International Journal on Advances in Life Sciences, vol 8 no 1 & 2, year 2016, http://www.iariajournals.org/life_sciences/

Multiscale Cancer Modelling in Terms of Copyright

Iryna Lishchuk, Marc Stauch
Institut für Rechtsinformatik
Leibniz Universität Hannover
Hannover, Germany
e-mail: lishchuk@iri.uni-hannover.de, stauch@iri.uni-hannover.de

***Abstract*—Multiscale cancer modelling is a complex process, which requires interdisciplinary effort. Simulation of tumor progression across multiple scales by computer models is a challenge for scientists, and determining applicable legal protection is a challenge for lawyers. Insofar as a computer model is defined as a computer program, software copyright comes into play. At this stage several questions arise: What elements in computer modelling are copyrightable? Is the modelling work protected? What about copyright in a hyper-model design? The intellectual effort and investment deployed into the modelling have been tested against the requirements of copyright. In fact, various elements in cancer modelling, which express original creative input, qualify as copyright works and are protected as such.**

Keywords-computer models; software copyright; copyright in compilations; non-literal expression; program expression.

I. INTRODUCTION

Computational modelling is one of the IT tools applied in support of the oncology of today. Simulation of cancer progression by computer models allows predicting variations more efficiently, saving time, money, and materials. The potential of computer modelling is that the models can be used as clinical decision support systems in future. For this, a decision based on prediction of a model must prove better than a decision based on the clinical standards of today. Apart from scientific and technological issues raised by computer modelling, a number of legal issues need to be solved before the cancer models are released into practice. One of the legal issues relates to protectability of cancer models by copyright [1]. Who is entitled to do what with existing models and data sets? What elements are copyrightable and what elements are void of copyright? How to deal with the risk that cancer models could be regarded as mere ideas and not protected as such? Before we approach and suggest answers to these questions, let us explore the substance of cancer modelling first.

Cancer is a complex disease. Heterogeneous types of tumor cells, uncontrolled behavior and invasion of tumor into the healthy tissue, interplay among the cells themselves and the microenvironment make cancer a challenging object for research and treatment [2]. The oncology of today requires an interdisciplinary approach and is increasingly supported by specialized software solutions. *In silico* cancer modelling is one of the IT solutions in this field. *In silico* oncology aims to improve cancer knowledge and treatment by creating reliable computer predictions.

Simulation of cancer progression in space and time requires the use of multiscale cancer modelling. Multiscaling is realized *in silico* by constructing elementary models (the ones, which correspond to elementary biological processes) and relation models (the ones, which reflect relations across them) into multiscale hyper-models [3]. "*A model is considered to be "multiscale" if it spans two or more different spatial scales and/or includes processes that occur at two or more temporal scales*." [2]. The four main biological scales, which are being modelled are the atomic, molecular, microscopic and macroscopic scales [2]. Processes, which occur at the atomic level, are linked to the processes at a higher level. The composite multiscale constructs of models (hyper-models or integrative models) are then able to synthesize and imitate the biological processes at several temporal and spatial levels (molecular, cellular, etc.) at once.

Research on multiscale cancer modelling is ongoing. In particular, the ICT research project CHIC, full title "*Computational Horizons In Cancer (CHIC): Developing Meta- and Hyper-Multiscale Models and Repositories for In Silico Oncology*", conducts research on multiscale cancer modelling [3]. The CHIC project "*proposes the development of clinical trial driven tools, services and secure infrastructure that will support the creation of multiscale cancer hyper-models (integrative models).*" [3]. The research groups from the different project partner institutions contribute single-scale models (from molecular to compartment models), which are then combined into integrated multiscale hyper-models. Linking and interplay between the models in CHIC is shown in Figure 1[4].

In general, the process of multiscale cancer modelling can be divided into three main stages:

(a) Scientific modelling: at this stage, modelers study the tumor types and biological processes, selected for simulation, investigate the types of cells and interactions among them, define algorithms and modelling techniques, which are capable to capture such processes best.

(b) Coding: at this stage, the tumor models are transformed into executable form. In this process, either the already developed tumor models are broken down into simpler models or computer codes of elementary biological processes (biomechanics) are developed anew.

(c) Hyper-modelling: in this step, the elementary models, each of which represents a biological process at a single scale, are combined with the other models into multiscale hyper-models. In this course, spatiotemporal simulation of tumor types, addressed by the CHIC project,

i.e., Wilms tumor, glioblastoma multiforme (GBM) and non small cell lung cancer (NSCLC), is achieved [3].

