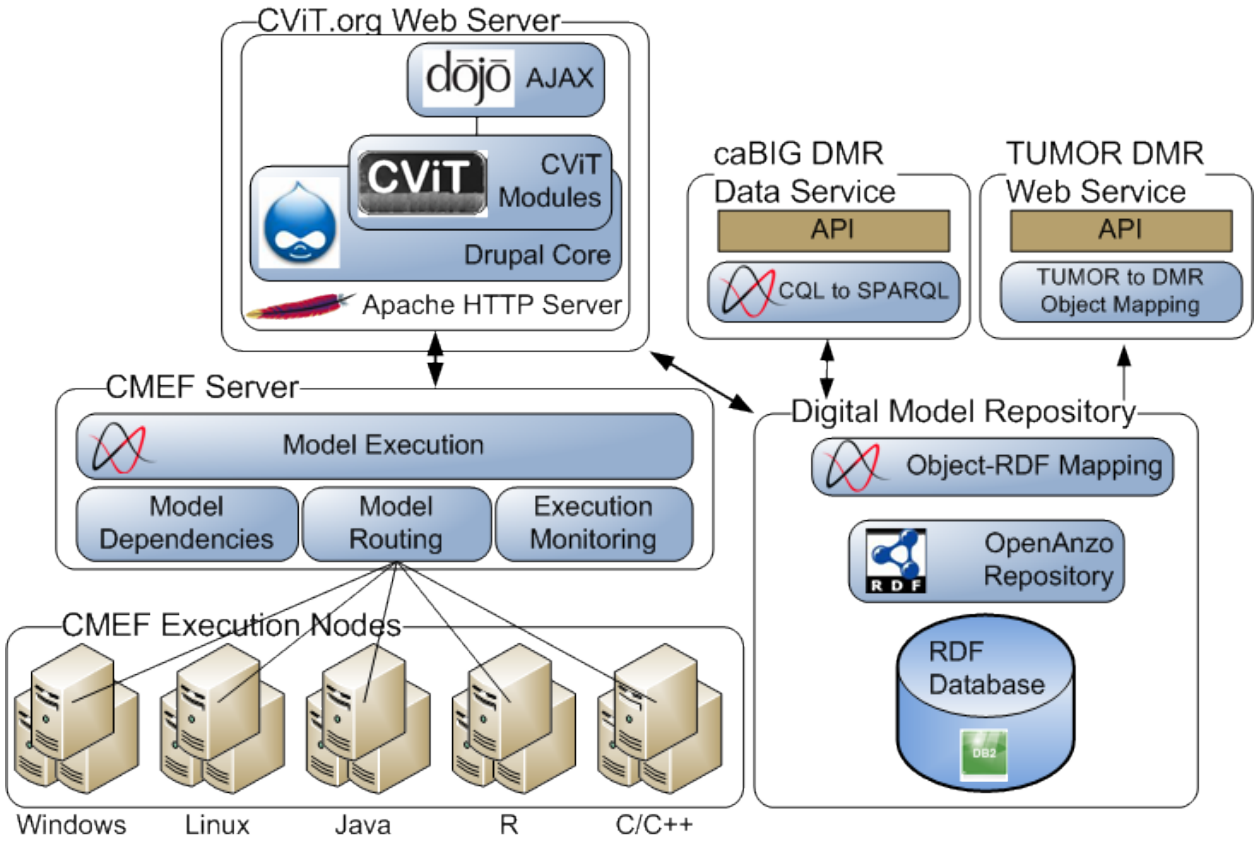
- *Silver-Level Compliant caBIG Data Service*
- *Securely access DMR through caGrid*
- *eLicensing Workflow preserved through caGrid*

Computational Model Execution Framework

- *Models annotated with execution metadata*
- *Grid-based execution of the models in the repository*

