Deliverable No. 2.1

State of the art knowledge for building hypermodels

Grant Agreement No.:	600841
Deliverable No.:	D2.1
Deliverable Name:	State of the art knowledge for building hypermodels
Contractual Submission Date:	30/11/2013
Actual Submission Date:	20/01/2014

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	**CHIC**
Project Full Name:	Computational Horizons In Cancer (CHIC): Developing Meta- and Hyper-Multiscale Models and Repositories for In Silico Oncology
Deliverable No.:	D2.1
Document name:	State of the art knowledge for building hypermodels
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	5
Actual Submission Date:	20/01/2014
Editor: Institution: E-Mail:	Kostas Marias FORTH kmarias@ics.forth.gr

ABSTRACT:

This deliverable provides a review of current knowledge for building cancer hypermodels. To do this we focus on the systems biology, the engineering and the software integration standpoints and explain with simple examples basic concepts/technologies that are necessary in cancer hypermodelling design and implementation. To this end a number of technologies are explained and previous relevant EC projects that dealt with hypermodelling are shortly described.

KEYWORD LIST:

State of the art, models, hypermodelling, computational biology, service composition, clinical hypermodelling scenarios.

---

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

MODIFICATION CONTROL			
Version	**Date**	**Status**	**Author**
1.0	01/11/2013	ToC	Kostas Marias
2.0	09/01/2014	1st draft document, compiled from partner contributions.	Kostas Marias
3.0	14/01/2014	2nd draft, Integration of additional contributions.	Kostas Marias
4.0	15/01/2014	Pre-final, circulated for internal review.	Kostas Marias
5.0	20/01/2014	Final version, submitted to EC.	Kostas Marias

List of contributors

- Kostas Marias, FORTH

- Giorgos Zacharioudakis, FORTH

- Stelios Sfakianakis, FORTH

- Georgios Stamatakos, ICCS

- Eleftherios Ouzounoglou, ICCS

- Fay Misichroni, ICCS

- Dimitra Dionysiou, ICCS

- Norbert Graf, USAAR

- Marco Viceconti, USFD

- Daniele Tartarini, USFD

- Debora Testi, CINECA

- Marc Stauch, LUH

Contents

1 Executive Summary

This deliverable is part of Task 2.1: State of the Art of Knowledge for building hypermodels (M1-8). The purpose of this particular deliverable is to provide a review of current knowledge for building cancer hypermodels. To do this we focus on the systems biology, the engineering and the software integration standpoints and explain with simple examples basic concepts/technologies that are necessary in cancer hypermodelling design and implementation. To this end a number of technologies are explained in detail and previous relevant EC projects that dealt with hypermodelling are shortly described.

2 Introduction

2.1 *Purpose of this document*

In the last years it was realized that in order to provide a comprehensive and in depth mathematical-computational description and virtual reproduction of several phenomena or aspects of normal human biology and disease, a combination of already existing and trustable models with new ones is necessary. To this end several efforts have been made and appeared in the literature [1]-[30].

Due to the high complexity and the multiscale character of biological phenomena, numerous mathematical and computational models of normal physiology, disease and treatment response rely on the combination of models of simpler phenomena or constituent biomechanisms. In this way certain kinds of hypermodels i.e. integrative or composite models have already appeared in the literature even if the term "hypermodelling" may not been explicitly utilized. In most cases the combination of simpler component models or hypomodels has led to the development of rather "monolithic" composite models i.e. of models which could not be easily decomposed so that their component models might be reused for the building of other new (hyper)models.

A number of efforts to decompose a complex biological phenomenon or a composite model of it into its crucial components (hypomodels) and then reconstruct the composite model in a well-designed, formal and reproducible way have appeared in literature[1][2][11][26][27][28][29][30]. Supportive technologies facilitating the building of hypermodels have also been proposed [27, 29]. However, to the best of our knowledge no semantic annotation has been utilised so far in order to characterize models in a standardized way. Such an element would considerably facilitate both the mining of already available models that may have been eventually developed by different modellers and their semi-automatic linking. One of the goals of the CHIC project is to address these issues.

From the domain perspective, CHIC will focus on cancer and oncology which constitute perhaps the most extreme paradigm of multiscaleness and complexity in medicine. The highly complex interplay of the various scales in cancer and treatment response dictates both advanced basic science approaches and advanced technologies for hypermodelling. This deliverable has the purpose to shed light into the cancer hypermodelling design and implementation starting from the biological phenomenon related to cancer to the engineering design and software implementation and connection of elementary models to hypermodels. The document is not meant to be exhaustive in all aspects and technologies; rather its purpose is to introduce the hypermodelling process and offer a state-of-the-art report on main technologies involved which might be taken into consideration also in the CHIC implementation.

2.2 *Structure of this document*

The document first focuses on explaining basic concepts based on cancer phenomena of systems biology with separate references to the molecular and the tissue level. Then in the next sections we present the engineering design perspective (where an example of hypermodelling design is presented) and the software integration perspective where we also list several technologies that are necessary for this integrative task. The deliverable concludes with a short account of the CHIC hypermodelling scenarios and the related previous projects (VPHOP and TUMOR) related to hypermodelling implementations.

3 Systems Biology perspective

Hanahan and Weinberg [31] proposed in 2000 (updated 2011 [32]) that the diversity of cancer and its underlying molecular mechanisms can be explained by ten biological processes, including cell adhesion and motility, signalling, transcriptional regulation, cellular metabolism, and intracellular trafficking and others. These molecular and cellular changes are called the 'Hallmarks of Cancer'. The concept of 'Hallmarks of cancer' is a powerful guide for translational research not only for drug development but also for early detection and the development of new and more targeted therapies that have fewer side effects and enhance the quality of life of cancer patients. This concept is explained in chapter 2.3 of the deliverable D2.2 in more detail. Most important for understanding the complex molecular nature of cancer are the following four statements [33]:

1. Cancer cells must acquire modifications in most of the 10 hallmarks if they are to develop and evolve towards a malignant, invasive state. This requires functional changes in multiple pathways.
2. Only stem and progenitor cells with a high plasticity may be able to sustain a coordinated perturbation of these different hallmarks.
3. Only a fraction of cells within a lesion can progress towards invasion and metastasis.
4. Cancer development involves interactions between cancer cells and their microenvironment, including inflammation and immune responses.

Based on these summarized principles the following sections discuss the modeling strategies and challenges at different scale levels.

3.1 *Molecular/Cellular level modeling*

Cancer cells are characterized by numerous mutations in the genome. Not all of these mutations are significant for cancer progression. A subset of them, often termed driver mutations, show a distinctive fitness advantage resulting in pathway aberrations related to the above mentioned 10 hallmarks of cancer. To simulate the behaviour of cancer cells the intracellular signalling network and the interaction of cancer cells with the microenvironment needs to be modelled. **Each of the sub-circuits as well as the interaction with the microenvironment can serve as a component model that is regarded as a model on the molecular scale**. A hypermodel of a cancer cell build from these component models will be up-scaled to the cellular level.

'One of the grand challenges of the understanding of cancer progression is to find mechanistic links between such alterations and the hallmarks of cancers such as increased proliferation and survival, aggressive invasion and metastasis, evasion of cell death, and increased metabolism. This challenge is also of quintessential clinical importance because patient outcome to therapy (both in terms of initial response to therapy and subsequent development of resistance to therapy) is now shown to depend on the genetic alterations (primary or acquired) in the individual patients. Traditional methods in cell biology and cancer biology such as phosphor-proteomics, immuno-precipitation, polymerase chain reaction, in-situ hybridization and molecular imaging, and direct sequencing, along with network-based theories and bioinformatics are reasonably poised to probe some of these altered traits, such as those connected with signalling, transcriptional regulation, and cellular metabolism, but are not directly amenable to dissect the underlying complexity of a cancer cell or a tumor tissue [35].

3.1.1 State of the art on building hypermodels on the Atomic and Molecular Level

During the last decades, due to the great advancements in spectroscopy ([36], [37]) and high-throughput molecular screening [38] (advancement in genomics, proteomics, metabolomics etc.), a plethora of datasets and knowledge regarding the molecular and cellular biology have emerged. This abundance in data and the heterogeneity among their types and qualities, resulted in the emergence of the fields of Bioinformatics and Systems Biology aiming at handling and analysing the available data and achieving a system-level understanding of living organisms by combining the existing and the additionally extracted knowledge.

However, despite the integrative philosophy beyond bioinformatics and systems biology, distinct research fields of computational biology have emerged in order for the complexity of the underlying biological phenomena to be studied in detail. These fields, which are going to be studied in the context of CHIC project, together with a short description of their scope and the commonly used software tools, standards and methods are listed below:

1. **Structural Bioinformatics and Molecular Interaction Simulation**

   Structural Bioinformatics (or Molecular Dynamics Modeling and Simulation) are related to the analysis and prediction of the three-dimensional structure of biological macromolecules such as proteins, RNA, and DNA. Some representative tools for this type of Analyses are NAMD[39], VMD [40] and Carma[41].

   Molecular Interaction Simulation (or Docking Simulation) aims to *in-silico* simulate the interactions between (macro)-molecules of known 3D structure. For example, a prediction of how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure could be given by tools like AutoDock Vina [42] and Glide[43]. Well known resources for the 3D structure of (macro)-molecules and their interaction and binding properties are RCSB PDB[44], ZINC[45], PubChem[46], PDBbind[47].

2. **Data Driven Network and Predictive Models**

   In data-driven network and predictive modeling, computational algorithms are used in order for large-scale data (high-throughput and time course experimental data) to be analysed and causal relationships among molecular entities to be inferred. An example of such an analysis is the searching of patterns in gene expression profiles that may distinguish patients with different prognosis or drug-sensitivity properties. More advanced analyses may integrate different modalities of molecular data (for example genome-scale DNA variation data, gene expression data, protein-protein interaction data etc.) in order to predict probabilistic, causal networks. In this field, standard tools and exchange formats are not yet well established and usually more general software tools like R [48] statistical software and MATLAB [49] are used.

3. **Mechanistic Modeling and Simulation (sub-cellular level)**

   In contrast with data-driven network inference, mechanistic models of interaction networks (also referred as biomolecular networks or pathway models) are created by a manual or semi-manual procedure (deep curation) by integrating knowledge from publications, databases (e.g. KEGG[50], Panther [51] and Reactome [52] pathway databases) and high-throughput data, letting the creator to introduce also mechanistic details of the molecular mechanisms. In this field, specific standards are defined in order for models to be represented. Systems Biology Markup Language (SBML) [53] and the Biological Pathways exchange (BIOPAX) [54] formats were designed to represent pathway models from different perspectives. Moreover, Systems Biology Graphical Notation is used to standardize a human-readable pathway notation. Additionally, Minimum Information Required in the Annotation

of Models (MIRIAM) [55] defines the rules for model annotation by external resources (Ontologies). Some commonly used tools for the creation of pathway models by deep curation are Cell Designer[56], Edinburgh Pathway Editor [57] and PathVisio[58].

Since in mechanistic pathway modeling a deep curation procedure is followed, the causality, the stoichiometry and the mechanisms of interactions could be introduced, it is possible for pathway models, instead of only providing a static picture, to be dynamically simulated. Ordinary Differential Equations (ODEs) have been used widely to model the dynamic behaviour of biological systems and pathway models defined in well-established markup languages like SBML and CellML [59] could be used to predict the dynamics of the pathway by using compatible simulation tools such as COPASI[60]. Moreover the Systems Biology Workbench [61] allows multiple application, such as software packages for modeling, simulation, analysis and visualization, to communicate with each other (for example Cell Designer with COPASI).

Although the above presented fields refer only to the atomic and molecular scales of bio-complexity (or more generally at the sub-cellular scale), **there is a clear need for combination of models, or equivalently, for the creation of multi-scale models that could be also thought as hypermodels**. These combinations may be done using either models of the same type or models referring to biological phenomena that are manifested at different scales. Two examples, one with intra-level combination and one with inter-level combination are given below:

1. **Linking Molecular Structure to Signaling Networks**

   A representative example of combining models of structural bioinformatics, molecular interaction and mechanistic modeling is given in [62]. Briefly, the target of this multi-scale modeling effort was to identify the consequences of a mutation in the Epidermal Growth Factor Receptor on the dynamics of the EGFR downstream signalling network. The combined models, together with the exchanged information between models, are shown in Figure 1.

Altered EGFR
Polypeptide
Sequence

Structural Model

Mutant
EGFR
Structural
Information

Molecular
Interaction
(Docking) Model

Predicted
ligand
binding
Kinetic
parameters

Signaling EGFR
Pathway Model
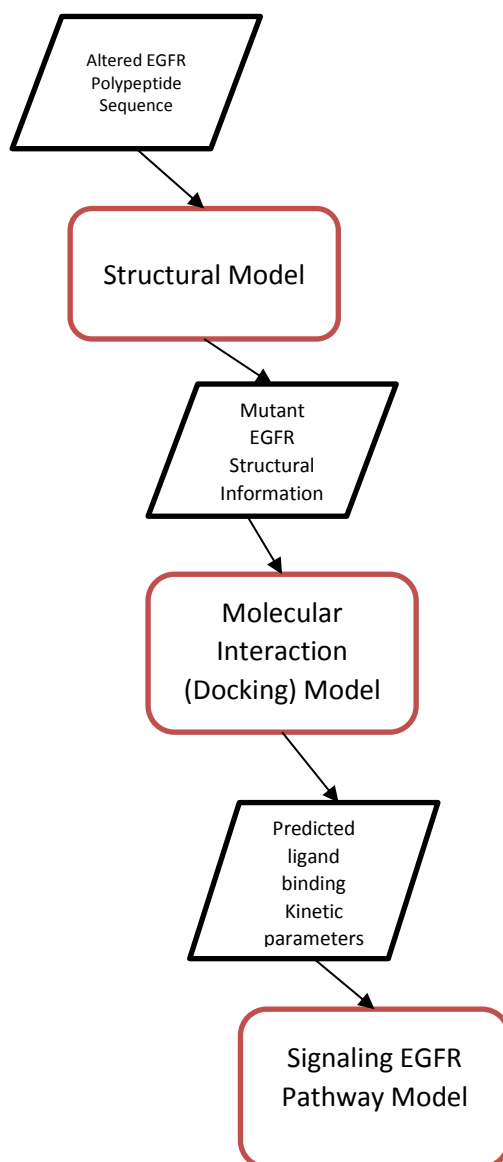
Figure 1: A multi-scale model linking the structural bioinformatics, molecular interaction dynamics and signalling pathway levels[3].

2. Merging two Pathway Models

Since mechanistic models of pathways are created by different researchers, having as base different knowledge resources and adopting different assumptions, similar but distinct models for the same pathway may be developed. Moreover, since in nature, cross-talk between biological mechanisms does exist, this crosstalk may not be captured by a pathway model that focuses on a specific biological process. Two abstract examples are given in Figure 2 and Figure 3.

---

[3] The component models are depicted by orange edge rectangles. The exchanged information between the levels is depicted by black edge parallelograms.

Pathway A, model 1

Pathway A, model 2
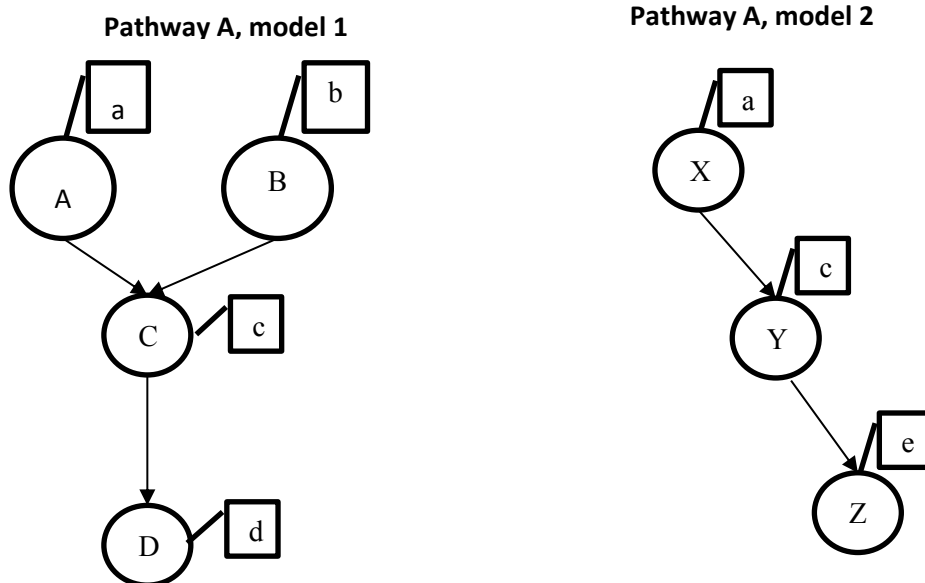
Figure 2: A case where two different models refer to the same pathway[4].

Pathway C model
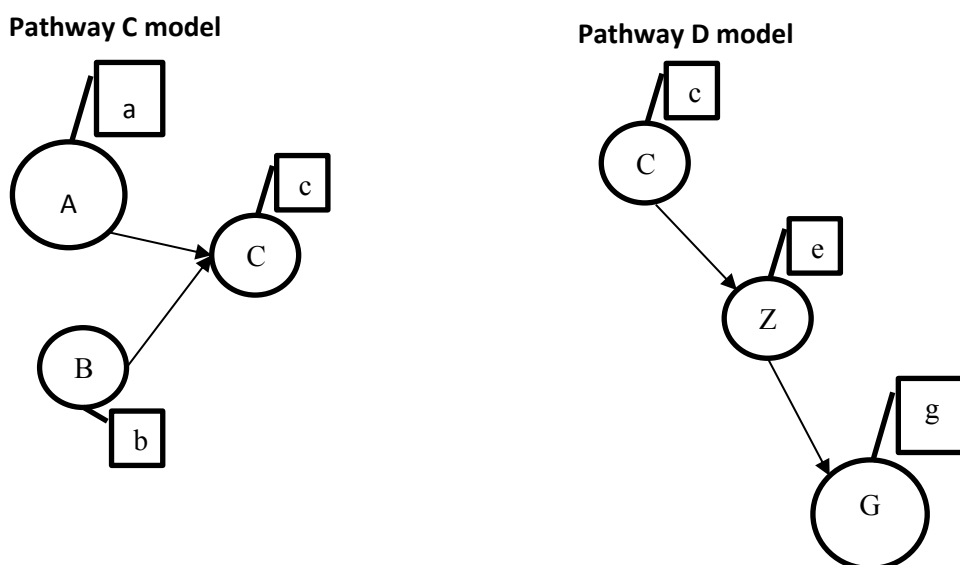
Pathway D model

Figure 3: A case where two models of different pathways show a cross-talk sharing the node C[5].