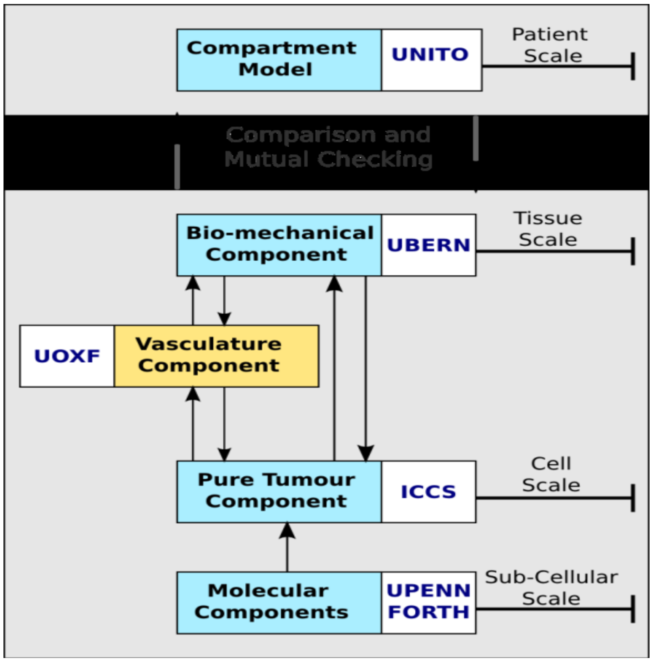

Figure 1. Schematic of the planned modelling framework for the CHIC project with the angiogenesis/vascular component highlighted.

The research on cancer modelling is of clinical relevance and is motivated by the perspective of using multiscale cancer models as a clinical decision support tool [4]. Meanwhile, the legal research in the project is concerned with the legal implications surrounding amalgamation of models. In terms of Intellectual Property (IP) law, this includes identifying the type of protection applicable to cancer models and hyper-models, the limits of such protection, and the conditions that protectable elements of the models must fulfill in order to be protected.

The cancer models, encoded into computer programs, constitute subject matter protectable by copyright [5]. However, these are not only the codes of computer models themselves, which may enjoy such protection. Also, the scientific modelling work as well as designing models into hyper-models may be copyrightable.

In this paper, we explore copyright, as a type of protection applicable to the cancer modelling, investigate in how far the scope of copyright reaches to protect the intellectual input and investment, deployed in the cancer modelling. We start with the overview of general principles of copyright in Section II, look at how it may apply in turn to three main aspects of modelling: coding models (Section III), structuring of the hyper-models (Section IV), and scientific modelling (Section V). Conclusions finalize the paper in Section VI.

II. WHAT COPYRIGHT PROTECTS

The law of copyright has its requirements, which all copyrighted works must fulfill in order to be protected. These criteria are examined below.

A. Protection of Computer Programs by Copyright

Copyright is a traditional type of protection, which both European and International law grant to computer programs. Article 4 WIPO Copyright Treaty [6] and Article 10 TRIPS Agreement [5] protect computer programs as literary works within the meaning of the Berne Convention (1886) [7]. The same principle is followed by European copyright law. Article 1 of Directive 2009/24/EC on the legal protection of computer programs (Software Directive) [8] recognizes computer programs as an object of copyright protection in the EU.

B. What Copyright Protects

The Software Directive protects programs, which are "*original in the sense that it is the author's own intellectual creation."* [8]. Original intellectual creation is a basic requirement for copyright protection. It is equally applicable to the other copyright works, be it software, writings, photographic works, or other works protectable by copyright [7]. No other criteria, such as whether a program is functional, or how many lines of code it has, etc. are relevant for copyright. However, the requirement of originality has its own interpretation in terms of copyright.

As interpreted by the Court of Justice of the European Union (CJEU), which dealt with copyrightability of text extracts from articles in the case *Infopaq International A/S*, "*it is only through the choice, sequence and combination of those words that the author may express his creativity in an original manner and achieve a result which is an intellectual creation."* [9].

In the case *SAS Institute Inc.* [10], which concerned copyright in a manual to a computer program, the court applied the same interpretative criteria in relation to copyrightability of elements, which appear in the manual and/or are implemented by a computer program. In doing so, the court reached the conclusion that "*the keywords, syntax, commands and combinations of commands, options, defaults and iterations ..., figures or mathematical concepts ..., considered in isolation"* are not an intellectual creation of the author of a computer program and are not protected by copyright. In this context, the keywords, syntax, commands, and other items in a computer program, considered in isolation, refer to the elements of a programming language, in which computer programs are normally written. By contrast, where the programmer "*by the choice, sequence and combination of those words, figures or mathematical concepts"* in his program succeeds in expressing his creativity in an original manner, this intellectual creation justifies the protection by copyright [10].

C. Program Expression for the Purposes of Copyright

Another important issue for copyright protection concerns the need for 'expression'. Thus, a characteristic feature of copyright is that the scope of protection is limited to the expression of a work, and not the underlying idea. (In contrast, "*a valid patent does not protect the expression of an idea but the underlying substance of it*" [11]; patent protection for cancer models is outside the scope of this paper).

As noted earlier, it has been established by EU and International copyright law, that the program object code and the source code constitute the object of protection of software by copyright. In this regard, the Software Directive in Article 1 (2) grants copyright protection to "*the expression in any form of a computer program*" [8]. What counts as a program expression for the purposes of the Directive has been established by the CJEU in its case law.