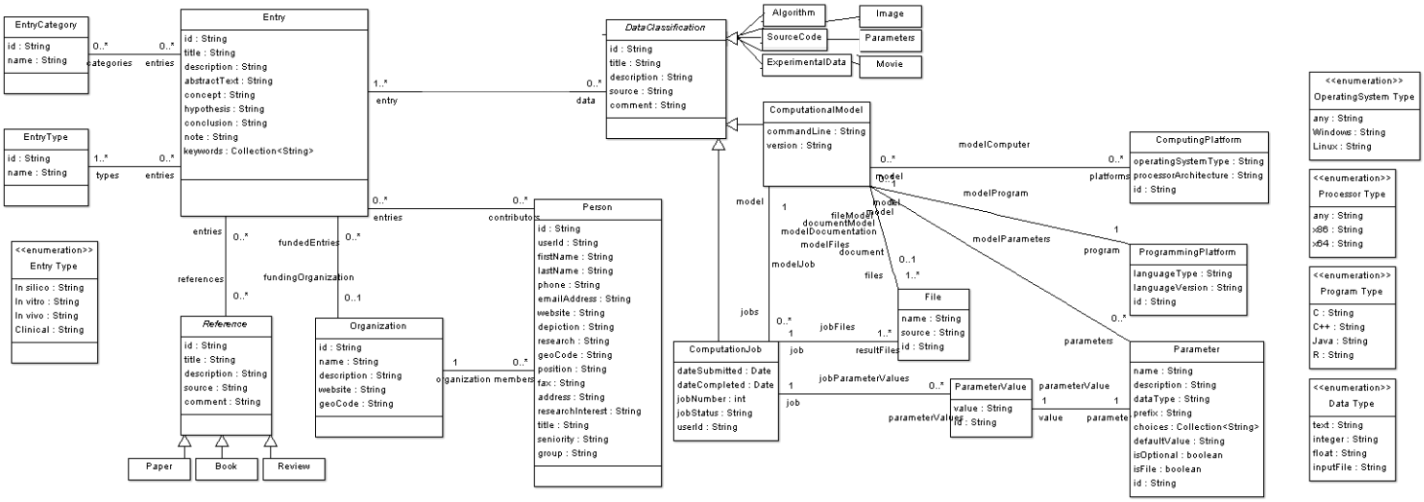
TUMOR Web Service

- *SOAP-RPC Web Service*
- *Access DMR Computational Models and Model Simulations*



MGH-CViT System Architecture

MGH-CViT DMR Domain Model



Entry	
Description	An entry in the repository contains relevant, uploaded information regarding a project. Entries can only be created by Principal Investigators (PIs). With a Licensing Officer's (LO's) approval a PI can have his entry published to other users of the repository. Designated contributors can annotate entries with metadata.
NCI Concept Code	C47885
Id	
Description	A unique entry identifier
Data Type	String
NCI Concept Code	C25364
Title	
Description	A name to identify the entry
Data Type	String
NCI Concept Code	C42774
Description	
Description	A paragraph explaining why the entry exists and what project it contains. This defines the scope for further data uploaded to the entry.
Data Type	String
NCI Concept Code	C25365
abstractText	
Description	A brief summary of the project's description
Data Type	String
NCI Concept Code	C60765
Concept	
Description	Background and basic idea of this project
Data Type	String
NCI Concept Code	C48910
Hypothesis	
Description	What assumption(s) will be proved by this experiment
Data Type	String
NCI Concept Code	C28362
Conclusion	
Description	The outcome and significance of the project
Data Type	String
NCI Concept Code	C54033
Note	
Description	Space for adding notes
Data Type	String
NCI Concept Code	C42619
Keywords	
Description	List of 3-5 searchable terms characterizing the entry
Data Type	Collection<String>
NCI Concept Code	C43513
Associations	
fundingOrganization	
Description	A source of funding for this particular entry
Data Type	Organization

NCI Concept Code	C19711, C17769
contributors	
Description	Repository users with write access to this entry
Data Type	Collection<Person>
NCI Concept Code	C25190
references	
Description	Upload, link to, or select (link within repository) references
Data Type	Collection<Reference>
NCI Concept Code	C25641
data	
Description	Data files can be included in an entry by way of upload, http reference, or internal repository reference
Data Type	Collection<DataClassification>
NCI Concept Code	C25474
types	
Description	Model entry classification denotes the origin of the data
Data Type	Collection<EntryType>
Value Domain	Computation | Invitro | Invivo | Clinical
NCI Concept Code	C25284, C25474, C47885
categories	
Description	A collection of values from the Entry Category enumeration describing the Entry.
Data Type	Collection<EntryClassification>
NCI Concept Code	C25372, C47885

Person	
Description	User profile
NCI Concept Code	C25190
id	
Description	A unique user identifier
Data Type	String
NCI Concept Code	C25364
userId	
Description	The persons DMR user id
Data Type	String
NCI Concept Code	C42694
title	
Description	Suffix titles. Usually Ph.D. or M.D.
Data Type	String
Value Domain	Ph.D. | M.D. | Sc.D.
NCI Concept Code	C25354
firstName	
Description	User's first name
Data Type	String
NCI Concept Code	C40974
lastName	