Therefore, a necessity for combining or merging pathway models has arisen as the number of available pathway models has expanded. Although these procedures may be done manually, they could be also assisted by specifically developed methods and tools. A well-established tool for

---

[4] The pathway components (usually proteins) are depicted by circles. Their semantic annotations are depicted by rectangular callouts.

[5] The pathway components (usually proteins) are depicted by circles. Their semantic annotations are depicted by rectangular callouts.

pathway models described in SBML is SemanticSBML[63]. SemanticSBML compares the MIRIAM annotations of two or more input models and suggests a preliminary version of the merged model, which then provides a starting point for manually completing the element matching, a procedure that is further assisted by highlighting the possible conflicts (e.g. different initial concentrations for a species or species without annotation).

Another modeling research field that increasingly gains interest, is the development of physiological models, which refer to levels of biocomplexity higher than the molecular level (cellular, tissue, organ and whole body levels) that would be linked with underlying pathway models and models from the fields of structural bioinformatics and molecular interaction simulation. **The International Union of Physiological Science (IUPS), the Physiome Project[64], the Virtual Physiological Human Project (VPH) [65] and the High-Definition Physiology (HD-Physiology) project [66] are initiatives that aim to promote basic science and to provide technological solutions for integrated physiological models.**

In the field of physiological modeling, an agreed standard for defining physiological functions and interconnections between models of multiple levels and for performing simulations has not been created so far. However, CellML is a pioneering effort to define a markup language for describing physiology models.

Moreover, in the HD-Physiology project, the PhysioDesigner [67] tool is used to model multi-scale physiological models. PhysioDesigner has been built based on the specifications of In-Silico Markup Language (ISML) [68] which is an emerging standard (XML markup language) for multi-level physiological modeling. The models defined in ISML are composed of modules that refer to the elements constructing a model and edges that define the structural and functional relationships among modules. Inside each module, several dynamical variables, constants, time-dependent parameters and morphology data, could be defined as physical quantities and the dynamics of these quantities could be explicitly described in MathML. A functional relationship between two modules could be defined by a functional edge linking an out-port of a module to an in-port of another module. Structural edges represent an ontology-like relationship among modules such as "has a" relationship. These structural edges, in terms of physiology may correspond to properties such as "constitute" (many cells constitute an organ) or "include" (a cell membrane includes organelles) and so on. Finally, ISML is compatible with CellML and models defined in SBML could be integrated in modules defined in ISML.

Finally, in the context of the VPH project Ricordo [69] the semantics-based multiscale model-description architecture SemSim [70] and the SemGen [71] tool that supports the SemSim for automating the modular composition and decomposition of biosimulation models have been defined. By converting models defined in SBML, CellML and MML declarative languages (that could be downloaded from BioModels, CellML model repository or physiome.org model repository or created from scratch) to interoperable SemSim models (which is an OWL representation of the model containing semantic information about a model contents in addition to all of its computational aspects), users are able to merge or decompose models based on their semantic meta-information. The basic workflow steps of this procedure are given in Figure 4.
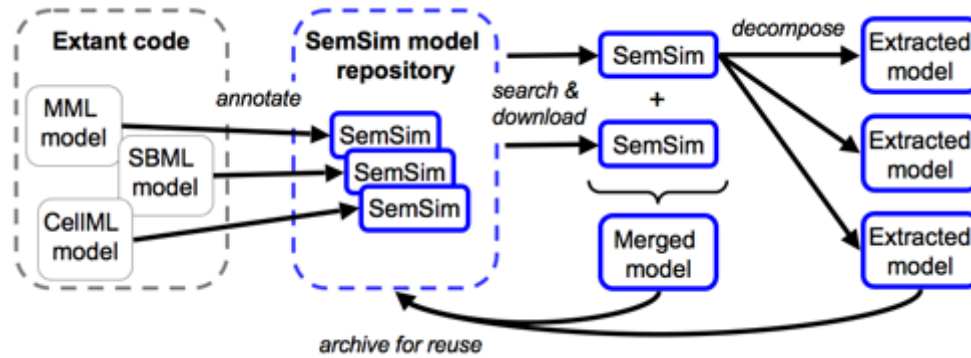
Figure 4: The workflow of SemSim procedure (accessible in (SemSim).

3.2 Tissue level

In addition to the above-described hallmarks of cancer the heterogeneity of tumor cells in a single cancer in a single patient is a further challenge. By collecting a tiny piece of a biological specimen from a tumor, one can question how representative this specimen is for the whole tumor under investigation. As we know today most of the cancers are heterogeneous. Therefore it is important to find the driver mutations and the most important deregulated pathways in a single tumor.

The tremendous advances achieved in the understanding of cancer biology have delivered unprecedented progress in molecularly targeted cancer therapy in the past decade. The fast growing category of targeted anticancer agents available for clinical use is accompanied by a conceptual revolution in anticancer drug development (see Figure 5). Nevertheless, molecularly targeted cancer therapy remains challenged by a high failure rate and an extremely small proportion of patients that can benefit [35]. This is mainly related to the heterogeneity within a single cancer. 'Uncontrolled cell division, which is required for full-blown malignancies, causes higher incidence of genetic instability arising from replication errors and increases opportunities for the emergence of multiple mutants. This genetic heterogeneity translates into phenotypic and functional heterogeneity, leading to coexistence of genetically divergent tumor cell clones. In addition, a substantial fraction of non-heritable phenotypic heterogeneity can arise from differentiation of cancer stem cells and morphological and epigenetic plasticity, driven by the selective evolutionary pressure from micro-environmental cues' [35]. As such treatment with a single targeted agent, may not be sufficient to treat a genetically heterogeneous tumor.
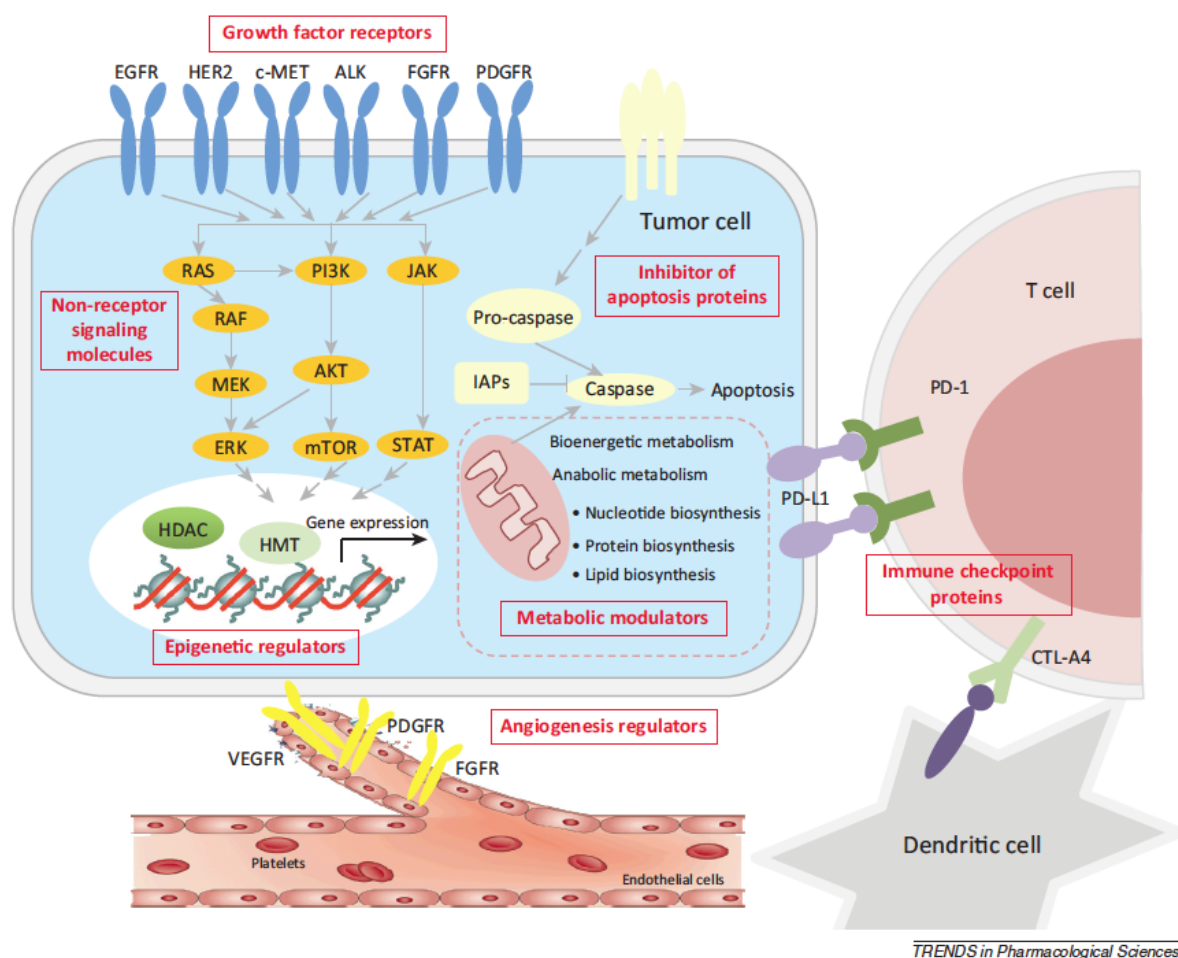
Figure 5: Emerging new targets for molecularly targeted cancer therapy[6].

In addition the crosstalk between tumor cells and the one between tumor cells and the immune system needs to be taken into consideration. A hypermodel as described in the chapter 4.1, representing one cancer cell, can serve as a component model, as many of these component models will build a hypermodel of a tumor on the tissue scale. This needs to include the crosstalk between tumor cells. Modeling the interaction of the tumor with body components like the immune or endocrine system will put such a hypermodel on the body scale.

Huang et al stated in 2013: 'We are experiencing a new era of personalized cancer medicine for cancer therapy. The precise identification of molecular drivers of cancer malignancy constitutes the basis for personalized therapy. Future efforts are anticipated to improve comprehensive assessment of molecularly based subsets of heterogeneous cancer, which requires technological advances in next-generation sequencing (NGS) with a high rate of output, new bioinformatics capabilities for handling large quantities of data, and collaboration among multidisciplinary scientists. This effort will pave the way for precise combinatorial approaches by specifically targeting each heterogenic

---

[6] After over decade-long development, growth factor receptors and downstream non-receptor signalling continue to be the most actively explored targets for drug discovery. A few new fields have emerged lately as a resource of promising targets including cancer metabolism and epigenetic modulation. In parallel, long-pursued targets such as inhibitor of apoptotic proteins remain being actively investigated but the strategy may have varied. Cancer immunotherapy, in particular targeting immune checkpoint proteins, probably represents the most promising field for targeted cancer discovery. Abbreviations: HDAC, histone deacetylase; HMT, histone methyltransferase; IAPs, inhibitors of apoptosis proteins. (Picture and legend taken from: [35])

composition. Meanwhile, clinical success of personalized therapy requires rationally guided clinical practice, where the integration of biomarker sets should provide the standard for adaptive clinical design. The co-development of predictive and response biomarkers along the entire drug discovery and development path will require deep insights into molecular mechanisms and explorations in suitable preclinical models. Outstanding questions for further research in the field are presented in the following list:

1. How can we best select 'biomarker sets' and properly apply them in clinical treatment of patients to identify optimal target patient subsets, to predict a patient's response, resistance, and toxicity, and to rapidly distinguish between responders and non-responders?
2. Is it possible to screen biomarkers using non-invasive approaches, such as circulating tumor cells, circulating DNA, cytokines, and chemokines? If not, how can we make technical breakthroughs to fully interpret the information of very limited patients' biopsies?
3. Is biomarker-based combinational therapy, that is, a 'cocktail' of highly-specific targeted drugs customized to individual patients according to their genetic aberrations, sufficient to largely overcome the resistance of targeted therapy?
4. How can innovative biomarker-based clinical design, that is, stratification of patients, assignment of specific drug therapy, and adaptive trial designs, increase the translation of targeted drugs from bench to bedside?
5. Given that the tumor microenvironment has an enormous impact on tumor development, how can we develop models that accurately reflect the tumor microenvironment, in particular the human immune system, for drug discovery?' [35]

4 Hypermodel engineering perspective

4.1 Definitions

Hypermodelling, integrative modeling, or multiscale modeling is a new emerging area of computational sciences and engineering. It is being driven by a variety of scientific domains, including mechanical civil, and chemical engineering, materials science, high-energy physics, biomedicine, meteorology, etc. Each domain uses its own terminology, and even its epistemological perspective. Thus, as a first step we propose a set of definitions that will be used in the CHIC project; then we briefly review the relevant state of the art, reconciling the various terminologies to these definitions.

4.1.1 Modeling

We observe nature, and we notice recurrences. We develop causal knowledge, first by induction, associating the current observable states to predicted future states, and then by inferring why such causal relation exists, recognising some fundamental principles, and then by deduction derive from these principles mechanistic explanations of the observations.

In this context we can define *scientific models* as: "finalized cognitive constructs of finite complexity that idealize an infinitely complex portion of reality through idealizations that contribute to the achievement of knowledge on that portion of reality that is objective, shareable, reliable and verifiable"[72].

4.1.2 Resources, data, and models

In this logical framework we have *Data*, which are the state quantities we use to describe the biological process of interest, and *Models*, which encapsulate some reductionist knowledge about that process. We refer to both entities with the general term of *Resource*. This makes possible to propose a logical taxonomy:

1. Resource

    a. Data: factual information, whether observed or predicted.

        i. Observed: generated through observation, measurement, etc.

        ii. Predicted: generated through speculative reasoning informed by existing knowledge

    b. Model: speculative information that represent the existing knowledge.

        i. Phenomenological: models capture predominantly knowledge generated inductively, by analysis of available data. Relies on implicit idealisations such as regularity, smoothness, etc.

        Mechanistic: models that capture predominantly knowledge generated deductively. Relies in explicit idealisations.

4.1.3 Models, hypomodels, and hypermodels

Because nature is infinitely complex, we conduct this process by "reducing" our attention to particular portions of nature, observed at particular characteristic space-time scales, assuming these

reductions are somehow independent by the rest of the nature. This process is called "reductionism", and it is the foundation of modern science.

But when we reduce we commit an error; the more entangled is the system we are looking at, the bigger is this error. In many biological processes by neglecting the systemic interaction across space-time scales we miss fundamental mechanisms; thus it is necessary to "recompose" the fragments of knowledge we produced at each specific space-time scale into systemic representations of the biological process of interest.

Scientific models are a very effective way to capture the reductionist causal knowledge we develop over a given biological process. Not only they are falsifiable as they use such knowledge to predict future observable states, but in principle they can be composed to other models in order to account for the systemic emergence that the reductionist investigation neglects, or to explicitly "unwrap" the mechanisms that occur at lower biological scales.

Such composition is also a model, but for clarity we refer to a composite model as *hypermodel*, and to its component models as *hypomodels*.

Thus we define a *hypermodel*[7] as the composition and orchestration of multiple *hypomodels*:

- Component hypomodels capture the existing knowledge about a portion of the process, typically at a characteristic space-time scale;

- Relation hypomodels define how certain properties predicted by one hypomodel transform within the set of idealisations used to build another hypomodel that takes such properties as input.

It should be noted that a hypermodel could be re-used as a hypomodel in another more complex hypermodel. This poses some potential issues from a terminology point of view.

4.1.4 Computer models, metamodels

We define *Computer models* as computational entities that are "actionable" (can be invoked and executed): a computer program that implements a scientific model, so that when executed according to a given set of control instructions (control inputs) computes certain quantities (data outputs) on the basis of a set of initial quantities (data inputs), and asset of execution logs (control outputs).

4.2 An overview of integrative modeling

4.2.1 Motivations

Integrative modeling is necessary when the complexity of the phenomenon imposes a rigid reductionist approach, but the systemic element cannot be missed. We acknowledge three types of integrations:

a) Across time-space scales: because observational methods have mostly a finite resolution, the space-time granularity imposes a volume of interest. As a result we tend to develop

---

[7] The term hypermodel is used in other context with other meanings. For example:

- In logistics, Hypermodelling indicated a combination of architectural models and operational functions;

- In educational technology, hypermodel has probably been coined by Robert Tinker and refers to a sort of pedagogically structured microworld or computer-based manipulative (CBM) and a model-based learning design. The "hypermodel," a new type of learning technology that blends aspects of models, simulations, and hypermedia.

completely distinct mechanistic theories for the same biological process if observed at the organism, organ, tissue, cell, or molecular scales.

b) Across organ systems: traditionally the human body is separated using the semantics of descriptive anatomy, and then each organ system is investigated separately. This is also reflected in the clinical specialties, so pathophysiology knowledge is almost always separated between musculoskeletal, cardiovascular, etc.

c) Across knowledge domains: life is such an interesting part of nature that almost every foundational scientific discipline has developed its own perspective on it, consistent with its epistemology and its methods. So the same biological process can be radically different in the description of a biologist, of a biophysicist, of a bioengineer, etc. But in many cases each of these descriptions contains essential elements that need to be combined.

A forth motivation for integrative modeling is when we need to make explicit the mechanistic description of the process at a different scale not because this is necessary to improve the predictive accuracy of our mechanistic model, but because it makes explicit a control variable that can be observed / described only at that scale.

4.2.2 Multiscale physics-based modeling

In its most general exception a physics-based predictive model describes how a certain distinguishable portion of reality, described by a number of state quantities change over space-time given an initial set of value for such state quantities and a set of given conditions the rest of reality impose on the portion of interest (boundary conditions).