In particular, the CJEU considered this issue in the case *Bezpečnostní softwarová asociace – Svaz softwarové ochrany (BSA)* [12]. The main question raised was whether the graphic user interface (GUI) of a computer program counts as a form of a program expression within the meaning of Article 1(2) of the Directive and is thus protected. The court, in consideration of the international copyright law, and Article 10 TRIPS Agreement, held that the source code and the object code of a computer program constitute forms of expression entitled to be protected by copyright. Indeed, the court took the view that this applies to any expression of a program, which permits reproduction in different computer languages. As it stated, "the *object of the protection conferred by that directive is the expression in any form of a computer program which permits reproduction in different computer languages, such as the source code and the object code."* [12]. The source code of a program usually constitutes a script, written in a human readable form. The source code, compiled into a binary executable, constitutes the object code, which gives the final instructions to the computer [13].

What can be drawn from this decision is that only such expression of a program, which permits reproduction or re-creation of a program into other computer languages, counts as a program expression for the purposes of software copyright in the EU.

Usually, such reproduction or translation of a program into other programming languages is possible from the source code. The source code for a program is normally written in one or another programming language. The programming languages, which are mostly used in computer programming now, are Java, C^{++}, Python, etc. The choice of a language is mostly dictated by a system, upon which a program is intended to run. The reason is that some computers operate and can read only certain languages [13].

However, in contrast to the code itself, which because of its creative substance, is considered as a copyright work, the programming languages, which, in principle, "*comprise ideas and principles*", on which the programs operate and which do not expose creative substance, are not protected by copyright.

A programming language itself, be it C or Java, is normally composed of keywords and other symbols and may include a set of pre-written programs to carry out various operations, such as displaying something on the screen or retrieving the cosine of an angle [14]. According to the CJEU *SAS Institute* decision [10], the programming language, which comprises ideas and principles, both as isolated symbols, figures, keywords, mathematical concepts, etc., which constitute the material of that programming language, are not copyrightable.

Subsequently, in its decision *SAS Institute* [10], the CJEU considered the copyrightability of various other elements, which appear or are implemented by a computer program. The following elements were concerned:

"(a) the selection of statistical operations which have been implemented in the First Program;

(b) the mathematical formulae used in the Manual to describe those operations;

(c) the particular commands or combinations of commands by which those operations may be invoked;

(d) the options which the author of the First Program has provided in respect of various commands;

(e) the keywords and syntax recognised by the First Program;

(f) the defaults which the author of the First Program has chosen to implement in the event that a particular command or option is not specified by the user;

(g) the number of iterations which the First Program will perform in certain circumstances?" [10].

On the basis of the previous case law, and rules established in the Infopaq and BSA decisions, the court concluded that "*neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit certain of its functions constitute a form of expression of that program for the purposes of Article 1(2) of Directive 91/250.*" [10]. The CJEU supported its decision by the argument that "… *to accept that the functionality of a computer program can be protected by copyright would amount to making it possible to monopolise ideas, to the detriment of technological progress and industrial development.*" [10]. Such decision may also be interpreted the following way: that the elements, which express rather general principles than original creativity, belong to the domain of science and should not be monopolized, even if included into a copyright work. On the other hand, computer programs, which implement functions of a particular programming language, both as original structuring of such functions for being accessible via language application programming interfaces (API), may qualify as copyrightable expression and be protected as such.

Apart from the program source code and the object code, which are recognized as a literal expression of a program, there are also other hidden elements, which define perception of a program by a user. Such elements constitute a non-literal expression. And the question on protection of non-literal expression of a program by software copyright has been raised for consideration and, indeed, answered by the courts.

D. Non-Literal Program Expression

The courts in common law countries, such as the UK and US, tend to extend the scope of software copyright beyond the literal code to a program non-literal expression. The UK courts approach copyrightability of non-literal expression as follows: "*Consideration is not restricted to the text of the code… That must be right: most literal copyright works involve both literal matter (the exact words of a novel* *or computer program) and varying levels of abstraction (plot, more or less detailed of a novel, general structure of a computer program)."* [15].

The elements of a non-literal expression may include a program structure, sequence and organization, a program "*look and feel*", input and output routines [16]. It is often the case that structure of one program is imitated or reproduced by another program. This is a typical case when copyright protection for a program structure is sought. The legal issue raised before the courts here is: "w*hether the structure (or sequence and organization) of a computer program is protectable by copyright, or whether the protection of the copyright law extends only as far as the literal computer code."* [14].

The reason why protection of the program structure by copyright is important is that structuring a program often takes more time and intellectual effort than writing the code. And the UK copyright law gives much importance to protecting the skill and labor, which the authors deployed in their works, be it programmers or writers [16]. Let us consider why such protection is justified.