Description	User's last name
Data Type	String
NCI Concept Code	C40975
Phone	
Description	Phone number
Data Type	String
NCI Concept Code	C40978
emailAddress	
Description	Email address
Data Type	String
NCI Concept Code	C42775
Website	
Description	User's homepage (URL)
Data Type	String
NCI Concept Code	C19467
Depiction	
Description	User's icon/picture (URL)
Data Type	String
NCI Concept Code	C54273
Research	
Description	Type of research being done by user. (used in cvit.org/mashup)
Data Type	String
Value domain	experimental | computational | both | nci
NCI Concept Code	C25284, C15429
geocode	
Description	Coordinates for placing organizations on cvit.org/mashup
Data Type	String
NCI Concept Code	C25341, C68643, C68642
Position	
Description	Job title
Data Type	String
NCI Concept Code	C19067, C25193
Fax	
Description	Fax number
Data Type	String
NCI Concept Code	C42879
Address	
Description	Current mailing address
Data Type	String
NCI Concept Code	C70946
researchInterest	
Description	Several paragraphs describing current research interests
Data Type	String

NCI Concept Code	C48910, C15429
group	
Description	Tracks groups of cvit users for cvit.org/teampages
Data Type	String
Value Domain	Main | AdvisoryBoard | NCI | ICBP | Unlisted
NCI Concept Code	C41167
seniority	
Description	Tracks a user's expertise level
Data Type	String
Value Domain	Faculty | Postdoc | GradStudent | Other
NCI Concept Code	C25554, C25193
Associations	
entries	
Description	List of Entries for which this Person is listed as a contributor. See: Entry.contributors
Data Type	Collection<Entry>
NCI Concept Code	C47885
organization	
Description	Institution with which user is affiliated. Within the DMR, users can only belong to one organization.
Data Type	Organization
NCI Concept Code	C19711

EntryCategory	
Description	Enumerates a set of classifications to facilitate discovery and retrieval of entries. Each entry may be tagged with an Entry Category so that it can be found in the classification tree.
NCI Concept Code	C47885, C25372
id	
Description	A unique data classification identifier
Data Type	String
NCI Concept Code	C25364
name	
Description	The name for the category
Data Type	String
NCI Concept Code	C42614
Associations	
entries	
Description	List of Entries for which this EntryCategory is relevant
Data Type	Collection<Entry>
NCI Concept Code	C47885

EntryType	
Description	Model entry classification denotes the origin of the data. (Computation } Invitro | Invivo | Clinical)
NCI Concept Code	C47885, C25474, C25284
id	
Description	A unique data classification identifier

Data Type	String
NCI Concept Code	C25364
Name	
Description	The name of the Entry Type. One of In silico, In vitro, In vivo, Clinical
Data Type	String
NCI Concept Code	C42614
Associations	
Entries	
Description	List of Entries for which this EntryType is relevant.
Data Type	Collection<Entry>
NCI Concept Code	C47885

DataClassification	
Description	Data files can be included in an entry by way of upload, http reference, or internal repository reference.
NCI Concept Code	C25474,C25161
Id	
Description	A unique data classification identifier
Data Type	String
NCI Concept Code	C25364
Title	
Description	Name of the upload
Data Type	String
NCI Concept Code	C42774
Description	
Description	A brief description of the file's contents
Data Type	String
NCI Concept Code	C25365
Source	
Description	Link to the file itself (URL)
Data Type	String
NCI Concept Code	C42743
Comment	
Description	Any additional user input
Data Type	String
NCI Concept Code	C25393
Associations	
Entry	
Description	List of Entries for which this Data is relevant. See: Entry.data
Data Type	Collection<Entry>
NCI Concept Code	C47885

Algorithm: DataClassification	
Description	Tag indicates uploaded data is an algorithm
NCI Concept Code	C16275

SourceCode: DataClassification	
Description	Tag indicates uploaded data is source code
NCI Concept Code	C47901

Parameters: DataClassification	
Description	Tag indicates uploaded data is a set of parameters
NCI Concept Code	C48913

Image: DataClassification	
Description	Tag indicates uploaded data is an image
NCI Concept Code	C48179

Movie: DataClassification	
Description	Tag indicates uploaded data is a movie (e.g., simulation, microscopy)
NCI Concept Code	ObjectClassConceptCode: C75001

ExperimentalData: DataClassification	
Description	Tag indicates uploaded data is experimental data
NCI Concept Code	ObjectClassConceptCode: C25474
	ObjectClassQualifierConceptCode1: C42790