4.3   An example of a cancer hypermodelling

In this section we provide a brief outline of a paradigm of a cancer hypermodelling focusing on treatment response which is outlined in [2].

The anatomic region of interest is discretized by a virtual cubic mesh of which each elementary cube is termed *geometrical cell*. A *hypermatrix* - i.e. a mathematical matrix of (matrices of (matrices…of (matrices or vectors or scalars))) - corresponding to the anatomic region of interest is subsequently defined. The latter describes explicitly or implicitly the local biological, physical and chemical dynamics of the region. The following (sets of) parameters are used in order to identify a cluster of biological cells belonging to a given equivalence class within a geometrical cell of the mesh at a given time point:

I. the spatial coordinates of the discrete points of the discretization mesh with spatial indices *i, j, k* respectively. It is noted that each discrete spatial point lies at the centre of a geometrical cell of the discretization mesh.

II. the temporal coordinate of the discrete time point with temporal index *l*.

III. the mitotic potential category (i.e. stem or progenitor or terminally differentiated) of the biological cells with mitotic potential category index *m.*

IV. the cell phase (within or out of the cell cycle) of the biological cells with cell phase index *n*. The following phases are considered: $\{G_1, S, G_2, M, G_0, A, N, D\}$,   where G_1 denotes the G_1 cell cycle phase; S denotes the DNA synthesis phase; G_2 denotes the G_2 cell cycle phase; M

denotes mitosis; G_0 denotes the quiescent (dormant) G_0 phase; A denotes the apoptotic phase; N denotes the necrotic phase and D denotes the remnants of dead cells.

For the biological cells belonging to a given mitotic potential category AND residing in a given cell phase AND being accommodated within the geometrical cell whose center lies at a given spatial point AND being considered at a given time point - in other words for the biological cells clustered in the same equivalence class denoted by the index combination *ijklmn* - the following state parameters are provided:

> **i.** local oxygen and nutrient provision level (through angiogenesis and neovascularization),
>
> **ii.** number of biological cells,
>
> **iii.** average time spent by the biological cells in the given phase,
>
> **iv.** number of biological cells hit by treatment,
>
> **v.** number of biological cells not hit by treatment.

The initial constitution of the tumor i.e. its biological, physical and chemical state has to be estimated based on the available medical data through the application of pertinent algorithms constituting particular hypomodels. This state corresponds to the instant just before the start of the treatment course to be simulated. The entire simulation can be viewed as the periodic and sequential application of a number of sets of algorithms (*operators or hypomodels or component models*) on the hypermatrix of the anatomic region of interest. The application of the operators-hypomodels on the hypermatrix of the anatomic region of interest takes place in the following order:

> **A.** Time updating i.e. increasing time by a time unit (e.g. 1h),
>
> **B.** Estimation of the local oxygen and nutrient provision level (through angiogenesis and neovascularization).
>
> **C.** Estimation of the effect of treatment (therapy) referring mainly to cell hitting by treatment, cell killing and cell survival. Available molecular and/or histological information is integrated primarily at this point.
>
> **D.** Application of cell cycling, possibly perturbed by treatment. Transition between mitotic potential cell categories such as transition of the offspring of a terminally divided progenitor cell into the terminally differentiated cell category is also tackled by this algorithm set.
>
> **E.** Differential expansion or shrinkage or more generally geometry and mechanics handling.
>
> **F.** Updating the local oxygen and nutrient provision level (through angiogenesis and neovascularization) following application of the rest of algorithm sets at each time step. It is noted that stochastic perturbations about the mean values of several model parameters are considered (hybridization with the Monte Carlo technique).

Figure 6 provides a visual rendering of part of the approach described above. It has been taken from [1] where more details are available.

GEOMETRICAL CELLS

GC (x_i, y_{j+1}, z_k, t_n)	GC $(x_{i+1}, y_{j+1}, z_k, t_n)$
GC (x_i, y_j, z_k, t_n)	GC (x_{i+1}, y_j, z_k, t_n)

PHASES

G_1
S
$G_2 = p_l$
M
G_0
A
N
D

INDICATIVE HYPERMATRIX ELEMENT CONTENT

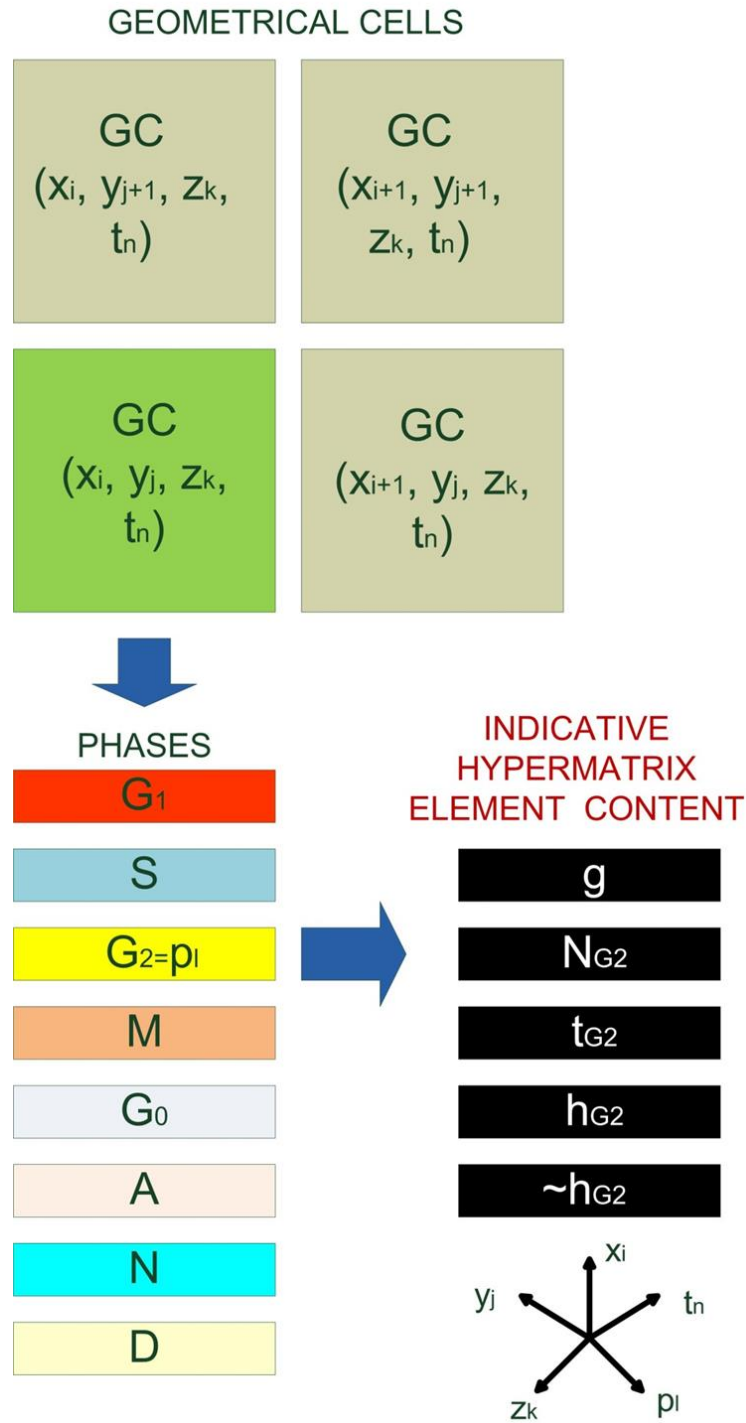| g |
|---|
| N_{G2} |
| t_{G2} |
| h_{G2} |
| $\sim h_{G2}$ |

x_i
y_j
t_n
z_k
p_l

Figure 6: A diagram showing the location of an indicative hypermatrix element $\overline{a}$ (x_i, y_j, z_k, p_l, t_n) **and its physically inhomogeneous and multidimensional content** $\left(g^{ijkln}, N_p^{ijkln}, t_p^{ijkln} \ h_p^{ijkln}, \tilde{h}_p^{ijkln} \right)$.

SYMBOL CONVENTION

Three dimensions (those corresponding to the variables x_i, y_j, z_k) represent space, another one (corresponding to the variable t_n) represents time and the fifth one (corresponding to the variable p_l) represents the cell phase within or out of the cell cycle in which a biological cell or a set of cells within a geometrical cell of the discretization mesh is found at a given instant.

p : phase within or out of the cell cycle

g : oxygen and nutrient provision

N_p : number of biological cells in phase p

t_p : mean time spent in phase p (time is usually measured in hours)

h_p : number of therapy hit cells residing in phase p

$\tilde{h}_p$: number of non-therapy hit cells residing in phase p

$x_i \in [x_{\min}, x_{\max}]$

$y_j \in [y_{\min}, y_{\max}]$

$z_k \in [z_{\min}, z_{\max}]$

$t_n \in [0, t_{\max}]$

$p_l \in [G_1, S, G_2, M, G_0, A, N, D]$

where

$\xi_{\min}, \xi_{\max}$ denote the minimum and maximum value respectively of the generic variable ξ during the simulation

G_1 denotes the G_1 cell cycle phase,

S denotes the DNA synthesis phase,

G_2 denotes the G_2 cell cycle phase,

M denotes mitosis,

G_0 denotes the dormant G_0 phase

A denotes the apoptotic phase

N denotes the necrotic phase

D denotes the remnants of dead cells

$g \in \{s, \tilde{s}\}$

s stands for sufficient oxygen and nutrient provision (for tumor cell proliferation)

$\tilde{s}$ stands for insufficient oxygen and nutrient provision (for tumor cell proliferation)

Obviously this binary character of the oxygen and nutrient provision is to be considered only a first simplifying approximation.

$N_p \in N_0$

N_0 is the set of non-negative integers

$t_p \in [0, t_{p\max}]$

$h_p \in [0, N_p]$

$\tilde{h}_p \in [0, N_p]$

$\sim h_{G2} \equiv \tilde{h}_{G2}.$

5 Software integration perspective

5.1 Service composition

A well-known paradigm in computer science is the encapsulation of information and functionality into distinct programming entities which interact with the rest of the computing environment via well specified end-points which are collectively known as an Application Programming Interface (API). These programming entities are given various names depending on the context or the usage, such as modules, components, libraries, classes, services etc.

This modularity allows for the abstraction and logical separation of functionality, the interchangeable usage of different components offering the same functionality with different implementation, the scalability of the implementation, the security of information through encapsulation, the ability for more elaborate error checking and many more advantages over a monolithic implementation which contains all functionality and information into one piece. These modules are often provided by third party implementers and can be used "off the shelf" as far as their programming interface matches the one that is needed. This is also the reason why it is often more critical to define in great detail the needed API of an application than to define the contained functionality, in order to automate as much as possible the interconnection of the modules, and equally important is also for applications to comply with standardized descriptions and protocols, so that this interconnectivity is easily achieved.

In the last years, this paradigm has been extensively used also in applications and systems that are geographically or logically distributed into many interconnected sub-systems via loosely coupled services provided on-demand. This type of modular and interconnecting architecture is known as Service Oriented Architecture (SOA). SOA techniques often apply the same also to applications which are only logically distributed, i.e. the services are running locally and communicate via inter-process communication (IPC) or other message passing techniques.

5.1.1 Web Services

A SOA can be implemented using various different technologies, such as RMI[8], CORBA[9], REST [80], Web Services[10] etc. Out of these technologies, the open technologies which rely on open standards, such as REST and Web Services, are the most promising and, from these two, Web Services is a W3C standard and thus it has received more attention over the last years and it has been standardized in a number of domains such as the description language, security mechanisms, binding and invocation mechanisms etc.

Web Services are described in a standardized format, WSDL[11] (Web Services Description Language) and the communication between different services is performed by using SOAP[12] messages, which is an XML-serialized message over HTTP protocol.

5.1.1.1 WSDL

The WSDL description of a service is composed of the following elements:

---

[8] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138781.html

[9] http://www.corba.org/

[10] http://www.w3.org/standards/webofservices/

[11] http://www.w3.org/TR/wsdl

[12] http://www.w3.org/TR/soap/

- **Types**, which describe the kinds of messages that the service sends and receives and can be simple types such as *double* or *string* or complex types composed from simple types.

- **Interface**, which describes the abstract functionality that the service provides.

- **Binding,** which describes how to access the service.

- **Service,** which describes where (URL) to access the service.

Although WSDL has conquered the world of Web services due to being a standard language to describe a service, its major drawback is that it focuses on the syntactic description and not on the semantics behind its types or interfaces.

5.1.1.2 SAWDSL

To overcome the problem of semantically describing a service, the Semantic Annotations for WSDL (SAWSDL[13]) was developed, defining a way to semantically annotate WSDL, linking its concepts to an ontology and thus helping in the service discovery, invocation and composition with other services. SAWSDL keeps the semantic model outside of the WSDL, making the approach independent from any ontology language. This is also a problem linked with SAWSDL, that being independent from the ontology language it is very difficult to define requests or matches between different services. As a result it is difficult with SAWSDL to support automated service discovery and composition.

5.1.1.3 OWL-S

OWL-S[14], a Semantic Markup for Web Services, is an ontology built on top of the Web Ontology Language (OWL) with the aim to alleviate the problems described above and to help users and programs to automatically discover, invoke and compose Web services.

The OWL-S ontology has three main parts:

- The **service profile**, which describes what the service does (for human reading).

- The **service model**, which describes how a client can interact with the service, its inputs, outputs, results etc.

- The **service grounding**, which specifies the details on how to interact with the service such as protocols, message formats etc.

The OWL-S atomic processes, inputs and output types correspond to WSDL operations and types respectively, so OWL-S is used in conjunction with WSDL. If only one of the two languages alone is used, it cannot fully describe a service both semantically and syntactically.

An extension of OWL-S is OWL-Q, which was developed to provide also a semantically rich model and description for the quality of service (QoS) aspects, such as metrics, constraints etc.

5.1.1.4 WSMO

The Web Service Modeling Ontology[15] (WSMO) is a conceptual model for describing various aspects related to Semantic Web Services. It provides a framework supporting the deployment and

---

[13] http://www.w3.org/TR/sawsdl/

[14] http://www.w3.org/Submission/OWL-S/

[15] http://www.wsmo.org/

interoperability of Semantic Web Services, with the objective to solve integration problems by defining a coherent technology.

The WSMO has the following main components:

- **Goals**, which describe a client's (either a human or computer agent) objectives when using a Web Service.

- **Ontologies**, which provide a formal semantic description of the information used by the other components, with domain specific terminologies.

- **Mediators**, which are connectors between components with mediation capabilities and provide interoperability between different ontologies, linking heterogeneous components when semantic incompatibilities exist.

- **Web Services**, which are semantic descriptions of the actual web services.

5.1.1.5 SWSF

Semantic Web Services Framework[16] (SWSF) is another attempt to realize the Semantic Web and has been influenced both by OWL-S and WSMO, described above. The SWSF is comprised of two major components, the Semantic Web Services Ontology (SWSO) and the Semantic Web Services Language (SWSL).

SWSL is used to specify formal characterizations of the Web service concepts and descriptions and it includes two sub-languages. The first language is SWSL-Rules for logic programming and reasoning and is used to support reasoning during the actual execution of services. The second language is SWSL-FOL (first order logic) and is used to express the formal characterization of concepts.

SWSO presents a conceptual model by which Web services can be described, similarly to OWL-S. And also similarly to OWL-S it is divided in Service description, Service Model and Service Grounding.

5.1.1.6 Comparison

An in depth analysis and comparison of these frameworks can be found in [73]. From the languages and frameworks described above, OWL-S is the most mature and commonly used, however, a problem of OWL-S is that it is mainly used to describe a service (a service model) but not to define how the services can collaborate, similarly to the notions of orchestration and choreography found in Workflow applications and frameworks. Consequently, the OWL-S approach faces problems when we try to apply it into automated or semi-automated web service composition.

5.1.2 Automated Web service Composition

A great number of surveys have been conducted [73], attempting to introduce solutions to the problem of automated service composition. A brief list of the requirements set by these attempts is:

- **Automation**: The automatic, as much as possible, generation of the service composition.

- **Dynamicity**: The characteristic whether the composition is static and cannot change after it has been built or whether it is dynamically created and can change, even after its execution has started.

---

[16] http://www.w3.org/Submission/SWSF/

- **Semantic capabilities**: For efficiency and quality reasons, semantically rich annotations of the services and their compositions are needed as much as possible.

- **QoS awareness**: Approaches which are QoS-aware take into account not only functional descriptions of the services but also other factors such response time, availability, etc.

- **Nondeterminism**: In many cases, an action may lead to different states, similarly to if-then-else control flow or loops in the execution. Consequently, the composition of the services may be depended not only on the underlying functionality but also on execution parameters and values and this must be taken into account.

- **Partial observability**: This requirement is linked to Artificial Intelligence (AI) based approaches, where only an incomplete view of the initial state or information may be available, thus a service composition may be attempted to be built based on partial, incomplete or even false information.

- **Scalability**: A requirement needed for real-world applications is scalability, since in real applications there may be performance restrictions or limitations.

- **Correctness**: Correctness is established through verification techniques and is applied in order to guarantee that a certain output will be produced under a certain set of inputs and conditions.

- **Domain independence**: This characteristic applies when we want a composition approach which will not be limited to a certain domain, but we want the same techniques applied to different domains and solving different types of problems. Usually this is connected with the semantic capabilities of our approach, in order to use domain-specific knowledge on each case.

- **Adaptivity**: The adaptivity requirement is linked with the ability of an approach to adapt when requirements change, and is viewed as going a step further from dynamicity requirement.

This list of requirements shows the wide range of both research challenges and technical difficulties invoked when attempting to cope with the automatic or semi-automatic service composition problem. Each composition model, strategy or technology employed on this challenge, usually manages to treat some of these requirements but it is very difficult to find one approach that implements most or all of them.

5.1.2.1 Composition models

There are various models for the realization of the automated composition. Most approaches use one (or a combination) of the following composition models:

- **Orchestration**, which is a description of how the services that participate in a composition interact, including the business logic and the order of the execution. It is different from the choreography in that it relies more on a "global" or centralized view of the whole composition. Service orchestrations are usually described and executed using a workflow language, with the most prominent being WS-BPEL[17] (Business Process Execution Language for Web Services). Initially WS-BPEL was only associated with WSDL descriptions; however there are attempts to provide support of semantics.