The code of a program itself consists of a set of instructions to the computer and is an end product of a complex software development process [17]. The latter process often occurs in several steps. First, the problem that needs to be solved by a computer is identified. In the next step, the outline for the solution follows. In the outline, the programmer breaks down the solution into smaller units called 'subroutines' or 'modules', each of which handles elements of a problem. The outline can be laid down in the form of a flowchart [13]. The next step is organizing the modules and subroutines into a program structure. A program structure is dictated by "*the functions of the modules in a program together with each module's relationships to other modules"* [14]. Usually, modules are arranged in such a way that a problem is solved in a more efficient way. As interpreted by the court in the case *Whelan Associates Inc. v. Jaslow Dental Laboratory* [13], "*Although two programs could produce the same result, one might be more efficient because of different internal arrangements of modules and subroutines. Because efficiency is a prime concern in computer programs (an efficient program being obviously more valuable than a comparatively inefficient one), the arrangement of modules and subroutines is a critical factor for any programmer.*" [13].

After defining the structure, a programmer decides about what data is needed, where and how the data should be introduced and how it should be combined with the other data. It is when the data is arranged into data files [13]. Once the program design is ready, the coding begins. The coding is a comparatively small part of programming. "*By far the larger portion of the expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program, and to debugging, documentation and maintenance, rather than to the coding."* [13].

Against this technical background, it becomes clear why the structuring of a program and arranging its modules in an efficient way takes a large amount of skill and work, is often copied and deserves protection on its own.

E. Idea-Expression Dichotomy

One of the tricky legal questions, which needs to be solved when copyright in non-literal expression of a program is sought, is separation of copyrightable expression from principles and ideas, which are non-copyrightable as such.

One of the general principles of copyright is that "*Copyright protection shall extend to expressions and not to ideas"* [5]. "P*rocedures, methods of operation*" are not copyrightable as such [5]. This principle applies to copyright in computer programs as well. The Software Directive says: "*For the avoidance of doubt, it has to be made clear that only the expression of a computer program is protected and that ideas and principles which underlie any element of a program, including those which underlie its interfaces, are not protected by copyright under this Directive. In accordance with this principle of copyright, to the extent that logic, algorithms and programming languages comprise ideas and principles, those ideas and principles are not protected under this Directive. In accordance with the legislation and case-law of the Member States and the international copyright conventions, the expression of those ideas and principles is to be protected by copyright."* [8].

Therefore, in order to grant copyright to non-literal program expression and, what is more important, to justify such decision, courts need to separate copyrightable expression from non-protectable ideas. Different criteria to address copyrightability of non-literal program expression have been elaborated.

F. Abstraction-Filtration-Comparison Test

One of the tests in this regard, which is widely applied both in the US [14] and in the UK [15], is the "*abstract-filtration-comparison*" test. The test has been established in the US case *Computer Associates International, Inc. v. Altai, Inc.* [18]. The idea behind the test is to reward programmers with copyright protection for creating "*innovative utilitarian works containing expressions*" and to leave non-protectable technical expressions in the public domain for further use [16].

The test comprises three steps. First, an original copyright program is broken down into its structural components according to the levels of abstraction. The second step extracts certain non-copyrightable structures (discussed below) until the copyrightable substance remains. In the third step, the portion of copying is compared with the copyrightable expression, left in the structure of the original program. Finally, it is estimated in how far such copying is substantial to justify infringement of copyright in a software program [18].

The three types of structures, identified as precluded from copyright, comprise: (a) elements dictated by efficiency, (b) elements required by external factors, (c) elements taken from the public domain.

1) Elements Dictated by Efficiency

First, copyright does not extend to structures dictated by efficiency (the doctrine of merger). Accordingly, copyright will not subsist in the expression, which is "*necessarily incidental to the idea being expressed*." In this step, it is determined "*whether the use of this particular set of modules is necessary efficiently to implement that part of the program's process" being implemented*." [18]. If efficiency dictates that the choice of modules is limited to just a few workable solutions (such as one or two options), such selection of modules may not be protectable as such.

2) Elements Dictated by External Factors

Secondly, copyright does not extend to structures dictated by external factors. External factors in software programming may include: compatibility requirements, mechanical specifications, computer manufacturer design standards, industry demands, and common programming practices. In US copyright law, this is known as the 'scenes a faire' doctrine [18]. In consequence, a particular set of modules, which need to be present as an integral part in all programs of the same category are non-copyrightable.

3) Elements Taken from the Public Domain

Thirdly, copyright does not protect structures that are found in the public domain. The rationale here is that such material, which is included as an element in a copyrightable work, may itself not be appropriated as it should remain free for use by the community [18].

G. Abstraction-Filtration-Comparison Test in Practice

The application of the test in practice may be illustrated by reference to the case Oracle America, Inc. v. Google Inc., C 10-03561 WHA, which dealt with copyrightability and copyright infringement in interfaces of the programming language Java. The central question related to the "*extent to which if at all, certain replicated elements of the structure, sequence and organization of the Java application programming interface are protected by copyright*" [14].