Reference	
Description	Link to a bibliographical reference. This can point to a PDF for upload, a PubMed id, or a reference already in the repository.
NCI Concept Code	C25641
id	
Description	A unique reference identifier
Data Type	String
NCI Concept Code	C25364
title	
Description	Name of reference
Data Type	String
NCI Concept Code	C42774
description	
Description	Details about/in the reference
Data Type	String
NCI Concept Code	C25365
source	
Description	Link to the reference file (URL)
Data Type	String
NCI Concept Code	C42743
comment	
Description	Additional comments from user
Data Type	String
NCI Concept Code	C25393
Associations	

Entries	
Description	List of Entries for which this Reference is relevant. See: Entry.reference
Data Type	Collection<Entry>
NCI Concept Code	C47885

Book: Reference	
Description	Tag indicates uploaded reference is a book
NCI Concept Code	C16360

Paper: Reference	
Description	Tag indicates uploaded reference is a paper
NCI Concept Code	C47902

Review: Reference	
Description	Tag indicates uploaded reference is a review
NCI Concept Code	C47902,C42729

Organization	
Description	An institution. Most of these will be colleges, universities, and research institutes. Each user in the repository is affiliated with one and only one organization. Each organization has one or more licensing officers to approve of user's licensing requests.
NCI Concept Code	C19711
Id	
Description	A unique organization identifier
Data Type	String
NCI Concept Code	C25364
Name	
Description	Name of the organization
Data Type	String
NCI Concept Code	C42614
Description	
Description	Brief description of the organization
Data Type	String
NCI Concept Code	C25365
Website	
Description	Organization's homepage (URL)
Data Type	String
NCI Concept Code	C19467
geocode	
Description	Geographical location of organization in “lat, long” format. Used in cvit.org/mashup
Data Type	String
NCI Concept Code	C25341, C68643, C68642
Associations	
Members	
Description	People associated with the Organization. Within the DMR, users can only belong to one Organization. See: Person.organization

Data Type	Collection<Person>
NCI Concept Code	C25190
fundedEntries	
Description	An entry funded by a particular organization
Data Type	Collection<Entry>
NCI Concept Code	C47885,C17769

ComputationalModel: DataClassification	
Description	Extends from DataClassification to encapsulate the content and metadata of an executable computational model
name	
Description	The name of the model
Data Type	String
description	
Description	Describes the operation of the model
Data Type	String
commandLine	
Description	Command line used to execute the model. Values enclosed in angle brackets are replaced by corresponding ParameterValues before the model is executed
Data Type	String
version	
Description	Software version of the computational model
Data Type	String
Associations	
modelFiles	
Description	Source files that constitute the computational model and requisite executable files. For example, a Java archive or a program configuration file
Data Type	Collection<File>
modelDocumentation	
Description	User's Guide or documentation describing the use of the computational model
Data Type	File
modelComputer	
Description	Operating system and CPU constraints required by the model
Data Type	ComputingPlatform
modelProgram	
Description	Program execution language constraints (Java, Perl, R, C++, etc.) required by the model
Data Type	ProgrammingPlatform
modelParameter	
Description	Program parameters that can be set for the model
Data Type	Collection<Parameter>

File	
Description	References a file
Name	
Description	The file's name
Data Type	String

Source	
Description	The file's location
Data Type	URL

ComputingPlatform	
Description	Describes model program execution hardware constraints
operatingSystemType	
Description	Indicates the operating system required to run the model (Any | Windows | Linux)
Data Type	OperatingSystem Type
processorArchitecture	
Description	Indicates the required CPU type to run the model(Any | x86 | x64)
Data Type	Processor Type

ProgrammingPlatform	
Description	Describes model program execution environment constraints
languageType	
Description	Classifies the programming language of the model (C | C++ | Java | R)
Data Type	Program Type
languageVersion	
Description	The minimum version of the programming language supported by the model
Data Type	String

Parameter	
Description	Defines the metadata for describing an individual input value for a computational model. Parameters may be input files or values entered on the command line. See ComputationalModel::commandLine
Name	
Description	The name of the parameter – for a file, it should be the file's name, for a command line parameter, it should match the name specified in commandLine, for example "<*name*>"
Data Type	String
Description	
Description	Description of the parameter or value constraints
Data Type	String
dataType	
Description	Defines the required parameter value type (Text | Integer | Float | File)
Data Type	Data Type
Prefix	
Description	A command line prefix that will be added if the value is present. For example "-F " or "-o "
Data Type	String
Choices	
Description	Set of values that constrain the input parameter
Data Type	Collection<String>
defaultValue	
Description	Default value used during model execution if no value is specified
Data Type	String
isOptional	