- **Choreography**, is a process where the participating parties (services) are in full control of their internal business logic, and are conceptually related with message exchanges that

---

[17] http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

follow the rules of an overall choreography. The most prominent language for defining choreographies is WS-CDL[18] (Web Services Choreography Description Language).

- **Service Coordination**, which groups services by following a coordination protocol. Usually the participating services communicate through a coordinator, which applies the coordination model. An existing coordination framework is WS-CF[19] (Web Services Coordination Framework) that can define a coordination based on three components: the coordinator, the participants and the coordination service.

  Component Model, is a model also referred to as service wiring and involves the actual linking of inputs and outputs of the composed services. In this architecture, independent of their implementation or programming language, the service components can be encapsulated so that they share similar descriptions and then be put together to form a composite service. An example of this model is Service Component Architecture[20] (SCA).

5.1.2.2 Automated Web Service Composition Approaches

Different approaches have been developed to tackle the problem of the automated (web) service composition that can be vastly categorized to the following groups:

- Workflow-based
- Model-based
- Mathematics-based
- AI planning approaches

5.1.2.2.1 Workflow-based approaches

A service composition is very similar to a workflow, so it comes natural that knowledge from workflow research is applied to this field. The most prominent and mature technology on workflow based service composition is BPEL (Business Process Execution Language).

Initially this work targeted static or manual service compositions, but recent attempts try also to automate the process of the composition. These attempts have resulted in frameworks which focus on the automated construction of a workflow, based on a high level goal expressed in BPEL which is then matched with low level services that offer the needed functionality, or BPEL descriptions of workflows that can dynamically select at runtime the services to be executed out of a variety of services with the same or similar interface. The high level goal can sometimes be expressed in natural language, and then through natural language processing techniques a workflow synthesis is made which tries to match the described functionality, taking into account if the existing services semantically match with the functionality of the workflow. We further analyse the workflow-based service composition in paragraph 5.2.

5.1.2.2.2 Model-based approaches

Model-based or model-driven composition approaches use already existing models to represent Web Services. They use a higher level description on top of the existing description in WSDL, OWL-S etc. with a combination of Finite State Machines (FSMs) [81][82] or UML activity diagrams [83][84] and

---

then try to automatically synthesize a BPEL description of a workflow. However, these techniques are questioned whether they produce a deterministic workflow result. Also, they greatly lack in dynamicity as the composite service is created manually. Petri Nets [85] - [87] and Finite State Automata [88] have been also used in other research efforts, including attempts for automated service composition incorporating the use of mathematics, such as algebraic languages, pi-calculus and Linear Logic [89] - [92].

5.1.2.2.3 AI Planning approaches

Another category of automated Web service composition approaches includes all research efforts that use Artificial Intelligence (AI) planning techniques in order to generate a composition schema. AI planning techniques involve generating a plan containing the series of actions required to reach the goal state set by the service requester, beginning from an initial state. All approaches in this family rely on one of the many planning techniques that the AI community has proposed and incorporates it in the process model creation phase of the composition framework. Service composition approaches of this category include Classical and Neoclassical Planning techniques, Heuristics, Control Strategies and other [93] - [108]. An elaborate analysis and comparison of these techniques can be found at [73].

5.2 Workflows

The Workflow Management Coalition[21] (WFMC) defines a workflow as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". In other words a workflow consists of all the steps and the orchestration of a set of activities that should be executed in order to deliver an output or achieve a larger and sophisticated goal. In essence a workflow can be abstracted as a composite service, i.e. a service that is composed by other services that are orchestrated in order to perform some higher level functionality. The compositing services (steps/tasks) can have a variety of complexity and usually are connected in a non-linear way, formulating a directed acyclic graph (DAG).

A Workflow Management System defines, manages and executes workflows through the execution of software that is driven by a computer representation of the workflow logic. The description of a workflow includes the definition of different tasks, their interconnection structure and their dependencies and relative order. This description of the workflows operational aspects can be expressed in textual (e.g. XML) or graphical form (e.g. as a graph in Business Process Modeling Notation [79] or Petri nets).

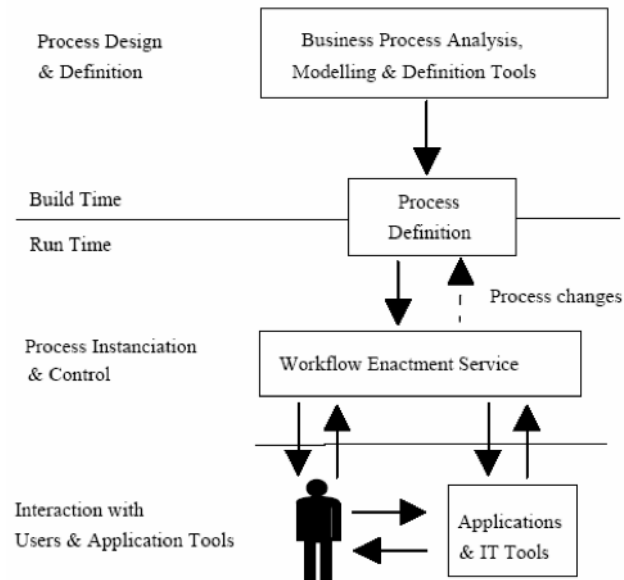The Workflow Reference Model (WFRM) proposes the following model:

---

[21] http://www.wfmc.org/

Figure 7: The Workflow Reference Model from the WfMC

This model defines two major phases for workflows:

- The build phase where the workflow is defined in terms of a textual or graphical language. Various modeling languages that have been proposed include the Web Services Flow Language (**WSFL**), Microsoft's **XLANG** for BizTalk, and Business Process Execution Language (**BPEL**), which is the cooperative merging of WSFL and XLANG for Web services orchestration.

- The run phase where the workflow is enacted according to its definition by a workflow execution (enactment) component or a *workflow engine*. At this phase the execution of some tasks may require the interaction with users or other software applications and tools.

More specifically we can identify at least four important aspects of a workflow building and enacting process:

- User environments, usually graphical, where the user can define a workflow

- Representation languages that are used to express workflows

- Translation or compilation of a workflow so that it could be enacted

- Execution of a workflow and runtime support.

There are cases where all these actors and stages in workflow design and enactment are supported by a (seemingly) single Integrated Development Environment (IDE) that hides the underlying complexity from the user.

In addition to the business oriented use cases, workflows have a lot of potential in scientific areas as well. At the current pace of information production there is an unprecedented demand for extraction and processing of knowledge. This is more than evident in various scientific fields such as molecular biology, high-energy physics, and astronomy. Consequently, scientific workflows have been proposed as a mechanism for coordinating processes, tools, and people for scientific problem solving purposes [78]. They aim to support "coarse-granularity, long-lived, complex, heterogeneous, scientific computations".

In this case however there are some special requirements that differentiate the scientific workflows from the business workflows: they should support large data sets and data flows with a large number of parameterized jobs and the execution is usually done in dynamic environment where resources are not know a priori. Scientists are usually having a hard time trying to locate the data they want, gather them, and find the necessary tools in order to process and analyse them. There are many software tools available to support their scientific experiments but they usually work differently and require a learning phase that's an impediment to their rapid deployment. Also the different data formats (even if we consider XML formatted documents only) impose either the time consuming task of manual data transformation, or the custom development of wrappers and converters (probably in some scripting programming language, e.g. Perl), which is definitely something beyond a scientist's area of interest and expertise. In the case of an experiment or study there are also additional issues that relate to the reproducibility of the scenario, the validation and the recording of the provenance of the data inputs. Therefore the composition of the available tools in terms of a scientific workflow in order to orchestrate them for performing some scientific scenario or experiment presents more challenges than in the case of business workflows.

6 Interoperability with other E.U. projects

6.1 VPHOP

The VPHOP[22] was an Integrated Project funded by the EC (FP7-Collaborative Project GA no 223865, 2008-2012) that developed ICT infrastructure and tools to predict the risk of fracture in osteoporotic patients. The prediction was based on the integration of the outputs produced by algorithms and models at different spatial scales (from the body level to the cellular one). In terms of composition and orchestration of the execution of the different models and relevant for the CHIC project, a hypermodelling technology have been developed during the project, called VPH-HF.

The project ended in 2012 but it is highly relevant for CHIC as some the developed technologies might be used as base components for the CHIC hypermodelling framework.

6.1.1 VPH-HF architecture

The VPH-HF technology design relies on the concept of the "wrapper", which is used to wrap the sub-model codes and also for exposing the other services of the infrastructure. This guarantees sufficient generalisation in case it is needed to substitute or add other services/modules

In Figure 8 the overall architecture of the VPH-HF is represented. The green boxes represent the components devoted to the communication and the blue ones the core services. The orange boxes represent end-users applications, which can be used for operations requiring user interaction or to configure, launch and monitor the VPH-HF execution.
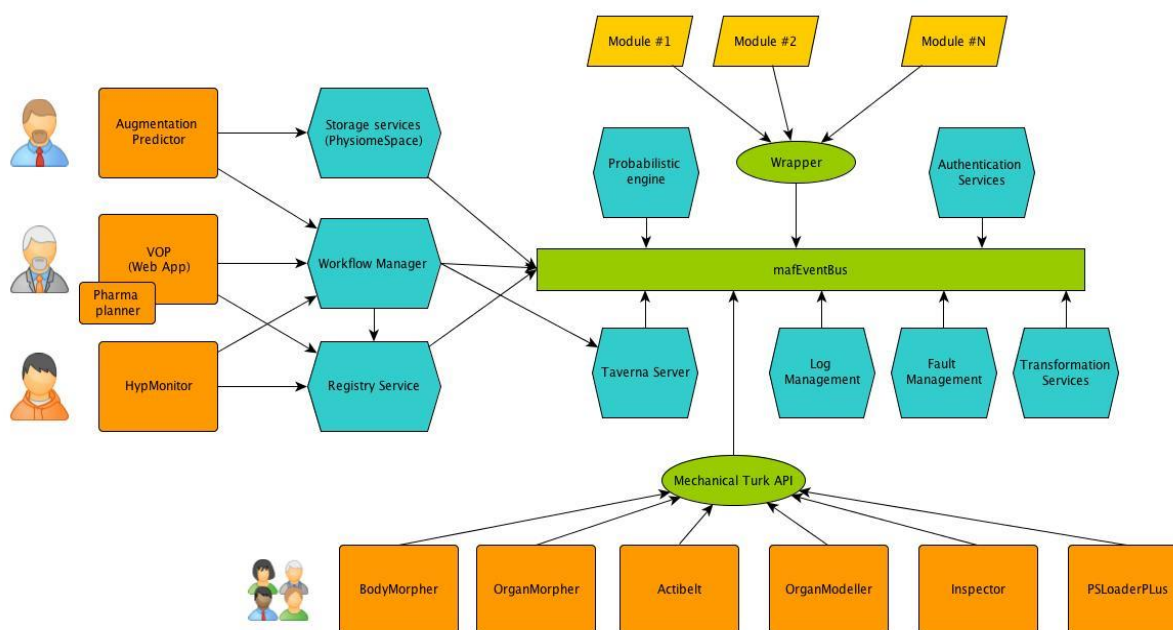

Figure 8: VPH-HF hypermodelling ICT architecture

The core components of the VPH-HF were implemented replying on the Multimod Application Framework (MAF). MAF[23] is an open source multiplatform framework (BSD-like licence) written in

---

[22] http://www.vph-op.eu/

[23] http://www.openmaf.org/

C++ for the rapid development of computer-aided applications. Its 3rd version (called MAF3) aims to improve concepts and design from the previous version, and to ensure the possibility of extending functionalities in a simple way, through the use of plug-in library.

MAF3 is based principally on Qt[24], and it has a set of pre-compiled libraries that represents the "Foundation Libraries" including last stable versions of ITK[25], and VTK[26].

6.1.2 VPH-HF components

6.1.2.1 Storage services

The storage services take care of the input/output data (including intermediate results coming from the sub-models). They have been developed relying on and extending the services of the data sharing application, PhysiomeSpace[27]. The storage services are also used by the Mechanical Turks (see later section for more details) to provide the data that need manual processing to the running workflow. In particular,

- Each module is able to communicate with the central data repository, via the Storage Services component, by pushing or pulling files if necessary, and sending commands through the APIs provided by PhysiomeSpace services.

- The high level design provides an extremely simple and flexible interface based on events containing a dictionary with commands and parameters to execute.

- Generally a module will check if all the input data are locally present for the execution and eventually retrieve from the repository the missing ones via PhysiomeSpace web-services; then, the module will run and produce results in terms of data file stored locally.

- After generating the output, the last operation is to upload the result files following the selected workflow for generating a new resource (execute create command and push the file into repository).

- The uniqueness of the data and logs between executing workflows has been achieved by the smart use of workflow id (unique for each workflow), which groups the resources in independent directories.

6.1.2.2 Communication services

The communication service allows the communication of all hypermodel modules and provides the message exchange between the different services. It has been implemented as part of MAF3 (*mafEventBus*) and it allows MAF3 objects to become signal emitters or observers in order to communicate with each other in a dynamic way using the signal/slot mechanism implemented inside the Qt framework (Figure 9).

---

[24] http://qt.project.org

[25] http://www.itk.org

[26] http://www.vtk.org
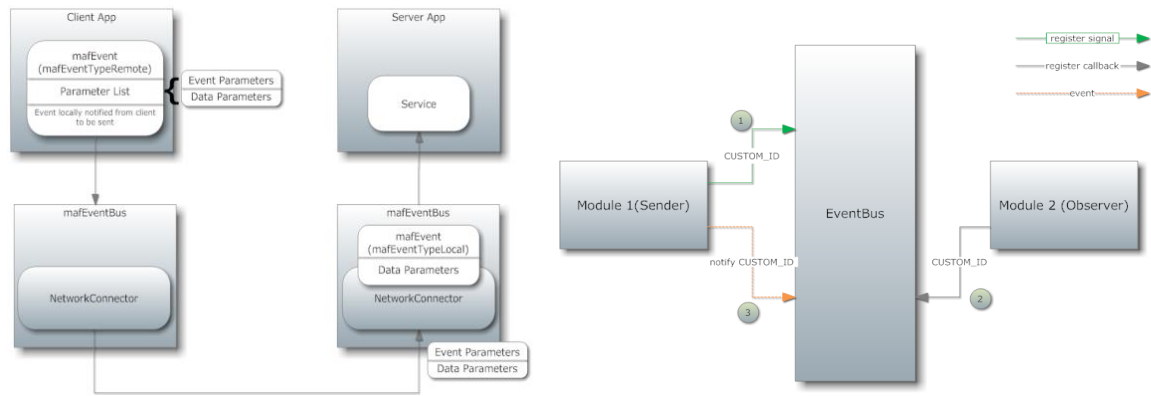
[27] http://www.physiomespace.com

Figure 9: mafEventBus architectural schemes

A façade class called *mafEventBusManager*, which hides all the complexity of event dispatching and managing, is responsible to make possible the registration of objects as event sender or observer.

To be able to send an event, the object has first to create a new ID, which is in the form of reverse DNS notation; in this way, the object can register its own signal to the event bus giving as parameters the ID, the sender and the signal signature. The signal is registered into a hash and then another object can become observer for that ID and be connected dynamically with that signal at runtime.

To register an object as observer it is necessary to call the specific macro and pass to it the observed ID, the pointer of the observer and the slot signature.

The event dispatching is realised following the Strategy pattern. There is a base abstract class that defines the interface through which the *mafEventBusManager* performs the event notification and then according to the dispatcher used (local or remote), the event is dispatched from the emitter to the observer.

Remote event dispatching is also implemented using another strategy pattern based on the remote protocol: in the current version the XMLRPC and SOAP protocol connectors have been developed.

6.1.2.3 Wrapper

During the design phase, it was taken into consideration that the wrapper should:

- be easy to be used by non-developers (i.e. researchers);
- allow the mapping from the general syntax of the hypermodel to the local one of the sub-model;
- be open to extension, which means new sub-models can be wrapped.

Thus, each sub-model has been wrapped by a *wrapper* which allows each service to be considered as a black-box without needing details on how it is implemented or how it works internally. In particular, in order to execute an algorithm/module inside the hypermodel environment, the code needs to be wrapped as a MAF3 operation, which can be then started from the *mafEventBus* (the communication component) call coming from remote requestor. The wrapper starts the algorithm execution and passes to it all the parameters coming from the *mafEventBus* call (see Figure 10).

The implementation of the wrapper is represented by the *mafAlgorithm* operation, which is registered into the MAF3 factory and allows starting a shell script with some parameters.
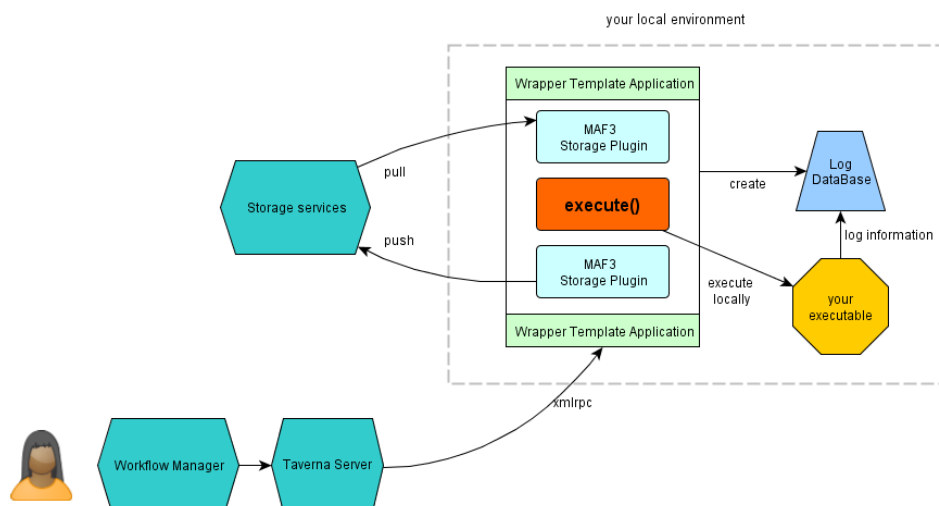
Figure 10: Wrapper architectural schemes

The Wrapper takes care of downloading all needed input resources from the remote storage server, creating a database to store the log information and eventually uploading output resources to the remote storage server. The wrapper then demands the execution to the external process.