Java is a powerful object oriented programming language, developed by Sun Microsystems, first released in 1996, and acquired by Oracle in 2010. Java has a number of pre-written programs, called "methods", which invoke different functions, such as retrieving the cosine of an angle. These methods are grouped into "classes" and organised into "packages". Software developers get access to those classes through the Java APIs [19]. In 2008 Java APIs had 166 "packages", split into more than six hundred "classes", all divided into six thousand "methods".

Google built its Android platform for the smartphones using the Java language and, according to Oracle, "*utilized the same 37 sets of functionalities in the new Android system callable by the same names as used in Java*" [14]. By doing that, Google wrote its own implementations of the methods and classes, which it needed. The only one substantial element, which Google copied from Java into Android was the names and headers of 37 API packages in question. Such copying of the headers amounted to replication of the structure, sequence and organisation of Java APIs. Oracle claimed copyright infringement, and Google defended with fair use, arguing that Java is an open solution (which Oracle did not dispute) and there was no literal copying of the Java code.

The court of first instance trying the case qualified the headers and method names in Java APIs as non-copyrightable, referring to the interpretation criteria of the US Copyright Office: "*Even if a name, title, or short phrase is novel or distinctive or lends itself to a play on words, it cannot be protected by copyright*." [20]. This lends support to non-protectability of isolated code items by copyright, as similarly recognized by the CJEU [10].

As regards copying of the declarations and duplicating the command structure of Java APIs, the judge found that the command structure of Java APIs amounts to a method of operation – a material not subject to copyright in the US [20].

In Java programming, the specific declarations in the Java APIs designate a method. A method can be implemented in different ways, but is invoked by that specific declaration only. The command format, used to call the methods in Java, reads:

"java.package.Class.method()."

Here, a formula "*a = java.package.Class.method()"* sets the field "a", which is equal to the return of the method called. For example, the following call would call the method from Java:

"int a = java.lang.Math.max (2, 3)"

This command line would instruct the computer to fetch "*the max method under the Math class in the java.lang package, input "2" and "3" as arguments, and then return a "3," which would then be set as the value of "a."* [14].

As interpreted by the judge, in Java, each symbol in a command structure is more than a simple name - each symbol carries a task to invoke a pre-assigned function.

Considering that for using Java class methods software developers need to replicate the Java declarations, the judge qualified the command structure of Java APIs as a method of operation – a functional element essential for interoperability, not subject to the US Copyright Act. This position was based on the merger doctrine and non-copyrightability of structures dictated by efficiency: *"... When there is only one way to express an idea or function, then everyone is free to do so and no one can monopolize that expression.*" [14]. However, on appeal, the Federal Circuit Court reversed that ruling [21]. The appellate court found the declaring code and the structure, sequence and organization of packages in Java APIs were entitled to be protected by copyright.

The appellate court supported its decision by the argument that Java programmers were not limited in a way how they could arrange the 37 Java API packages at issue and had a choice to organize these API packages in other ways. For instance, instead of using the command format "*java.package.Class.method()"*: language – package – class – method, the same method could be called by the format: method – class – package – language. By making a decision to arrange the declarations in Java in this way and by having also other choices, the programmers were not prevented by the factor of efficiency, which would preclude copyright. By that, the programmers had a scope to exercise their creation, which they, in view of the court, exercised, indeed. This creation, realized in sequencing the Java APIs, amounted to a copyrightable expression. Against these considerations, the court concluded that, "*the structure, sequence, and organization of the 37 Java API packages at issue are entitled to copyright protection."* [21].

Google argued fair use and petitioned the US Supreme Court to hear the case. The US Supreme Court, referring to the opinion of the US Solicitor General, denied the petition. In the result, a new district court trial began. On 26 of May 2016 the district court jury found that Google´s Android does not infringe Oracle copyrights, because Google´s re-implementation of 37 Java APIs in question amounted and was protected by fair use. According to a Google spokesperson, *"Today's verdict that Android makes fair use of Java APIs represents a win for the Android ecosystem, for the Java programming community, and for software developers who rely on open and free programming languages to build innovative consumer products."* [22].

These cases deal with challenging questions of copyright law: free use of programming language APIs, copyrightability of APIs and an attempt "*to control APIs with copyright law"* and counter-balance between copyrights and "*fair use*" [22]. As established, the APIs, although elements necessary for interaction between computer programs, but which amount to intellectual creation, can be protected by copyright, at least in the opinion of one court of appeals. At the same time, as the jury found, the APIs, although protected by copyright, may be reused in other software systems, if such re-use is covered by fair use of open and free programming languages, like Java.

III. Cancer Models in terms of Copyright

In this section, we return to the cancer modeling. We explore the substance of cancer models and look into transformation of scientific models into computer models.

First of all, the two types of models need to be differentiated here: scientific models, which represent the biological processes, and computer models, which implement these biological processes *in silico*.