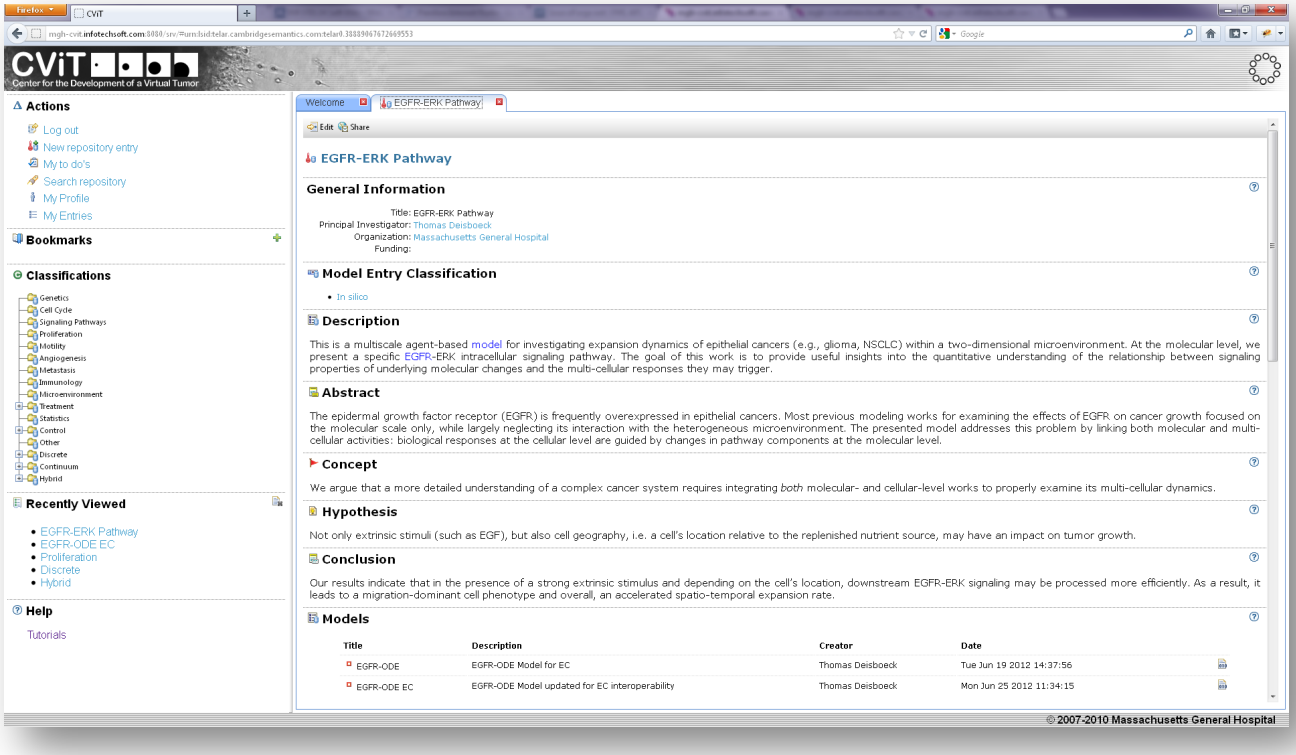
Description	Indicates that the parameter value can be omitted (if true)
Data Type	Boolean
isFile	
Description	Indicates that the parameter value is a file
Data Type	Boolean

ComputationJob: DataClassification	
Description	Extends from DataClassification to encapsulate the content and metadata of an executing (or executed) computational model
dateSubmitted	
Description	The date/time that the computation job was submitted for execution
Data Type	Date
dateCompleted	
Description	The date/time that the computation job execution completed
Data Type	Date
jobNumber	
Description	System-assigned number identifying the model execution job
Data Type	Integer
jobStatus	
Description	Result of running the model (Success | Failure)
Data Type	String
userId	
Description	Identifier of the user who initiated the ComputationJob
Data Type	String
Associations	
jobParameterValue	
Description	Values set for each Parameter of the model for the job
Data Type	Collection<ParameterValue>
jobFiles	
Description	Files produced by executing the model computation job, may include execution log, console output, and output files
Data Type	Collection<File>

ParameterValue	
Description	Value that will be utilized during the computation job for the given parameter
Value	
Description	Value that will be utilized during the computation job for the given parameter. Input files should be specified by URLs
Data Type	String
Associations	
parameterValue	
Description	Parameter that this value sets
Data Type	Parameter

MGH-CViT DMR Website

The CViT DMR website provides all of the functionality of the DMR-TUMOR Web Service, in additionally, it provides authenticated users with the ability to upload, share, and execute (on a back-end grid) computational cancer models.