The communication from the wrapper to the external process is realized by passing command line parameters. The communication from the Taverna Server (VPH-HF component for the workflows orchestration) to the wrapper is realized by a remote call handled by the *mafEventBus* in a transparent way.

6.1.2.4   Workflow manager

Together with the communication one, this service has a central role in the hypermodel as it takes care to launch the sub-models in the right order and to receive the information on when the sub-model stops or ends its execution. It is also the interface between the end-user application and the backend hypermodel technology.

In summary, the Workflow Manager is the workflow choreographer, and it:

- Handles association between users and workflows,

- Handles communication within the end user applications and other hypermodel core modules, and

- Permanently serializes workflows related information.

The Workflow Manager is implemented based on the Flask Python Microframework[28]. It aims at sharing the active session between all modules/services along the execution of a workflow: a Postgres SQL database is used to map a user to his/her running workflow and related session cookie; auto-completion of workflow XML file adds the Workflow ID and cookie (for authentication) for each step; it posts the workflow and input definition to the Taverna Server and launches the workflow execution; and it returns the workflow status and its outputs.

The service is completed with persistent information on users and workflows and their parameters, to properly expose the necessary information and output results to the end-users applications.

---

[28] http://flask.pocoo.org/

6.1.2.5 Taverna server

MAF3 hypermodel workflow is based on the client-server paradigm in which a Taverna Server represents the orchestrator of the hypermodel. Taverna[29] is an open source and domain-independent Workflow Management System – a suite of tools used to design and execute scientific workflows. The Taverna Workbench enables the graphically creation, editing and running of workflows locally. The Taverna Server is the remote workflow execution service that enables a dedicated server to be set up for executing workflows remotely.

In particular, Taverna Server more relevant characteristics in the VPH-HF are:

- XML-based Workflow Definition language,

- Support for calling service on local or remote machines, and

- Secure access to resources on the Web.

The Taverna server has been installed and configured for managing the execution of workflows launched from the end-users applications.

6.1.2.6 Registry service

The Registry service contains information on all the available sub-models (together with information on the type of data for input and output). It collects information on all modules available to compose a workflow. This information includes the description and status of the module (i.e. available, down, free, busy), the response time, location, permission level, etc.

As for the workflow manager, it is implemented based on Flask Python Microframework and it provides the possibility for the end–users applications to gather information with a remote call.

The registry service can include not only the sub-models but also all the basic components for the architecture so that they can be launched and/or monitored via the client applications.
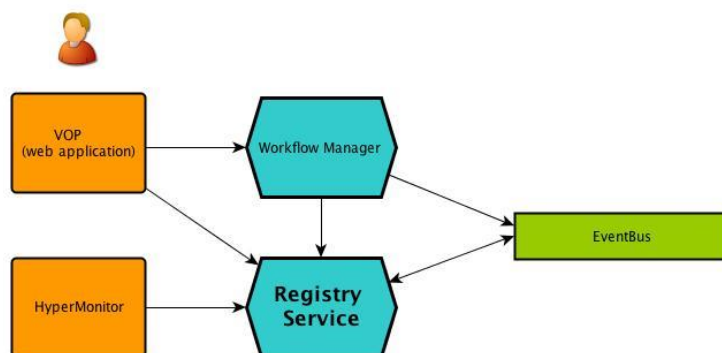

Figure 11: Registry service communication scheme

6.1.2.7 Authentication service

The authentication service provides the mechanisms to authenticate the user into the system with features for accounting and granting permissions only to certain parts of the VPH-HF. The service has been implemented using the Biomed Town[30] OpenID Identity provider service and the Apache

---

[29] http://www.taverna.org.uk/

[30] http://www.biomedtown.org

mod_auth_tkt module[31]. The mechanism is based on a session cookie with an expiry time, which is passed to all services.
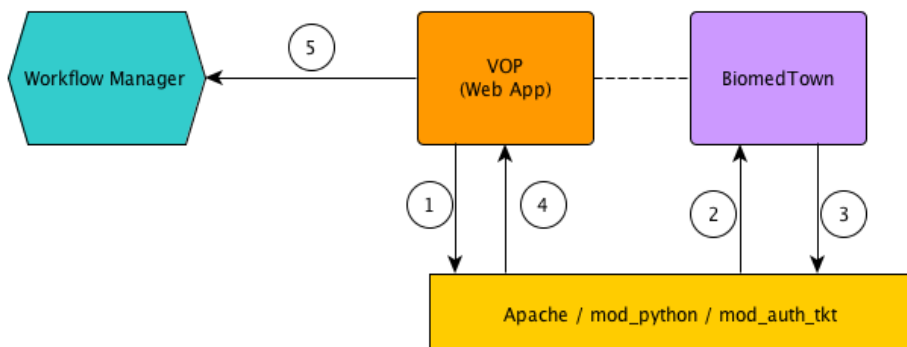

Figure 12: Authentication service scheme

6.1.2.8   Log management service

The log mechanism is an important aspect during the workflow execution. Each module/service composing the workflow needs to communicate information about the execution status and in some occasion may have a failure for different reasons, especially in a distributed system: network error, resource overloading, un-availability of services, etc.

Each hypermodel component logs information locally. Then, the Log Management service is the module which requests by periodic polling from each module a detailed log, and saves in a database record the result of the call. It has been implemented using the Flask framework, as for the workflow manager; it gathers information that can then be queried by other modules and provides APIs for accessing the database.
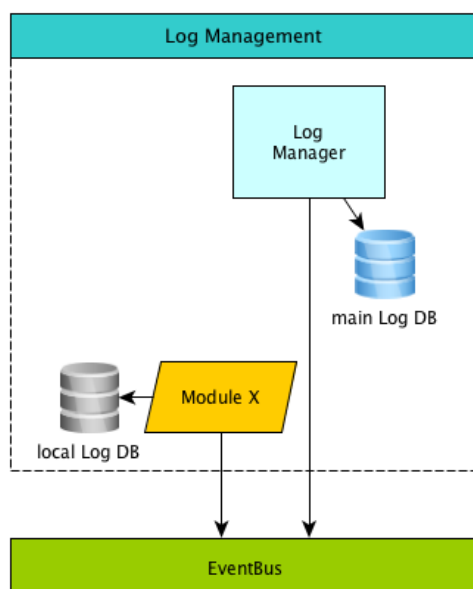

Figure 13: Log management architectural scheme

---

[31] http://search.cpan.org/~gavinc/Apache-AuthTkt-0.08/AuthTkt.pm

6.1.2.9 Mechanical Turk

The "*Mechanical Turk*" is a common API, which is used to integrate into the VPH-HF sub-modules that require human intervention or manual operations during the execution.

When this module is encountered during the workflow execution, the Mechanical Turk execution responsible person is notified with an email, which reports also the link to the input data on the storage services; the user processes the data and after uploading it back on to the storage services, he/she confirms the completion of the tasks; at this point, the workflow managers is notified and the execution of the following modules is launched.

6.1.2.10 Front-end applications

The VPH-HF was implemented with different end-user interfaces to allow users to transparently access the hypermodel architecture and run the prediction workflow to get the risk of fracture for a select patient:

- VOP, web application, which allows clinicians and researchers to access the patients' data, and ask for the personalised risk of fracture based on the available data and clinical workflows; furthermore, researchers can monitor the status of single services or add/modify new services and workflows;

- HyperMonitor, MAF3-based client application, which allows researchers to compose the different modules present in the hypermodel infrastructure into new workflows and execute them.

6.2 TUMOR

The TUMOR project[32] (FP7-Collaborative Project GA no 247754) developed a European clinically oriented semantic-layered cancer digital model repository from existing EU projects that is 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) which is NIH/NCI-caGRID compatible. The objective is to 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. To ensure the clinical relevance of this joint effort, the development of the project was based upon specific clinical scenarios that were implemented within an integrated EU-US workflow environment prototype for predictive, In Silico Oncology-guided clinical studies. As an end result, a specific, clinically relevant workflow involving both EU and CViT models was demonstrated, which highlighted the need for models and model repositories interoperability.

The project is highly relevant to the work plan of CHIC and although it finished on September 2013 it has a lot to offer both in term of experience but also in terms of infrastructure. In particular, in the context of TUMOR a semantics-enabled Model Repository has been designed and built, alongside with a web based workflow designer to facilitate the model linking and integration.

6.2.1 The TUMOR project Model Integration Strategy

The aim of TUMOR project was to develop a European clinically oriented semantic-layered cancer digital model repository from existing EU Virtual Physiological Human (VPH) related projects

---

[32] http://tumor-project.eu/

designed to be interoperable with the US grid enabled semantic-layered digital model repository platform at CViT[33] which is NIH/NCI-caGRID compatible. The ultimate goal was to build an integrated, interoperable transatlantic research environment offering the best available models and tools for clinically oriented cancer modeling and serving as an international validation/ clinical translation platform for predictive, in-silico oncology.

To achieve this ambitious goal, an interoperable, transatlantic environment is needed to offer a range of services to international cancer modellers, bio-researchers and eventually clinicians in fostering both basic cancer research and individualized optimization of cancer treatment.

The TUMOR integrated environment comprises a distributed software system and therefore it is essential to describe its architectural characteristics, i.e. its components, the interactions between these components, and the principles and non-functional characteristics of these interactions. A general overview of the system is shown in the figure below:
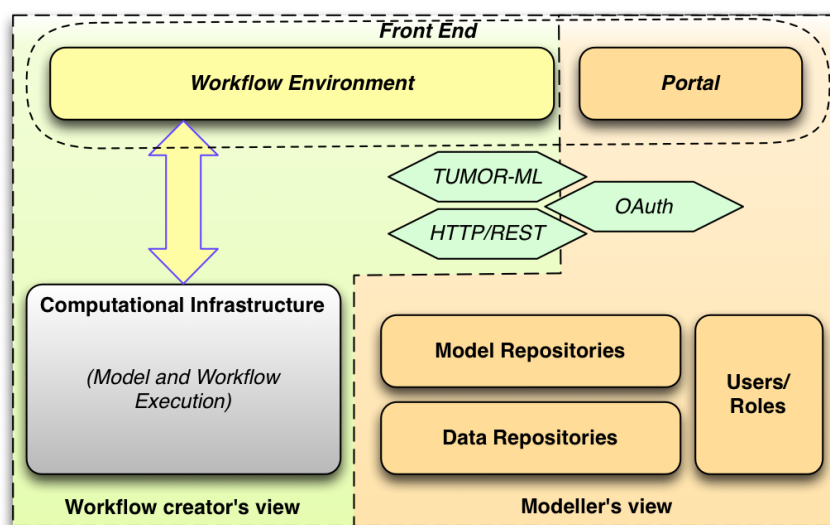

Figure 14: The main components of the TUMOR platform

We consider two main "facades" of the system:

- In the "Modellers' view" the main functionality of the system deals with the management of computational models and relevant data in the Model and Data repositories of the platform. In this façade users upload/register models and data, and the system through its Portal supports their discovery, navigation, and download. The primary entry point for this view of the system is the EU Model and data repository through its portal, which is also the "Common Access Point" for the whole platform.
- In the "Workflows creator view" the system supports the linking and execution of the linked models in a "software as a service" way. The actual execution happens in the TUMOR's servers backend without imposing any load in the users local machines. The entry point for this view of the platform is the TUMOR workflow environment.

The actual deployment of the system is shown in the next figure:

---

[33] Center for the Development of a Virtual Tumor, Massachusetts General Hospital (MGH), Boston, USA) http://www.cvit.org/ [73]
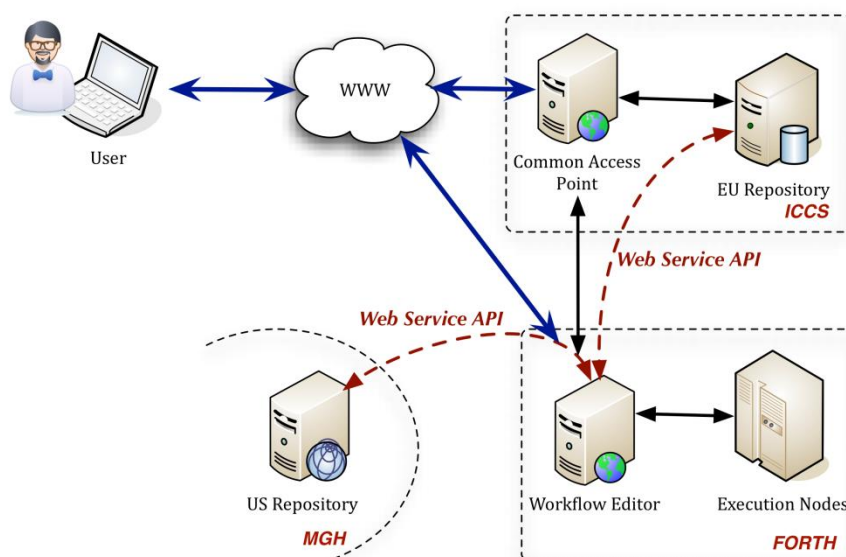
Figure 15: The Tumor platform deployment

The EU Repository and its Common Access Point (CAP), i.e. the Portal, are hosted in the ICCS premises. The Workflow Editor and Engine alongside with its supporting execution infrastructure is deployed in FORTH. The US Repository is hosted on the other side of the Atlantic, in the CViT.org infrastructure.

6.2.2 Model Description

To build the envisioned workflow environment existing standards should be selected wherever possible, while designing new ones to cover missing domains. The idea is to facilitate model linking with no extra effort to port existing models to a new framework, or re-implementing them, both costly and error prone activities. Hence, the need to fuse disparate models together, in the presented platform, is addressed using the Systems Biology Markup Language, SBML to model the biochemical processes at the molecular scales, whereas the higher and more clinically relevant scales, specific to cancer modeling, are addressed using the newly developed TumorML markup language.

6.2.2.1 SBML

Among the numerous standards related to model description at the sub-cellular level CellML[34] and SBML[35]are the most widely accepted ones. Both attempt to describe the structure and underlying mathematics of sub-cellular models. SBML is more specific and constrained in exchanging information about pathway and reaction models and uses successive hierarchical declarations of model constituents. There is also a wide community supporting SBML and tools to convert CellML to SBML. We prefer SBML mainly based on its constrained nature, which allows the language to be adopted quickly and evolve with the requirements of the representation and understanding of systems biology.

---

[34] http://www.cellml.org

[35] http://www.sbml.org/

6.2.2.2   TUMOR Markup Language (TumorML)

The higher scale models enrolled in the TUMOR environment are described using TumorML[76], a new markup language (ML) for describing cancer models. The development of TumorML contributes to enabling some of the key aims within the TUMOR project.

Firstly, by annotating cancer models with appropriate document metadata, digital curation is facilitated in order to make publishing, search, and retrieval of cancer models easier for researchers and clinicians using the TUMOR digital repository. Secondly, markup will be used to describe abstract interfaces to published implementations allowing execution frameworks to run simulations using published models. Finally, TumorML markup facilitates the composition of compound models, regardless of scale and source, enabling multiscale models to be developed in a modular fashion and models from all around the globe may be integrated with any related models in the TUMOR transatlantic platform. The TumorML model description also incorporates and integrates with the MIRIAM guidelines[75] in order to provide reference correspondence, attribution annotation and external resource semantic annotation to the described models.

An example of a TumorML description is shown in the next figure. As can be seen the format is XML based and it incorporates the following information:

- Descriptive metadata, like the title, description, author and publication information for the corresponding model. This metadata section also includes domain specific information, such as the type of cancer simulated, the types of mathematics used (e.g. discrete or continuous), the "biocomplexity direction" (e.g. bottom-up, top-down), etc.

- Model parameters information. This includes the parameter names, data types (e.g. double), default values, and semantic annotation.