In the context of the CHIC project, scientific models are defined 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*." [23]. These scientific models correspond to the biological processes, being simulated. The scientific models are then implemented *in silico* via computer models. Computer models represent an executable form of scientific models, these being encoded into computer programs.

A computer model is defined as: "*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 a set of execution logs (control outputs)*." [24].

As noted above, computer programs constitute subject matter protectable by copyright [5]. Cancer models, written in computer programs, may also enjoy such protection, if they stand the requirements for protection. Such requirements would be fulfilled, if a model amounts to a copyrightable expression and is expressed in a form, which counts as a program expression for the purposes of software copyright.

According to the case law of the CJEU, a model would constitute a copyrightable expression, when the modeller "*through the choice, sequence and combination"* of commands in the model code succeeds to "*express his creativity in an original manner and achieve a result which is an intellectual creation"* [9]. On the other hand, if the code is generated automatically, for example by automatic translation into another programming language, or compiled from bits of code collected from the public domain, copyright protection in such code is most likely to be denied.

As regards the form of expression recognized by the software copyright, so it is the "*source or object code*", which constitute the object of copyright protection [5]. In relation to cancer models it means that models, which are exposed either in the source code or provided as binary executables shall be protected by copyright.

Models, as computer programs, are written in one or another programming language. In the context of the CHIC project, the models, embodied in source code, are mostly written for interpreted languages, such as Python, Perl or MATLAB; models provided in object code are usually binary C or C++ compiled executables [25]. Hence, it is the code, in which a model is embodied, be it the source code written in Python or the code compiled in C++, which is protected by copyright. The programming language itself, the biological process implemented by the model, the general process of its implementation, both as formats of data files, used by exchange of data between the models remain outside the scope of copyright subsisting in the model.

Once we established that computer models are protectable by copyright as computer programs and defined the scope of copyright protection, we can proceed to the next step, namely copyright issues in the hyper-modelling. At this stage it is necessary to state that copyright in the models, arranged into a hyper-model, remains by the models and does not extend to protecting the hyper-model as a copyright work on its own [5].

IV. Copyright in hyper-models

In this section, we continue the analysis on the scope of copyright in relation to cancer modeling. We proceed to a higher stage, namely coupling singe models into multiscale integrative hyper-models and explore the protection of hyper-models by copyright.

A. Substance of Hyper-Models

As mentioned at the beginning of this paper, multiscale cancer modelling is achieved by combining simple models into hyper-models.

In the context of the CHIC project, hyper-models comprise "*choreographies of component models, each one describing a biological process at a characteristic spatiotemporal scale, and of relation models/metamodels defining the relations across scales*." [3]. The aim thereby is that in a given instance (e.g., when populated with the data for a given patient) the hyper-model is able to reproduce biological processes of tumor progression, involving multiple phenomena, which are respectively captured in single models. A hyper-model "*emerges from the composition and orchestration of multiple hypomodels, each one of which is capable of simulating a specific entity or phenomenon… and can simulate an entity or phenomenon that may be more complex than the ones simulated by each separate simpler model*." [24].

The term hyper-model first appeared in 2008 in relation to Virtual Physiological Human (VPH) [26]. In 2011, the notion of a hyper-model was used in the context of computer science. Here, it was defined as "*a concrete instance of an integrative model, built as the orchestration of multiple computer models that might run on different computers at different locations using different simulation stacks*." [27]. The first implementation, based on web services, was tested on biochemical models [28].

An inter-play between single-scale models in a multiscale hyper-model is illustrated in Figure 1 [4]. Processes at lower levels occur faster, and processes at higher levels last longer. Above all it is the accurate correlation of these multiscale processes that makes multiscaling such a tricky task and requires the greatest intellectual effort. The scales are correlated to one another by the exchange of data where the output of one model serves as an input into another. The relevant interaction may be demonstrated by the example of a model of tumor growth where a cellular bio-model is coupled to biomechanical simulations [29].

Here the model of tumor growth is composed from a macroscopic mechanical model, which provides directions of least pressure in the tissue, and the cellular level model, which simulates concentration of cells inside the element. The microscopic cell-level model requires the direction of cells proliferation by tumor growth as an input. This information is provided by the mechanical model. In its turn, the biomechanical model requires the number of cells inside the element. This information is calculated by the biological cell-level model and fed to the biomechanical model. This model of tumor growth is structured via interplay and exchange of information between the models. The flowchart of the diffusion/mass effect coupling simulation used to simulate tumor growth is represented in Figure 2 [29].

The hyper-modelling execution process, such as the one just illustrated, is to a large extent facilitated and semi-automated by the underlying technical infrastructure. For example, in CHIC, the process of hyper-modelling is managed by VPH-Hypermodelling Framework [25].