MGH-CViT DMR TUMOR Web Service

PROVIDES READ-ONLY ACCESS TO AUTHORIZED MODELS IN DMR.

Based on the planned clinical scenarios (D2.1), the MGH-CViT Digital Model Repository must be extended to provide the required functionality to provide the models and model metadata stored in the CViT repository to the EU/TUMOR platform for remote execution. The TUMOR clinical scenario indicates a read-only interface to the models, metadata, and simulation results are sufficient to support the requirements. The TUMOR clinical scenarios do not envision altering the content of the CViT repository.

Web Service API

`try[] getEntries()`

Returns an array of all Entry objects in the DMR to which the user has access.

`Entry getEntry(String entryUID)`

Returns the Entry object corresponding to the provided entryUID.

`Reference getReference(String referenceUID)`

Returns the Reference (Article, Book, Abstract, etc.) object corresponding to the provided referenceUID.

`DataClassification[] getDataForEntry(String entryUID)`

Returns an array of all DataClassification (Algorithm, ComputationalModel, ComputationJob, etc.) objects associated with the Entry with the given entryUID.

`ComputationalModel[] getModelsForEntry(String entryUID)`

Returns an array of all ComputationalModel objects associated with the Entry with the given entryUID.

`ComputationJob[] getJobsForModel(String modelUID)`

Returns an array of all ComputationJobs objects associated with the Model with

TUMORService	
Description	The MGH-CViT Digital Model Repository provides authenticated access to computational models, model metadata, and model simulation results through a TUMOR-specific Web Service API. The TUMOR Web Service API extends the DMR caBIG Data Service API by providing additional methods to retrieve computational models and computation job results executed by the Computational Model Execution Framework.

getEntries
Enumerates the set of Entries in the DMR that the user has authorization to read.
Parameters
none
Returns
Entry[] – list of all Entries in the repository, filtered by user authorization
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getEntry
Returns the Entry specified by the provided Entry UID.
Parameters
String entryUid – unique Entry identifier
Returns
Entry – specific Entry identified by entryUid, null if Entry does not exist
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getDataForEntry
Enumerates the set of DataClassification associated with the given Entry UID that the user has authorization to read.
Parameters
String entryUid – unique Entry identifier
Returns
DataClassifcation[] – array of DataClassification sub-classes that are associated with the specified Entry, filtered by user authorization
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getData
Returns the DataClassification specified by the provided DataClassification UID
Parameters
String dataClassificationUid – unique DataClassification identifier
Returns
DataClassification – DataClassification subclass identified by dataClassificationUid
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getModelsForEntry
Enumerates the set of ComputationalModel associated with the given Entry UID that the user has authorization to read.
Parameters
String entryUid – unique Entry identifier
Returns
ComputationalModel[] – array of ComputationalModel classes that are associated with the specified Entry, filtered by user authorization

Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getModel
Returns the ComputationalModel specified by the provided ComputationalModel UID.
Parameters
String computationalModelUid – unique ComputationalModel identifier
Returns
ComputationalModel – ComputationalModel identified by computationalModelUid
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getJobsForModel
Enumerates the set of ComputationJob associated with the given ComputationalModel UID that the user has authorization to read.
Parameters
String computationalModelUid – unique ComputationalModel identifier
Returns
ComputationJob[] – array of ComputationJob classes associated with the specified ComputationalModel, filtered by user authorization
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getJob
Returns the ComputationJob specified by the provided ComputationJob UID
Parameters
String computationJobUid – unique ComputationJob identifier
Returns
ComputationJob – ComputationJob identified by computationJobUid
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getReference
Returns the Reference specified by the provided Reference UID
Parameters

String referenceUid – unique Reference identifier
Returns
Reference – Reference sub-class identified by referenceUid
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

getOrganization
Returns the Organization specified by the provided Organization UID
Parameters
String organizationUid – unique Organization identifier.
Returns
Organization – Organization identified by organizationUid, null if Organization does not exist.
Exceptions
LoginException – if authenticated session has not been established.
AuthorizationException – if User does not have sufficient privilege to perform this action.