- Execution information which contains links to executables, command line arguments, etc.
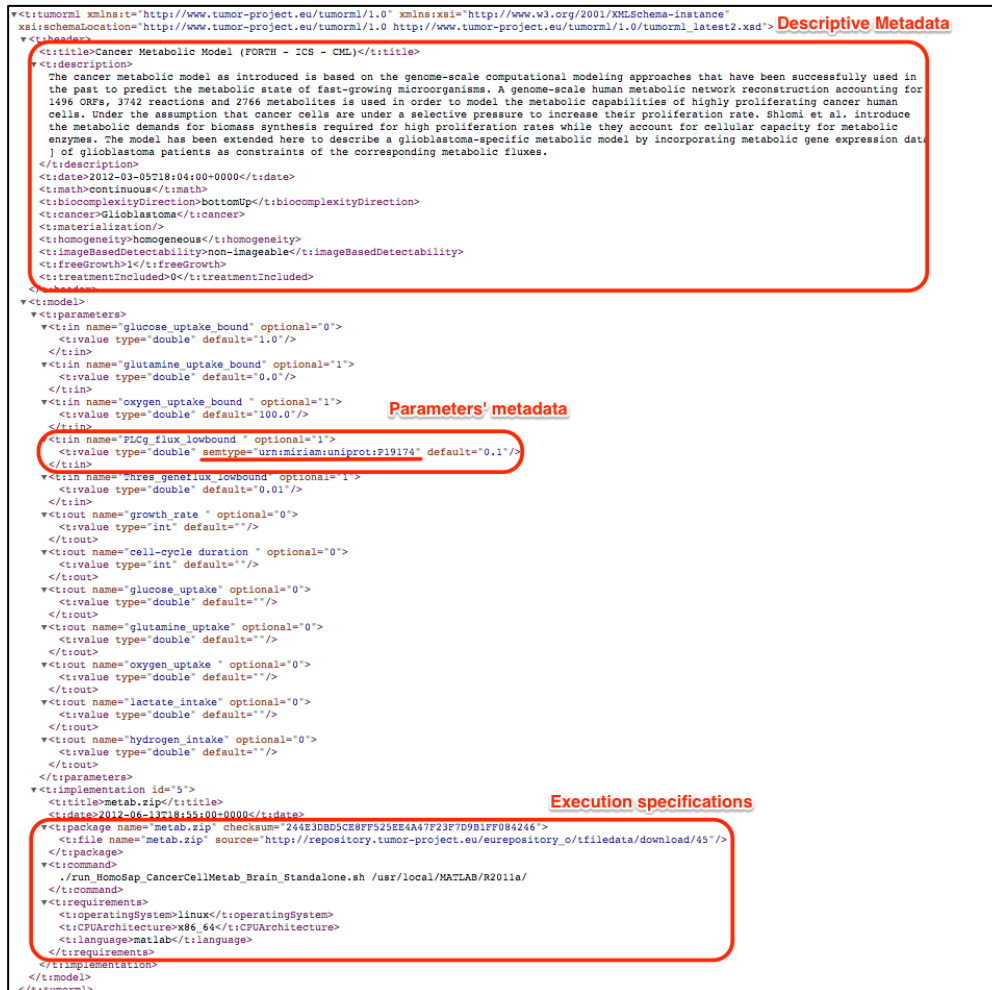
Figure 16: An example of the TumorML description of a model

6.2.3 Model Execution

There are two main execution frameworks in the TUMOR platform. The first is based on the SBML description of a model whereas the second one is more generic in the sense that a model can be provided as a self-contained executable. An SBML description of a model is a declarative artifact. It describes the mathematics required, typically in the form of Ordinary Differential Equations (ODEs), to implement the model and nothing else. In order to implement the model a solver is required to numerically resolve the equations, and execute the corresponding reactions based on the kinetic laws and the prescribed parameter values. This solver can be a simulation environment, a compiler that links the SBML file with numerical library and generates a standalone executable or a partial evaluator that attempts to unfold the ODEs with respect to known solving algorithms. In general the SBML models can be classified as deterministic or stochastic, with the latter using Monte Carlo simulation and related methods. The TUMOR execution infrastructure supports deterministic and stochastic models, through the incorporation of the COPASI simulator[36]. The use of COPASI software allows the parsing of SBML models and their execution but nevertheless there are a couple of parameters that need to specified prior to the execution:

- The simulation time for the model

---

[36] http://www.copasi.org/

- The algorithm to be used, e.g. deterministic, stochastic or hybrid.

- These parameters are not specified by SBML but they are essential in order for the models to produce the desired results. In order to support flexibility, the users can input values for both parameters at runtime. These parameter values are then passed to the COPASI solver for simulating the models.

- In the more generic case, the model is provided with no information on its internals. The supplied code, either in binary or in source format, should be able to be run as a command line program with its inputs and outputs specified either as command line options or as files. For example, if the execution framework (as in our case) is a Linux 64bit environment, the supplied executable code should be compliant with it. Of course, in the case where the source code of the model is available in the form of a scripting language, like Python or Perl, there are fewer restrictions imposed to the model creators.

Irrespective of the models' type (SBML or generic/command line formats), TumorML offers a generic metadata "envelope" to describe both their interface, i.e. input parameters and output results, and execution requirements. The interface definition provides valuable information for *linking* models in the workflow editor, based on the required input and the generated output. On the other hand the execution information is utilized from the workflow's runtime, when the models are simulated or executed.

6.2.4 Functionality of the Workflow Engine

The workflow engine is the server side of the environment and its main responsibilities are:

- The authentication of the users. This is subsequently delegated in the model repositories using the OAuth protocol, an open web based standard for authentication and authorization.

- The communication with the model repositories to retrieve the TumorML descriptions of the models and the corresponding executable programs and other data needed. This is implemented using web services.

- The storage and retrieval of user workflows. The persistence of the workflows is supported by a MongoDB[37] database server.

- The execution of the workflows. The TumorML descriptions retrieved from the model repositories provide detailed information about the inputs and the outputs of each model. Using this information the workflow engine is the "orchestrator" of the models executions, deciding what to run next, and how to pass the data from the one model to the next.

For the authentication and authorization aspects, there is the need for authenticating the users with the minimal possible distraction and also supporting authorization and access control. The workflow environment is a separate web application that stores neither user login information, nor the models and the accompanied data. So there is a need for *Single Sign On*, so that the users are not required to signup twice or to provide the same credentials twice when they access the EU Repository and the Workflow Environment. Furthermore, the users should be allowed to make secure and authenticated requests to the model repositories through the workflow environment, e.g. for retrieving the models' descriptions. To address both of these concerns, they are using the OAuth 2.0 (Open Authorization, version 2.0) protocol[38] that is also supported by Google, Microsoft, and Facebook in their web applications.

---

[37] http://www.mongodb.org

[38] http://oauth.net/2/
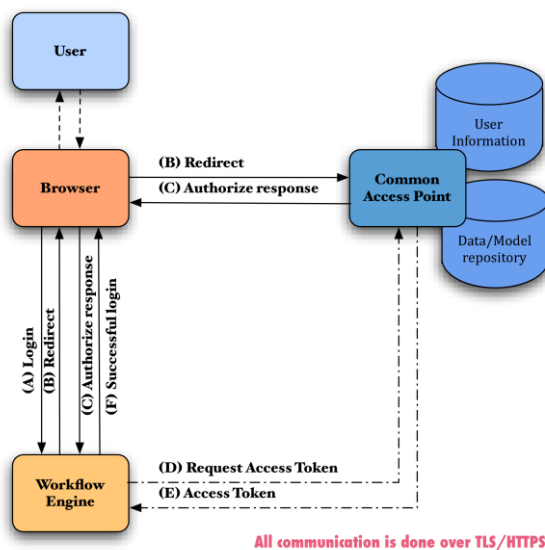
Figure 17: The flow of information and control in the OAuth based single sign on.

In "layman's terms", through the use of the OAuth protocol, the Workflow Environment can access the model repositories on the users' behalf without knowing their passwords or other authenticating information. Currently only the EU repository requires authenticated access based on the user information that it maintains. The US repository provides open access to its TUMOR compliant models and therefore there's no need for the users to go over a separate authentication process with the CViT based repository.

The flow of the information among the different components for logging into the Workflow environment is shown in Figure 17 with the various interactions (labeled as (A)-(F)) and it's as follows:

Figure 18: The main working area in the Workflow Editor for designing hypermodels.

- A user visits the Workflow Editor web site through his/her browser (A).
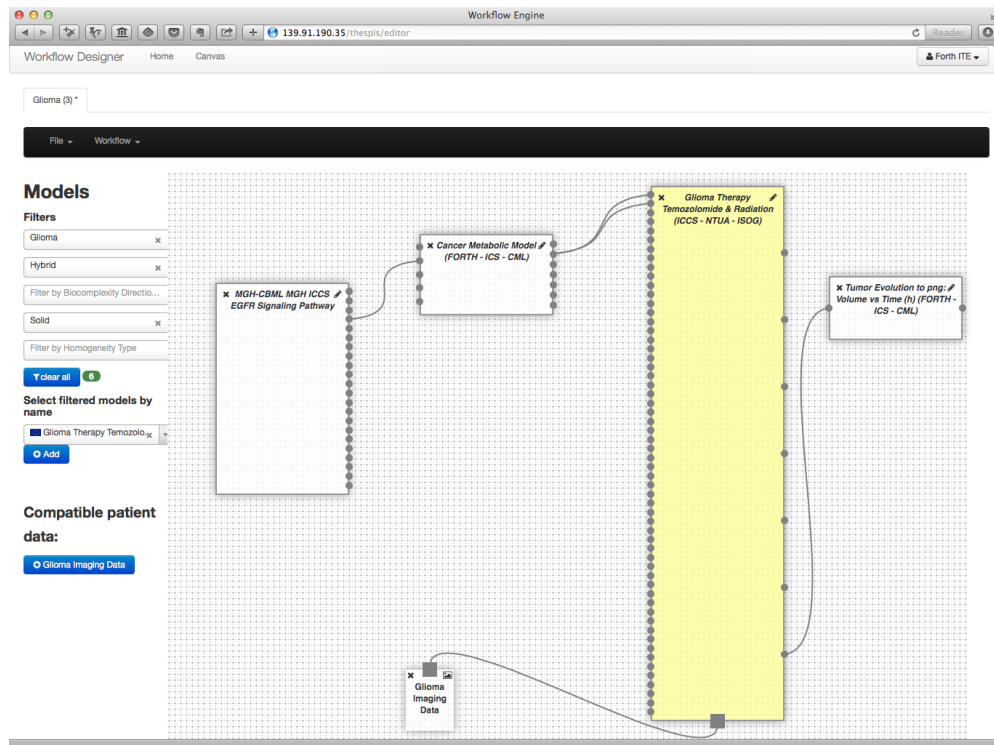
- The Workflow environment web site finds out that this is a user that hasn't been authenticated and therefore needs to be authenticated (logged-in). It subsequently presents a login page (Figure 18) and redirects (B) the user to the EU Repository web site. This redirection includes a "Redirection URI" representing the workflow environment that should be followed when the authorization process is complete in the EU repository.

- The user is presented with the login form of the EU Repo at the Common Access Point web site.

- The user fills in the username password and submits the form.

- The Common Access Point validates the credentials and if successful it asks the user to give permission to the Workflow application to access his/her data. If the user agrees, it redirects the user/browser back (C) to the 'Redirection URI' that was supplied in interaction (B). This redirection carries an "authorization code".

- The Workflow environment takes the authorization code and uses it to make a "behind the scenes" (i.e. without the user noticing it) request (D) to the Common Access Point to validate it.

- The Common Access Point (CAP) responds with an "access token" (E) that the Workflow Environment stores in the user's session and can be used in subsequent communication with the CAP.

- The Workflow Environment presents the welcome screen to the user (F).

All the communication is done over the "secure HTTP" protocol (HTTPS), which supports the integrity and non-repudiation of the transmitted messages since it is based on the Public Key Infrastructure (PKI) and digital certificates.

The models are retrieved from the model repositories where they are initially stored. In order to validate the requests and authorize the user, the requests from the workflow environment convey the OAuth "access token" that was acquired during the login phase as described above. The repositories therefore have enough information in order to perform any access control based on the user's identity and roles. After the successful response from the model repositories, the models descriptions as retrieved in TumorML are stored in a user specific "cache" database for the duration of his/her session. Any subsequent queries from the workflow editor (the "frontend") and any filters applied are using this "local" cache in the server side of the application. For each model its origin (EU or US repository) is also kept.

The workflows are stored in a custom document oriented database. For each workflow, the creator's id, the workflow name, description, and creation date are stored. Each workflow therefore is accessible only to the user who created it, i.e. it's private by default. The workflow description language follows a custom format and references the models used in the workflow by their TumorML identifiers. During the execution of the workflow, the "implementation" information of the models are important. The following is an XML snippet from the TumorML description of the EGFR-ERK Pathway model provided by the MGH and retrieved from the US repository:

```xml
…
<t:implementation id="urn:lsid:cvit.org:cmef:0.919920521935164">
  <t:title>EGFR-ODE Model for EC revision #3 (6/25/2012) from Massachusetts General
Hospital. Calculates Cell Cycle Time for EGF concentration. (Updated for command line
parameters)</t:title>
  <t:date>2012-06-25T00:00:00+00:00</t:date>
  <t:package name="EGFR_ODE_EC" checksum="">
    <t:file name="EGFR_ODE_EC" source="http://mgh-
cvit.infotechsoft.com:8080/repository_files/deisboeck/2012/5/25/EGFR_ODE-2012-06-25.zip"/>
  </t:package>
  <t:command>EGFR_ODE_EC $egf</t:command>
  <t:requirements>
    <t:operatingSystem>linux</t:operatingSystem>
    <t:CPUArchitecture>x86_64</t:CPUArchitecture>
    <t:language>cpp</t:language>
  </t:requirements>
</t:implementation>
…
```

The important information here is the "package" and the "command" tags. The "package" information references the binary code bundle as a Zip archive. The Workflow engine at workflow execution time downloads this file from the specified network location and "unzips" in a temporary directory. In order to execute the referenced model, it then "spawns" the specified command, passing in the command line the required parameters (the "egf" value in this case). The "requirements" information is also checked as a prerequisite in order to validate that the model will be able to run, prior to its execution[39].

The input parameters are usually passed in the command line specified in the "command" element or in an XML file, as described by the model's TumorML description. The output parameters are again either written by the model in an output XML file or produced as output files in the temporary directory that the model's code was extracted and run. The workflow engine checks which of these options is the actual case, and makes sure that the produced output values are passed further down the workflow graph to the other models. The execution status information is always forwarded to the user's browser in real time. The intermediate and the final results along with the workflow result status are stored also in the server side so that the user is able to see what the previous runs of the

---

[39] It can be the case that the same model has multiple implementations for different operating systems and CPU architectures. In this case, the workflow engine chooses the most applicable to its deployment.

workflows were, their input, output, intermediate values and results etc. This functionality is not yet supported in the "front-end" application (the workflow editor).

6.2.5 Implementation

The server side is implemented in Node.js[40], which is a Javascript framework for networking applications based on the V8 javascript engine[41] used in the Google Chrome browser and Chromium, its open source version. Node.js is event based and (by default) single threaded but it is highly praised for its scalability for IO bound applications, e.g. network proxies and the majority of the web applications[77]. In essence, the models are executed in separate processes, as standalone command line executable programs, so this has no impact on the main workflow engine process. Of course the models should have been implemented like this, i.e. standalone executable programs, which somehow restricts the model implementers. But in fact the approach is general because in TumorML there is information about how to get a whole "package" of the model that contains the required binaries, libraries etc. and the command that the workflow engine or a human user needs to run in order to execute the model. In this way even Matlab[TM] scripts can be used as implementations assuming that there's a Matlab installation on the workflow server side and the executable is a wrapper script around the invocation to the Matlab interpreter.

As mentioned above the workflows are stored in MongoDB database. MongoDB is an open source "document oriented" database that stores JSON[42] formatted documents. For the "cache" database where the models and user session information and other temporary information is kept they are using Redis[43], a memory based key value store. Both storage engines are examples of the NoSQL databases that are not using SQL and the relational model for the querying and manipulation of data.

6.3 p-medicine

The p-medicine project[44] (FP7-ICT-2009.5.3) is an ongoing project which develops an innovative and integrated technological solution to enable personalised medicine. The emphasis of the project is on formulating an open, modular framework of tools and services, so that p-medicine can be adopted gradually, including efficient secure sharing and handling of large personalized data sets, enabling demanding multiscale simulations (in silico oncology), building standards-compliant tools and models for VPH research, drawing on the VPH Toolkit and providing tools for large-scale, privacy-preserving data and literature mining, a key component of VPH research. The p-medicine tools and technologies will be validated within the concrete setting of advanced clinical research. Pilot cancer trials have been selected based on clear research objectives, emphasising the need to integrate multilevel datasets, in the domains of Wilms tumor, breast cancer and leukaemia. To sustain a self-supporting infrastructure realistic use cases will be built that will demonstrate tangible results for clinicians.

The p-medicine project is also relevant to the CHIC work plan, since it has to offer both in infrastructure as well as concrete use cases. In the context of p-medicine a data warehouse is being built which contains semantically annotated data collected from clinical trials, which can be exploited in the context of CHIC.

---

[40] http://www.nodejs.org/

[41] https://developers.google.com/v8/

[42] JSON (JavaScript Object Notation) is a text-based open standard designed for human-readable data interchange. More information can be found at http://json.org

[43] http://redis.io

[44] http://p-medicine.eu/

6.4 VPH-Share

VPH-Share[45] is an ongoing IP project funded by the EC, which is developing the infrastructure to allow users to share, and search for data and tools. The VPH-Share system will provide also the interface to run pre-constructed or user-defined workflows combining the available tools and running with the available data.   The VPH-Share infrastructure thus aims at providing the services and tools necessary to the biomedical community to share information and support the building of new knowledge.

While there are similarities in some ICT aspects, the main difference between CHIC and VPH-Share is that CHIC is focusing on a specific medical domain, the cancer one, while VPH-Share is going to provide a general purpose infrastructure. On the other hand, synergies might be sought during the CHIC infrastructure development to rely on services already provided by VPH-Share instead of re-implementing them or on the possibility to expose into the VPH-Share web portal some of the components developed by CHIC and of potential interest also to other medical communities.

VPH-Share is setting up a beta user program to involve institutions external to the project to the first VPH-Share services testing and use: CHIC is invited to be part of this programme. Moreover, VPH-Share is also organising a workshop in early 2014 to which the main VPH infrastructure projects will be invited to attend in order to share technical solution and favour integration and re-use of already available components.

---

[45] http://www.vph-share.eu/

7 Exemplar hypermodelling scenarios

7.1 *Short Description of CHIC clinical scenarios involving hypermodels*

Within the CHIC project the following cancer domains are addressed:

1. Nephroblastoma

2. Glioblastoma

3. Non-small cell lung cancer

4. Prostate cancer

Within each cancer domain a hypermodel will be developed that is based on a clinical scenario. Each scenario starts with a question that is relevant for clinicians. These questions need to be as easy as possible but also need to address a complicated or complex phenomenon that cannot be answered by current medical practice. The goal of such hypermodels will be to give a validated answer to clinicians that will provide them with better treatment options for individual patients.

To be successful in the development of hypermodels many iterative steps need to be done during the developmental process. It is important to understand that from an architectural perspective a hypermodel is always a composition of different component models (hypomodels). At the end the hypermodel will give a result, which is the answer of the initial asked clinical question.

7.1.1 Nephroblastoma

The most important question to be answered by the hypermodel for nephroblastoma is the following:

"Will the nephroblastoma of a patient shrink in response to preoperative chemotherapy?"

The answer to this question should be 'YES' or 'NO'. Shrinkage is determined by using imaging studies to measure the change in tumor volume. If the tumor volume is less than 75% of the initial tumor volume then the answer to the question is 'yes'. If there is an increase of more than 25 % in tumor volume after preoperative chemotherapy then the answer will be 'no'. If the tumor volume is between 75% and 125% of the volume before preoperative chemotherapy then the tumor volume is regarded as unchanged and therefore the answer is 'no' as well.