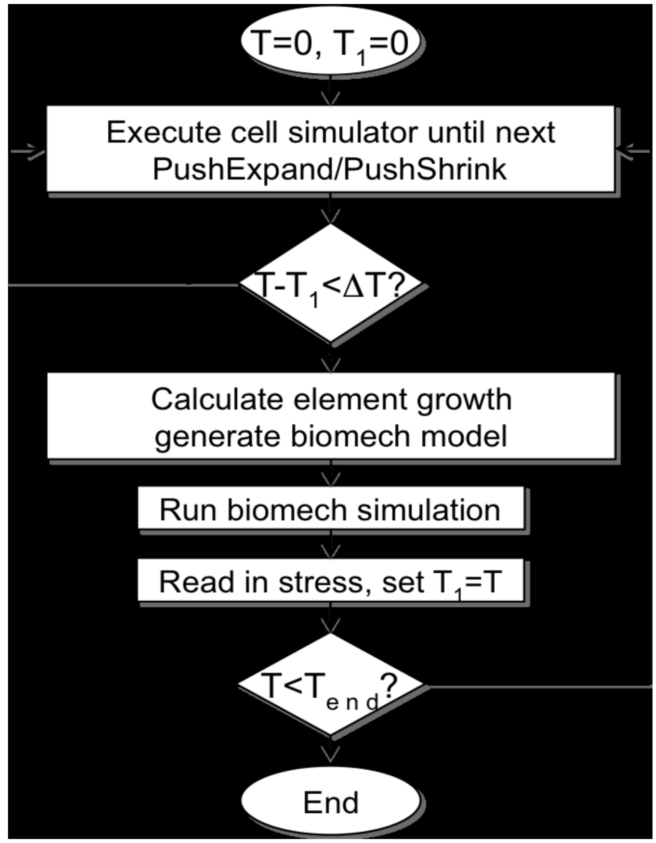

Figure 2. Flowchart of the diffusion/mass effect coupling simulation used to simulate tumor growth. Since the cellular effects occur at a different time scale than the mechanical reaction, the biomechanical calculations occur only after a larger number of cycles of the cell simulator biomodel.

However, at the earlier stage of hyper-model construction, human input into the hyper-model design is indispensable. Particularly where the intention is to use hyper-models for decision support in the clinical setting, the automatic linking of hypo-models is unjustifiably risky and thus not acceptable [30]. The hypermodelling strategy, the hypomodel linking, the hypomodel integration are instead normally designed by the modelling party, who has substantial expertise in the field of bioinformatics. Moreover, it is this intellectual input, deployed in designing a hyper-model, which may render a hyper-model copyrightable.

B. Hyper-Models in Terms of Copyright

The appropriate concept for qualifying hyper-models in terms of copyright is arguably that of a "compilation". Compilations, as works protectable by copyright, are introduced by Article 5 WIPO Copyright Treaty [6], Article 10 (2) TRIPS Agreement [5]. The TRIPS Agreement provides: "*Compilations, whether in machine readable or other form, which by reason of the selection or arrangement of their contents constitute intellectual creations shall be protected as such*." [5].

The crucial aspect in a compilation, which attracts copyright, is the intellectual creation represented by "*the selection or arrangement"* of its contents. In the case of

CHIC hyper-models, such intellectual creation may reside in the innovative "*composition and orchestration of multiple hypomodels",* thereby making a hyper-model a copyrightable expression.

C. Requirements for Protection

By applying the criteria of the *Infopaq* decision [9], which defined the meaning of the "*author´s own intellectual creation"* in the context of copyright, we can conclude that a hyper-model would constitute a copyrightable expression if the modeller "*through the choice, sequence and combination"* of models in a hyper-model *"may express his creativity in an original manner and achieve a result which is an intellectual creation."* [9].

In the CJEU case law, the cases on infringement of copyright in compilations are mostly tried in relation to database rights. The reason is that the Directive 96/9/EC on the legal protection of database (Database Directive) [31] also uses the term "compilation", referring to collections of works, data or other materials. In this case, copyright is a reward for the intellectual effort, which the author deployed in the selection or arrangement of the database contents [31].

The criteria for measuring originality in a database structure have been established by the CJEU in the case *Football Dataco Ltd. and Others v Yahoo! UK Ltd. and Others* [32]. Accordingly, the originality is measured by the creativity, which the database maker expressed in making free and creative choices, thereby stamping his work with a personal touch. On the other hand, if the technical constraints leave no space for creativity, the element of originality in structuring a database will be lacking and copyright protection unjustified [32].

If we compare the criteria for measuring creativity in a program structure, considered above, with the criteria of originality in the compilation, we can see that these criteria inter-relate with each other. Thus, if the author (either a programmer or a compiler of a database) had some scope to express his creativity in arranging the modules in a program (or contents into a database) by making free and creative choices, such a structure may show originality and qualify as an intellectual expression in terms of copyright. On the other hand, if creativity may not be realized because of some technical considerations, then further analysis is needed to decide if and how far copyright in the structure may still arise. In particular, the abstraction-filtration-comparison test and criteria, defined by the courts, when they dealt with protection of a program structure by copyright, may be a helpful pointer.