Web Service Clients

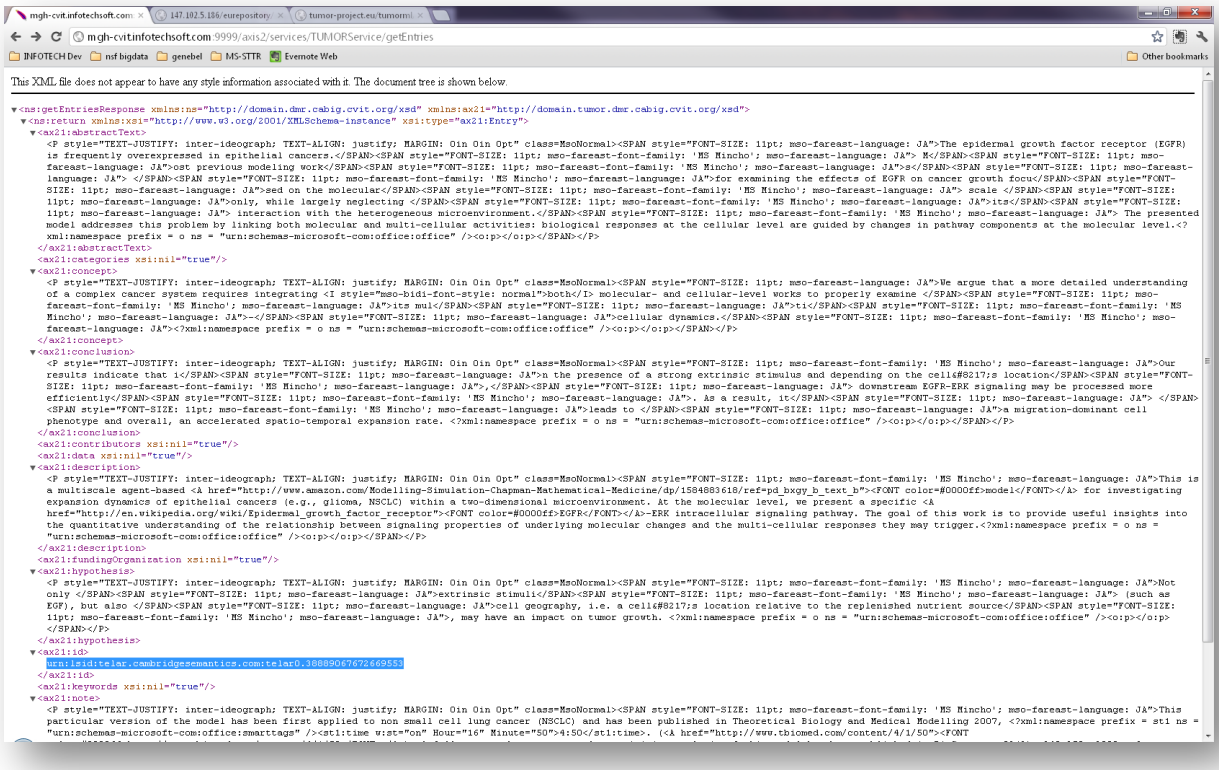
DMR-TUMOR HTTP Service Access

Access to the DMR TUMOR Web Service through a Web Browser. Results are returned in XML and can be viewed within the browser.

Web Service URL: http://mgh-cvit.infotechsoft.com:9999/axis2/services/TUMORService

Example Workflow for Retrieving a Cancer Model Executable

1. Call getEntries – to list the available Entries and locate the Entry of interest
2. Call getModelsForEntry – to identify the Computational Model of interest attached to the Entry
3. Call getModel – to retrieve the full content of the Computational Model
4. Download model files to the local computer
5. Run the downloaded model executable



DMR-TUMOR Java Client Access API (tumor-client-1.0.0.zip)

Java source and compiled code to access the DMR-TUMOR Web Service through Java.

Class: org.cvit.cabig.dmr.tumor.client.TUMORServiceStub

Java client API for accessing the DMR-TUMOR Web Service.

Package: org.cvit.cabig.dmr.tumor.domain

Domain classes for the DMR-TUMOR Java client. Correspond to the classes in the DMR Domain Model.