This question is of importance as all patients diagnosed with a kidney tumor between the ages of 6 months and 16 years will receive preoperative chemotherapy within the SIOP (International Society of Paediatric Oncology) protocol, if imaging studies suggest nephroblastoma as the most probable diagnosis. Chemotherapeutic treatment is currently based solely on imaging studies without histologically proven diagnosis. The reason to start with preoperative chemotherapy is the fact that 90% of tumors do shrink, making surgical removal of the nephroblastoma easier and resulting in downstaging of the tumor with less postoperative treatment. However in about 10% of patients the tumor will not shrink but increase in size. Such behaviour results in a worse situation for the patient that should be avoided. At present it is not possible to predict which tumors will shrink and which will not. By collecting all available data of a patient with nephroblastoma at the time of diagnosis the response to preoperative chemotherapy can hopefully be simulated with these data. If the developed model will predict the correct answer to the above described question patients will benefit from such an approach by applying them the best treatment right from the time of diagnosis. Physicians will only believe such a model if the results of the simulations are validated. Therefore this model needs to be developed with retrospective data and validated with prospective data in an iterative process.

The hypermodel for nephroblastoma will be an advanced oncosimulator simulating the response of chemotherapy on nephroblastoma in the computer (in silico).

More details are given in chapter 5 of deliverable D2.2.

7.1.2 Glioblastoma

The first question that needs to be answered by the Glioblastoma Multiforme (GBM) hypermodel is:

"Will a specific patient benefit from adding Dendritic Cell vaccination (DC vaccination) to the standard treatment for GBM?"

The answer should be 'YES' or 'NO' and will be answered by measuring Progression Free Survival (PFS) after 6 months (the confirmatory primary end-point of the phase IIb prospective double blind placebo-controlled randomized clinical trial HGG-2010 trial, EurdraCT 2009-018228-14). In the case of 'YES' there is a benefit of DC vaccination in terms of higher probability of reaching the 6 months PFS point (PFS 6m).

Since there is a more complex design in the study (cfr. below), patients in the placebo group receive vaccination after ±6 months in case relapse did not occur. A second question will hence be: Will a patient benefit more from early (within the first 6 months of standard treatment) or late (after 6 months of standard treatment) vaccination for overall survival.

A third question to be answered is which immune profile (cfr. below, cluster analysis) will predict good outcome after vaccination.

GBM is the most common primary brain cancer with an incidence of 3-4/100.000/year. Current standard treatment consists of maximal surgical resection, followed by 6 weeks of concomitant radiotherapy (30x2Gy) and Temozolomide, followed by 6 cycli of adjuvant Temozolomide (i.e. Stupp protocol [47]. Despite this multimodal treatment, median PFS is only 6.9 months and median OS 14.6 months. Hence, there is an urgent need for additional, safe and effective therapies.

DC vaccination has been studied for many years as an experimental therapy to treat GBM. After gross total or subtotal removal of the GBM, the tumor itself is processed in the laboratory to make whole tumor lysates. These whole tumor lysates contain multiple antigens expressed by the GBM. After surgery and after weaning of corticosteroids, a leucapheresis is performed to harvest a large amount of monocytes. These monocytes are cultured and differentiated to dendritic cells (DCs) under specific laboratory conditions. The dendritic cells are afterwards loaded/pulsed with tumor lysate, after which maturation is induced with a second cocktail of cytokines. Finally, the autologous mature lysate-loaded DCs (DCm-HGG-L) are injected back in the patient at specified moments (cfr. infra). The activated DCs will present the tumor antigen to specific cytotoxic T-cells, leading to an effective activation of the adaptive immune system and subsequent killing of residual or recurrent intracranial tumor cells.

There have been multiple study reports of DC vaccination for GBM, mainly after recurrence and relapse resection. We have incorporated DC vaccination in the standard therapy (cfr. schedule below) in the HGG-2006 phase I/II trial [48] proving the safety and feasability of this treatment in newly diagnosed GBM. This study also showed that RPA classification (cfr. infra) stratified patients on pretreatment variables and this was related with outcome. Although the previous clinical trials provide proof of principle and remarkable results in a subset of patients, not all patients benefit from DC vaccination. Defining the patient subgroup likely to benefit from this highly personalized and labor-intensive therapy is thus of clinical and economical importance.

We are currently running the HGG-2010 trial in which patients with newly diagnosed GBM are randomized between DC vaccination and placebo injections. After ±6 months (after completing adjuvant Temozolomide), there is an unblinding procedure so that placebo-treated patients can start

DC vaccination. The primary outcome of the study is PFS at 6 months. OS is a secondary outcome. Multiscale prospective data are collected (cfr. infra), and probably a combination of parameters can predict if a certain patient will benefit from DC vaccination.

Through hypermodelling within the CHIC environment, we want to explore if a patient with certain patient-, surgery- and tumor-related characteristics will have a response to DC vaccination, and which immune profiles are more likely to predict a good response.

After developing the oncosimulator, it will have to be validated in a next prospective patient cohort.

More details are given in chapter 6 of deliverable D2.2.

7.1.3 Non-Small-Cell Lung Cancer (NSCLC)

The most important question to be asked is the question:

"Can tumor-specific pathways predict the most promising therapy very early after tumor diagnosis?"

For that purpose a system biology model will be developed based on the transcriptome analysis of up to 100 tumor specimen to get new insights in the biology of NSCLC. This data will form the basis for the bottom-up approach of the in silico model for NSCLC and thus improving the accuracy of the developed in silico Hyper-Multiscale Models. All relevant clinical data, tumor typing according to the current adenocarcinoma classification as well as radiological, macroscopic, quantitative microscopic data, proliferation data and angiogenesis data were retrieved or collected prospectively from lung cancer resection specimens of NSCLC. Genetic profiles of the relevant pathways, miRNA data, and deep sequencing data of at least a limited number of well-defined NSCLCs, will be added for comprehensive analysis. Small biopsy samples of lung cancer, which are often the only tumor tissue available from patients with advanced NSCLC, manual dissection, laser-microdissection, quantitative few cell PCR approaches, DNA sequencing, biochip reverse-phase hybridization, mRNA preamplification and whole genome amplification are available as well as epidemiological and follow-up parameters from the Saarland Tumor Center and will be integrated into the Models for In Silico Oncology in order facilitate the therapy-related clinical decisions.

An in-depth analysis is given in chapter 7 of deliverable D2.2.

7.1.4 Prostate cancer

The main question of our investigation is related to the management of the biochemical recurrences that sometimes follow the surgical/radiotherapy radical approach of the prostate cancer (see Figure 19).

As a matter of fact, this scenario is the most dramatic one both for the patient and for the clinician, who faces the problem with salvage therapies and/or hormonal therapy (Androgen Deprivation Therapy) but often doesn't know enough about timing, dosage and success probability of such actions.

Modeling the natural evolution of the pathology and/or the effects of the therapies maybe extremely useful provided the models are properly validated on robust clinical data.
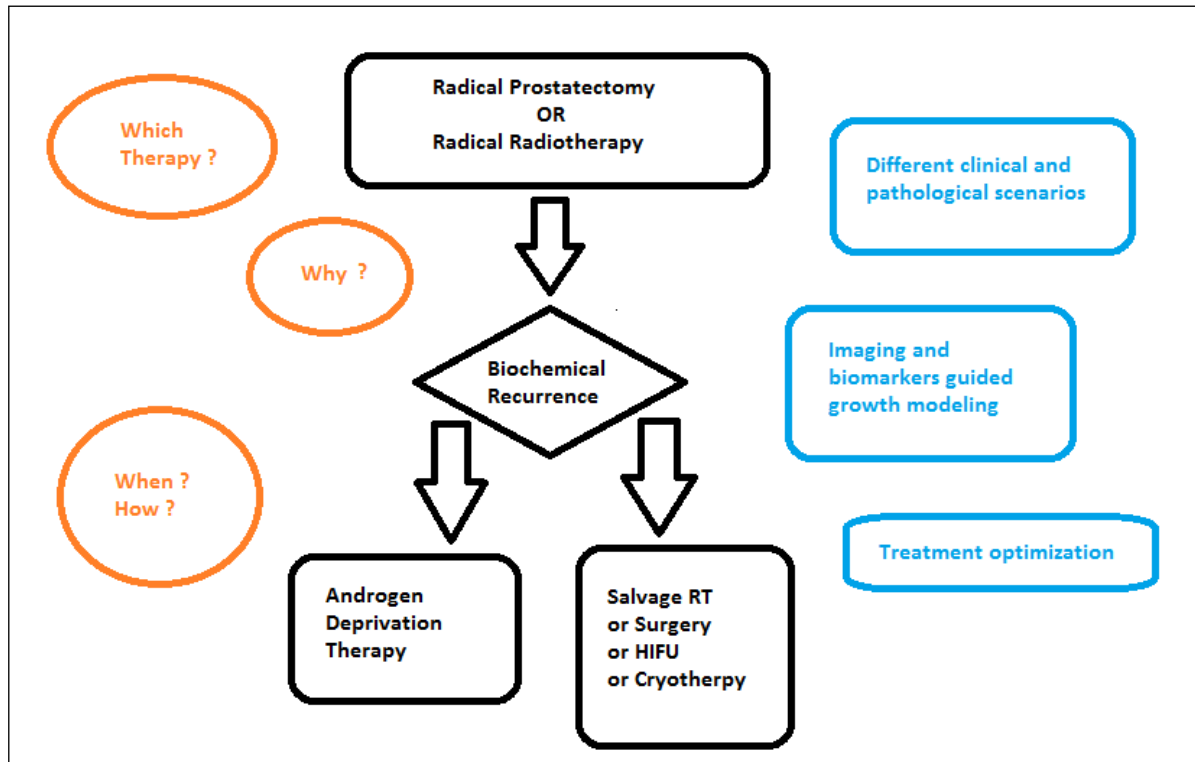
Figure 19: Rationale of prostate cancer studies.

A great extent of "hidden" knowledge is potentially available querying properly the clinical data from patients affected by the same disease. However there are many issues, from a practical (data are often on paper, such as medical records, in untidy and often incomplete form), ethical and, especially, methodological point of view.

Large data collections are normally interrogated using more or less complicated statistical models, in order to get information from a more or less homogeneous population, or to build nomograms and/or practical indications about therapies. Much less often they are used to properly validate mathematical models founded on biological assumptions, and to estimate with great accuracy the values of biologically relevant parameters.

In the framework of the European ICT Project CHIC focused on the building of a common repository for data and models in the field of human cancer, the activity of each research group is challenged by a number of requirements:

- data should be representative (similar in pertinence, accuracy, provenience, without bias in selection and treatment);

- data should be properly pseudo-anonymized to be shared among groups in different countries, with possibly different legal issues; furthermore valid informed consent and/or ethical approval from relevant regulatory bodies to patient data use should be obtained;

- data should be collected in a database easily included into larger or more structured ones;

- the database size and structure are selected in order to be able to host also diagnostic images and metadata, provisionally available from further perspective studies or analyses;

- data should be easily queried, using different statistical tools and different data mining approaches;

- data need to be protected against involuntary or voluntary hackerage;

- data should be easily usable to validate models proposed by researchers, using different software and requiring different input formats.

In this context the prostate cancer group would specify three primary goals:

1) To fill in a huge database about prostate cancer in an homogeneous cohort (Italian Piedmont region population) with thousands of patients and about a hundred multi-scale fields, including clinical, serological, pathological, molecular and radiological data;

2) For the retrospective studies to develop a statistical model about reliable correlations between prognosis and risk factors, several of them already present in the existent formulas and some of them, like PSA doubling time, collected specifically and extensively by the present study;

3) For the perspective studies to develop mathematical models, more deterministic than the previous ones, taking into consideration, beyond the classical risk parameters, molecular markers and tumor volume, density, cellular burden and dynamics of growth estimated through imaging and molecular biology assays.

More details are given in chapter 8 of deliverable D2.2.

8   References

[1]   G.S.Stamatakos, D.D. Dionysiou Introduction of Hypermatrix and Operator Notation into a
        Discrete Mathematics Simulation Model of Malignant Tumour Response to Therapeutic
        Schemes In Vivo. Some Operator Properties Cancer Informatics 7, 239 - 251, 2009

[2]   G.Stamatakos "In Silico Oncology Part I: Clinically Oriented Cancer Multilevel Modeling Based
        on Discrete Event Simulation" In T.Deisboeck and G. Stamatakos Eds " Cancer Multiscale
        Modeling, " CRC Press, pp. 407-436. Print ISBN: 978-1-4398-1440-6 eBook ISBN: 978-1-4398-
        1442-0 DOI: 10.1201/b10407-19 Boca Raton, Florida, USA, 2010

[3]   T. Deisboeck and G. S . Stamatakos Eds. Multiscale Cancer Modeling. CRC Press 2010. Pp. 407–
        436. Print ISBN: 978-1-4398-1440-6, eBook ISBN: 978-1-4398-1442-0. DOI: 10.1201/b10407-19

[4]   HM Byrne (2010). Dissecting cancer through mathematics: from the cell to the animal model.
        Nature Reviews Cancer.10(3): 221-230.

[5]   D.D.Dionysiou, G.S. Stamatakos, N.K.Uzunoglu, K.S.Nikita, A. Marioli , A Four Dimensional In
        Vivo Model of Tumour Response to Radiotherapy: Parametric Validation Considering
        Radiosensitivity, Genetic Profile and Fractionation , J. theor. Biol. , 230 , 1-20 , 2004

[6]   C.May, E.Kolokotroni, G. StamatakoS, P.Buechler Coupling biomechanics to a cellular level
        model: an approach to patient-specific image driven multi-scale and multi-physics tumor
        simulation Progress in Biophysics and Molecular Biology, vol. 107(1) pp. 193-199 2011

[7]   Project Success Stories – In-Silico medicine reaches the clinic
        http://cordis.europa.eu/fetch?CALLER=PRINT_OFFR&SESSION=&ACTION=D&RCN=6061 (
        webpage visited on 1 Jan. 2014)

[8]   G.S.Stamatakos, E.Ch.Georgiadi, N.Graf, E.A.Kolokotroni, and D.D.Dionysiou, "Exploiting
        Clinical Trial Data Drastically Narrows the Window of Possible Solutions to the Problem of
        Clinical Adaptation of a Multiscale Cancer Model", PLoS ONE 6(3), e17594, 2011

[9]   A. J. Shih, S.E.T., R. Radhakrishnan, Analysis of Somatic Mutations in Cancer: Molecular
        Mechanisms of Activation in the ErbB family of Receptor Tyrosine Kinases. Cancers, 2011. 3(1):
        p. 1195-1231.

[10]   R. Radhakrishnan and Tamar Schlick, Orchestration of cooperative events in DNA synthesis and
        repair mechanism unraveled by transition path sampling of DNA polymerase beta's closing.
        Proc. Nat. Acad. Sci. 05/2004; 101(16):5970-5. DOI: 10.1073/pnas.0308585101

[11]   G. Stamatakos, D. Dionysiou, A. Lunzer, R. Belleman, E. Kolokotroni, E. Georgiadi, M. Erdt, J.
        Pukacki, S. Rueping, S. Giatili, A. d' Onofrio, S. Sfakianakis, K. Marias, C. Desmedt, M. Tsiknakis,
        and N. Graf, "The Technologically Integrated Oncosimulator:Combining Multiscale Cancer
        Modeling with Information Technology in the In Silico Oncology Context IEEE Journal of
        Biomedical and Health Informatics, 2014 (in press)

[12]   G. S Stamatakos, N. Graf and R. Radhakrishnan, Multiscale Cancer Modeling and In Silico
        Oncology: Emerging Computational Frontiers in Basic and Translational Cancer Research,
        Editorial, J Bioengineer & Biomedical Sci 2013, 3:2 http://dx.doi.org/10.4172/2155-
        9538.1000e114

[13]   G.S.Stamatakos, E.Ch.Georgiadi, N.Graf, E.A.Kolokotroni, and D.D.Dionysiou, "Exploiting
        Clinical Trial Data Drastically Narrows the Window of Possible Solutions to the Problem of
        Clinical Adaptation of a Multiscale Cancer Model", PLOS ONE 6(3), e17594 2011

[14]   G.S.Stamatakos, E.A.Kolokotroni, D.D.Dionysiou, E.Ch.Georgiadi, C.Desmedt. "An advanced
        discrete state - discrete event multiscale simulation model of the response of a solid tumor to

chemotherapy: Mimicking a clinical study." Journal of Theoretical Biology 266, 124-139, 2010

[15] D.D. Dionysiou, G.S. Stamatakos, D. Gintides, N. Uzunoglu, K. Kyriaki "Critical Parameters Determining Standard Radiotherapy Treatment Outcome for Glioblastoma Multiforme: A Computer Simulation," The Open Biomedical Engineering Journal 2, pp. 43-51, 2008

[16] N. Graf, A. Hoppe, E. Georgiadi, R. Belleman, C. Desmedt, D. Dionysiou, M. Erdt, J. Jacques, E. Kolokotroni, A. Lunzer, M. Tsiknakis, G. Stamatakos, " In Silico Oncology for Clinical Decision Making in the Context of Nephroblastoma." Klinische Pädiatrie 221, pp.141-149, 2009.

[17] G.S. Stamatakos, D.D. Dionysiou, E.I. Zacharaki, N.A. Mouravliansky, K.S.Nikita, N.K. & Uzunoglu, "In silico radiation oncology: combining novel simulation algorithms with current visualization techniques," IEEE Proceedings: Special Issue on Bioinformatics: Advances and Challenges , 90(11) , pp. 1764-1777 , 2002

[18] D.D.Dionysiou, G.S. Stamatakos, N.K.Uzunoglu, K.S.Nikita, A. Marioli, "A Four Dimensional In Vivo Model of Tumour Response to Radiotherapy: Parametric Validation Considering Radiosensitivity, Genetic Profile and Fractionation ," J. Theor. Biol. , 230 , 1-20 , 2004

[19] G.S.Stamatakos, V.P.Antipas, and N.K. Uzunoglu, "A spatiotemporal, patient individualized simulation model of solid tumor response to chemotherapy in vivo: the paradigm of glioblastoma multiforme treated by temozolomide ," IEEE Transactions on Biomedical Engineering , 53(8) , pp. 1467-1477 , 2006

[20] G.S.Stamatakos, D.D.Dionysiou, N.M.Graf, N.A.Sofra, C.Desmedt, A.Hoppe, N.Uzunoglu, M.Tsiknakis , "The Oncosimulator: a multilevel, clinically oriented simulation system of tumor growth and organism response to therapeutic schemes. Towards the clinical evaluation of in silico oncology," in Proceedings of the 29th Annual International Conference of the IEEE EMBS Cite Internationale, August 23-26, SuB07.1:pp. 6628-6631 , Lyon, France , 2007

[21] E.A. Kolokotroni, D.D. Dionysiou, N.K. Uzunoglu, G.S. Stamatakos, "Studying the growth kinetics of untreated clinical tumors by using an advanced discrete simulation model", Mathematical and Computer Modelling," 54, pp. 1989-2006, 2011

[22] Georgiadi ECh, Dionysiou DD, Graf N, Stamatakos GS, "Towards in silico oncology: adapting a four dimensional nephroblastoma treatment model to a clinical trial case based on multi-method sensitivity analysis." Comput Biol Med. 2012 Nov;42(11):1064-78.

[23] Ouzounoglou, E.N. ; Dionysiou, D.D. ; Stanulla, M. ; Stamatakos, G.S., "Towards patient personalization of an Acute Lymphoblastic Leukemia Model during the oral administration of prednisone in children: Initiating the ALL Oncosimulator," Proc. Advanced Research Workshop on In Silico Oncology and Cancer Investigation - The TUMOR Project Workshop (IARWISOCI), 2012 5th International, IEEE Xplore INSPEC Accession Number: 13309781

[24] Argyri, K.D. ; Dionysiou, D.D. ; Stamatakos, G.S. "Modeling the interplay between pathological angiogenesis and solid tumor growth: The anti-angiogenic treatment effect," Proc. Advanced Research Workshop on In Silico Oncology and Cancer Investigation - The TUMOR Project Workshop (IARWISOCI), 2012 5th International, IEEE Xplore INSPEC Accession Number: 13325485

[25] Giatili, S and G. Stamatakos, "A detailed numerical treatment of the boundary conditions imposed by the skull on a diffusion–reaction model of glioma tumor growth. Clinical validation aspects," Applied Mathematics and Computation 218 (2012) 8779–8799.

[26] The "Clinically Oriented Translational Cancer Multilevel Modelling" Project http://www.contracancrum.eu/ (webpage visited on 1 Jan. 2014)

[27] The "Osteoporotic Virtual Physiological Human" Project http://en.wikipedia.org/wiki/VPHOP

(webpage visited on 1 Jan. 2014)

[28] Thiel R, Stroetmann KA, Stroetmann VN, Viceconti M., Designing a socio-economic assessment method for integrative biomedical research: the Osteoporotic Virtual Physiological Human project. Stud Health Technol Inform. 2009;150:876-80.

[29] The "Transatlantic Tumour Model Repositories" Project http://tumor-project.eu/ (webpage visited on 1 Jan. 2014)

[30] K. Marias, D. Dionysiou, V. Sakkalis, N. Graf, R. M. Bohle, P. V. Coveney, S. Wan, A. Folarin, P. Buechler, M. Reyes, G. Clapworthy, E. Liu, J. Sabczynski, T. Bily, A. Roniotis, M. Tsiknakis, E. Kolokotroni, S. Giatili, C. Veith, E. Messe, H. Stenzhorn, Yoo-Jin Kim, S. Zasada, A. N. Haidar, C. May, S. Bauer, T. Wang, Y. Zhao, M. Karasek, R. Grewer, A. Franz and G. Stamatakos, Clinically driven design of multi-scale cancer models: the ContraCancrum project paradigm, Interface Focus, June 6, 2011 1:281-285 2011

[31] Hanahan D, Weinberg RA: The hallmarks of cancer. Cell 100:57–70, 2000

[32] Hanahan D, Weinberg RA: Hallmarks of cancer: the next generation. Cell 144:646–674, 2011

[33] Hainaut P, Plymoth A: Targeting the hallmarks of cancer: towards a rational approach to next-generation cancer therapy. Curr Opin Oncol 25:50–51, 2013

[34] Stamatakos GS, Graf N, Radhakrihnan R: Multiscale Cancer Modeling and In Silico Oncology: Emerging Computational Frontiers in Basic and Translational Cancer Research. J Bioengineer & Biomedical Sci 3:2, 2013; http://dx.doi.org/10.4172/2155-9538.1000e114

[35] Huang M, Shen A, Ding J, Geng M: Molecularly targeted cancer therapy: some lessons from the past decade. Trends Pharmacol Sci 013 Dec 19. pii: S0165-6147(13)00220-4. doi: 10.1016/j.tips.2013.11.004. [Epub ahead of print]

[36] "X-ray crystallography," [Online]. Available: http://en.wikipedia.org/wiki/X-ray_crystallography.

[37] N. m. r. spectroscopy. [Online]. Available: http://en.wikipedia.org/wiki/Nuclear_magnetic_resonance_spectroscopy.

[38] "High-throughput screening," [Online]. Available: http://en.wikipedia.org/wiki/High-throughput_screening.

[39] "NAMD," [Online]. Available: http://www.ks.uiuc.edu/Research/namd/.

[40] "VMD," [Online]. Available: http://www.ks.uiuc.edu/Research/vmd/.

[41] "Carma," [Online]. Available: http://utopia.duth.gr/~glykos/Carma.html.

[42] "AutoDock Vina," [Online]. Available: http://autodock.scripps.edu.

[43] "Glide," [Online]. Available: http://www.schrodinger.com/productpage/14/5/.

[44] "RCSB PDB," [Online]. Available: http://www.rcsb.org/pdb/home/home.do.

[45] "ZINC," [Online]. Available: http://zinc.docking.org/search/structure.

[46] "PubChem," [Online]. Available: http://pubchem.ncbi.nlm.nih.gov/.

[47] "PDBbind," [Online]. Available: http://sw16.im.med.umich.edu/databases/pdbbind/index.jsp.

[48] "The R Project for Statistical Computing," [Online]. Available: http://www.r-project.org/.

[49] "MATLAB," [Online]. Available: http://www.mathworks.com/products/matlab/.

[50] "KEGG: Kyoto Encyclopedia of Genes and Genomes," [Online]. Available: http://www.genome.jp/kegg/.

[51]   "PANTHER Gene List Analysis," [Online]. Available: http://www.pantherdb.org/.

[52]   "Reactome," [Online]. Available: http://www.reactome.org/.

[53]   "SBML.org," [Online]. Available: http://sbml.org/Main_Page.

[54]   "BIOPAX," [Online]. Available: http://www.biopax.org/.

[55]   "MIRIAM," [Online]. Available: http://co.mbine.org/standards/miriam.

[56]   "Cell Designer," [Online]. Available: http://www.celldesigner.org/.

[57]   "Edinburgh Pathway Editor," [Online]. Available: http://epe.sourceforge.net/SourceForge/EPE.html.

[58]   "PathVisio," [Online]. Available: http://www.pathvisio.org/.

[59]   "The CellML project," [Online]. Available: http://www.cellml.org/.

[60]   "COPASI," [Online]. Available: http://www.copasi.org/.

[61]   "Systems Biology Workbench," [Online]. Available: http://sbw.sourceforge.net/.

[62]   Jeremy E. Purvis et al., "Linking Oncogenic Signaling to Molecular Structure," in Multiscale Cancer Modeling, CRC Press, 2011, pp. 31-44.

[63]   "SemanticSBML," [Online]. Available: http://semanticsbml.org/.

[64]   "Physiome Project," [Online]. Available: http://physiomeproject.org/.

[65]   "Virtual Physiological Human network of excellence," [Online]. Available: http://www.vph-noe.eu/.

[66]   "HD Physiology," [Online]. Available: http://hd-physiology.jp/.

[67]   PhysioDesigner. [Online]. Available: http://physiodesigner.org/.

[68]   "InSilicoML," [Online]. Available: http://www.physiome.jp/wiki/isml:specifications.

[69]   "RICORDO project," [Online]. Available: http://www.ricordo.eu/.

[70]   "SemSim," [Online]. Available: http://sbp.bhi.washington.edu/projects/semsim.

[71]   "SemGen," [Online]. Available: http://sbp.bhi.washington.edu/projects/semgen.

[72]   Viceconti M 2011 A tentative taxonomy for predictive models in relation to their falsifiability. Philos Transact A Math Phys Eng Sci 369(1954):4149-61

[73]   "Automated Web Service Composition: State of the Art and Research Challenges", Technical Report ICS-FORTH/TR-409, http://www.ics.forth.gr/tech-reports/2010/2010.TR409_Automated_Web_Service_Composition.pdf

[74]   Le Zhang Sean Martin, Thomas S Deisboeck, "Advancing Cancer Systems Biology: Introducing the Center for the Development of a Virtual Tumor, CViT," *CIN*, vol. 5, p. 1, 2007.

[75]   Le Novère N., Finney A., Hucka M., Bhalla U., Campagne F., Collado-Vides J., Crampin E., Halstead M., Klipp E., Mendes P., Nielsen P., Sauro H., Shapiro B., Snoep J.L., Spence H.D., Wanner B.L. (2005) Minimum Information Requested In the Annotation of biochemical Models (MIRIAM) Nature Biotechnology, 23: 1509-1515.

[76]   D. Johnson, J. Cooper, and S. McKeever, "TumorML: Concept and requirements of an in silico cancer modelling markup language," presented at the Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, 2011, pp. 441–444.

[77]   S. Tilkov and S. Vinoski, "*Node.js: using Javascript to build high-performance network programs,*" Internet Computing, IEEE, vol. 14, no. 6, pp. 80–83, 2010.

[78]   Munindar P. Singh, Mladen A. Vouk "Scientific Workflows: Scientific Computing Meets Transactional Workflows", Position paper in *Reference Papers of the NSF Workshop on Workflow and Process Automation in Information Systems: State-of- the-art and Future Directions*, May 1996.
http://www.csc.ncsu.edu/faculty/mpsingh/papers/databases/workflows/sciworkflows. html

[79]   Object Management Group, Business Process Modeling Notation,
http://www.omg.org/spec/BPMN/2.0/

[80]   R. T. Fielding and R. N. Taylor. Principled design of the modern Web architecture. ACM Transactions on Internet Technologies, 2(2):115-150, 2002.

[81]   D. Berardi, D. Calvanese, G. D. Giacomo, M. Lenzerini, and M. Mecella. Automatic service composition based on behavioral descriptions. Int. J. Cooperative Inf. Syst., 14(4):333-376, 2005.

[82]   D. Berardi, F. Cheikh, G. D. Giacomo, and F. Patrizi. Automatic service composition via simulation. Int. J. Found. Comput. Sci., 19(2):429-451, 2008.

[83]   R. Gronmo and M. C. Jaeger. Model-driven semantic web service composition. In APSEC '05: Proceedings of the 12th Asia-Pacic Software Engineering Conference, pages 79-86, Washington, DC, USA, 2005. IEEE Computer Society.

[84]   D. Skogan, R. Gronmo, and I. Solheim. Web service composition in uml. Enterprise Distributed Object Computing Conference, IEEE International, 0:47-57, 2004.

[85]   S. Narayanan and S. A. McIlraith. Simulation, Verication and Automated Composition of Web services. In WWW, pages 77-88, 2002.

[86]   R. Hamadi and B. Benatallah. A petri net-based model for web service composition. In ADC '03: Proceedings of the 14th Australasian database conference, pages 191-200, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[87]   A. Brogi and S. Corni. Ontology- and behavior-aware discovery of web service compositions. Int. J. Cooperative Inf. Syst., 17(3):319-347, 2008.

[88]   W. Fan, F. Geerts, W. Gelade, F. Neven, and A. Poggi. Complexity and composition of synthesized web services. In M. Lenzerini and D. Lembo, editors, PODS, pages 231-240. ACM, 2008.

[89]   R. Milner. Communication and concurrency. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[90]   R. Milner. Communicating and mobile systems: the pi-calculus. Cambridge University Press, fifth edition, 2004.

[91]   N. Milanovic and M. Malek. Current solutions for web service composition. IEEE Internet Computing, 8(6):51{59, 2004.

[92]   J. Rao, P. Kungas, and M. Matskin. Logic-based web services composition: From service description to process model. In ICWS, pages 446-453. IEEE Computer Society, 2004.

[93]   S. Beauche and P. Poizat. Automated service composition with adaptive planning. In A. Bouguettaya, I. Kruger, and T. Margaria, editors, ICSOC, volume 5364 of Lecture Notes in Computer Science, pages 530-537, 2008.

[94]   M. Ghallab, C. K. Isi, S. Penberthy, D. E. Smith, Y. Sun, and D.Weld. Pddl – the planning domain definition language. Technical report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.

[95]   P. Bertoli, M. Pistore, and P. Traverso. Automated composition of web services via planning in asynchronous domains. Artif. Intell., 174(3-4):316-361, 2010.

[96]   F. Lecue, A. Leger, and A. Delteil. DL Reasoning and AI Planning for Web Service Composition. In Web Intelligence, pages 445-453. IEEE, 2008.

[97]   M. Klusch and A. Gerber. Semantic web service composition planning with owlsxplan. In Proceedings of the 1st Int. AAAI Fall Symposium on Agents and the Semantic Web, pages 55-62, 2005.

[98]   M. Phan and F. Hattori. Automatic web service composition using congolog. In ICDCS Workshops, page 17. IEEE Computer Society, 2006.

[99]   M. Pistore, A. Marconi, P. Bertoli, and P. Traverso. Automated composition of web services by planning at the knowledge level. In IJCAI, pages 1252-1259, 2005.

[100] M. Pistore, P. Traverso, and P. Bertoli. Automated composition of web services by planning in asynchronous domains. In ICAPS, pages 2-11, 2005.

[101] J. Rao, D. Dimitrov, P. Hofmann, and N. M. Sadeh. A mixed initiative approach to semantic web service discovery and composition: Sap's guided procedures framework. In ICWS, pages 401-410. IEEE Computer Society, 2006.

[102] I. Paik and D. Maruyama. Automatic web services composition using combining htn and csp. In CIT, pages 206-211. IEEE Computer Society, 2007.

[103] S. A. McIlraith and T. C. Son. Adapting golog for composition of semantic web services. In D. Fensel, F. Giunchiglia, D. L. McGuinness, and M.-A. Williams, editors, KR, pages 482-496. Morgan Kaufmann, 2002.

[104] S. Sohrabi and S. A. McIlraith. Optimizing web service composition while enforcing regulations. In ISWC 2009: Proceedings of the 8th International Semantic Web Conference, Chantilly, VA, USA, pages 601-617, 2009.

[105] S. Sohrabi, N. Prokoshyna, and S. A. McIlraith. Web service composition via the customization of golog programs with user preferences. In A. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. K. Yu, editors, Conceptual Modeling: Foundations and Applications, volume 5600 of Lecture Notes in Computer Science, pages 319-334. Springer, 2009.

[106] M. Trainotti, M. Pistore, G. Calabrese, G. Zacco, G. Lucchese, F. Barbon, P. Bertoli, and P. Traverso. Astro: Supporting composition and execution of web services. In ICSOC, pages 495-501, 2005.

[107] Z. Wu, A. Ranabahu, K. Gomadam, A. P. Sheth, and J. A. Miller. Automatic composition of semantic web services using process and data mediation. Technical report, Kno.e.sis Center, Wright State University, 2 2007.

[108] J. Peer. A pop-based replanning agent for automatic web service composition. In A. Gomez-Perez and J. Euzenat, editors, ESWC, volume 3532 of Lecture Notes in Computer Science, pages 47-61. Springer, 2005.

Appendix – Abbreviations and acronyms

AI	Artificial Intelligence
API	Application Programming Interface
BIOPAX	Biological Pathways exchange
BPEL	Business Process Execution Language
BSD	Berkeley Software Distribution
CAP	Common Access Point
CBM	Computer-Based Manipulative
CCS	Calculus of Communicating Systems
COPASI	Complex Pathway Simulator
CORBA	Common Object Request Broker Architecture
DAG	Directed Acyclic Graph
DNA	Deoxyribonucleic acid
EGFR	Epidermal Growth Factor Receptor
FSM	Finite State Machine
GBM	Glioblastoma Multiforme
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
IDE	Integrated Development Environment
IPC	Inter-Process Communication
ISML	In-Silico Markup Language
ITK	Insight Segmentation and Registration Toolkit
IUPS	International Union of Physiological Science
JSON	JavaScript Object Notation
KEGG	Kyoto Encyclopedia of Genes and Genomes
MAF	Multimod Application Framework
MGH	Massachusetts General Hospital
MIRIAM	Minimum Information Required in the Annotation of Models
NGS	Next Generation Sequencing

NSCLC	Non-Small-Cell Lung Cancer
ODE	Ordinary Differential Equations
OWL	Web Ontology Language
OWL-S	Semantics OWL / Semantic Markup for Web Services
PKI	Public Key Infrastructure
QoS	Quality of Service
REST	Representational State Transfer
RMI	Remote Method Invocation
RNA	Ribonucleic acid
RPC	Remote Procedure Call
RSCB - PDB	Research Collaboratory for Structural Bioinformatics - Protein Data Bank
SAWSDL	Semantic Annotations for WSDL
SBML	Systems Biology Markup Language
SCA	Service Component Architecture
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSO	Single Sign On
SWSF	Semantic Web Services Framework
SWSL	Semantic Web Services Language
SWSO	Semantic Web Services Ontology
UML	Unified Modelling Language
VMD	Visual Molecular Dynamics
VPH	Virtual Physiological Human
VTK	Visualization Toolkit
WFMC	Workflow Management Coalition
WFRM	Workflow Reference Model
WS-BPEL	Web Services Business Process Execution Language
WS-CDL	Web Services Choreography Description Language
WSDL	Web Service Description Language
WSFL	Web Services Flow Language

WSMO	Web Service Modeling Ontology
XML	Extensible Markup Language