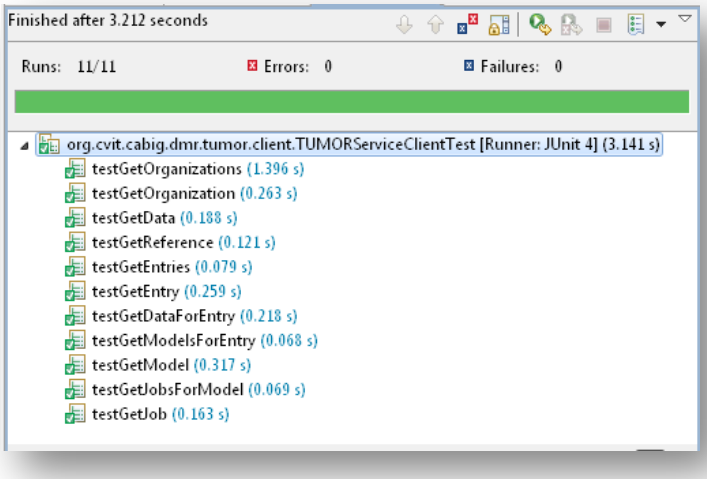
Package: org.cvit.cabig.dmr.tumor.client.soap

SOAP classes for accessing SOAP request/response documents.

DMR-TUMOR Java Client Access JUnit Tests

Class: org.cvit.cabig.dmr.tumor.client.TUMORServiceClientTest

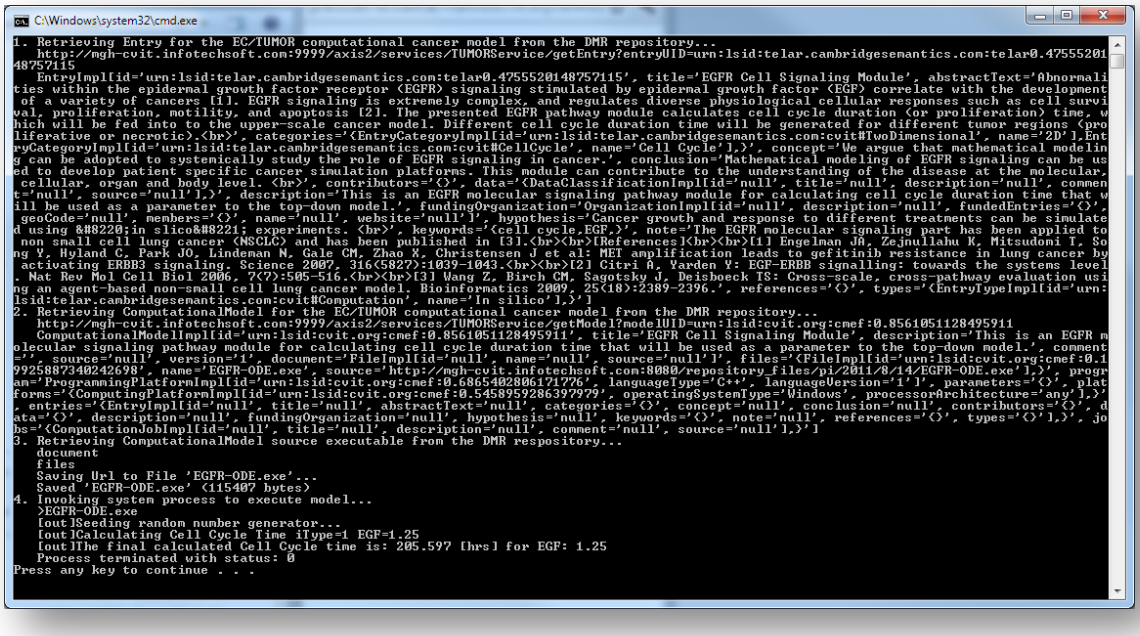
JUnit tests of the Java Client Access API to verify that Java Client API is functioning properly.



DMR-TUMOR Java Client Access Demonstration

Class: org.cvit.cabig.dmr.tumor.client.TUMORServiceClientMain

Command-line program that demonstrates the use of the DMR-TUMOR Web Service to retrieve and utilize DMR content.



5 Appendix I - Abbreviations and acronyms

ACGT – Advancing Clinico Genomic Trials on Cancer

caBIG – cancer Biomedical Informatics Grid

ContraCancrum - Clinically Oriented Translational Cancer Multilevel Modelling

CViT – Center for the Development of a Virtual Tumor

DMR – Digital Model Repository

HTTP – HyperText Transfer Protocol

MGH – Massachusetts General Hospital

NCI – National Cancer Institute

NIH – National Institute of Health

REST – Representational State Transfer

TUMOR – Transatlantic Tumor Model Repositories

TumorML – The TUMOR Markup Language

UID – Unique Identifier

URL – Uniform Resource Locator