D. Abstraction-Filtration-Comparison Test in Relation to Hyper-Models

By way of providing a concrete illustration, let us return to the hyper-model of tumor growth that we described above. This hyper-model consists of two component models: a microscopic cellular bio-model, which shows the concentration of cells in the invaded tissue, and a macroscopic biomechanical model, which provides directions of least pressure in the tissue. The overall function of the hyper-model is to simulate the "*geometrical evolution of the tumor predicted at the cellular level*." [29].

First, according to the test, we dissect a hyper-model into its structural components, i.e., component models. The hyper-model structure is dictated by the two component models and a relationship, in which they stand to each other.

1) Elements Dictated by Efficiency

In this step we look whether the choice and combination of component models is "*necessary efficiently to implement*" the function of this hyper-model [18].

If a bio-model of cell concentration and biomechanical model of tissue are the only models and the combination of these models by exchange of output/input data is the only way, in which the geometrical evolution of the tumor predicted at the cellular level can be simulated, then it may mean that this hyper-model is designed in this way for reasons of efficiency. In such a case, the modeler´s choice of component models and their combination has merged with the underlying idea and according to the merger doctrine may not be appropriated by copyright.

Otherwise, where the "*evolution of the tumor predicted at the cellular level*" may be represented by the selection of other models and/or the combination of these or other models in another way, then the hyper-model structure may qualify as copyrightable expression.

2) Elements Dictated by External Factors

According to the "*scenes a faire*" doctrine, copyright does not extend to structures dictated by external factors. In the context of programming, such external factors may be the compatibility requirements, computational constraints, demands of the industry being served, widely accepted programming practices, etc.

For example, elements or methods, which are "indispensable or at least standard" in the computer modelling community, would be non-copyrightable. Such elements may include: computational demands (especially for models, which work on higher spatial and temporal resolutions), or requirements of Digital Model Repository (an innovative platform for exchange of cancer models), or criteria of Heterogenous Multiscale Method (HMM) etc. [2].

3) Elements Taken from the Public Domain

The other group of elements, which may not be copyrighted, are elements taken from the public domain. Copyright may not be claimed in "*an expression which, if not standard, then commonplace in the computer software industry."* [18].

For example, if the component models, selected for a hyper-model in question, and/or the method of combining them have been borrowed from the public domain, the same models and/or a method of coupling them should remain outside the scope of copyright, residing in the hyper-model.

Admittedly, the above analysis is somewhat schematic, and rather hypothetical. Its main purpose is to outline some general guidance and describe some general cases when certain elements and structures may be protectable by copyright or not.

V. Copyright in Preparatory Design Material

Another major element in cancer modelling, which also consumes much of intellectual effort and may well deserve copyright protection, is the scientific modelling. In fact, the modelling work, which precedes the development of computer codes for the cancer models, may qualify as a preparatory design material to a program and fall under the scope of software copyright [8].

Scientific modelling comprises the works, performed in the course of transforming scientific models, which represent biological processes, into computer models, which are encoded into computer programs. Such modelling work comprises various stages. These include identification of elementary processes for simulation (e.g., cell cycling, the angiogenesis process, declination of a cell to apoptosis after a particular treatment, etc.), the definition of modelling techniques - discrete, continuum, or hybrid, and development of computer codes for the cancer models, which would simulate those biological processes *in silico* [2]. These modelling steps follow in many aspects the stages of software development. First, the problem to be solved by a computer is analyzed, then methods of solving the problem are adopted and stages of running the program are identified. Subsequently, detailed instructions for a computer to perform operations necessary for the execution of those stages are developed [17].

As noted earlier, both as a computer program stands at the end of a long development process [17], the model code, embodied in the computer program, results from the foregoing modelling work. A code of a program may be compared with the top of the iceberg, which reaches out above the water, where 90% percent is hidden under the surface. And the same may be said in relation to the earlier scientific modelling (which precedes the computer modelling).

In view of the skill, time and labour, which a programmer invests in studying a problem, elaborating a solution for it and making the solution executable by a computer, a justifiable legal question arises whether this pre-programming work is protectable by software copyright. In fact, in contrast to some of the other issues considered in this paper, here the Software Directive gives a reasonably clear affirmative answer. It does so by extending the scope of software copyright to the preparatory design work that preceded the creation of a computer program.

In particular, Article 1 (1) of the Directive, when defining the object of protection, provides: "*the term 'computer programs' shall include their preparatory design material*." [8]. What is meant by the "*preparatory design material"* is explained in Recital 7. A preparatory design work must be of a kind "*such that a computer program can result from it at a later stage."* [8]. As further interpreted by the CJEU, the preparatory design work counts as protectable by the directive along with a program, if it is "*capable of leading, respectively, to the reproduction or the subsequent creation of such a program"* [12].

In contrast to the form of program expression, protectable by copyright, there are no specific requirements to the form or mode, in which the preparatory design work must be expressed. According to one decision of the Federal Court of Justice of Germany, which dealt with protection of preparatory design work by copyright, it will be sufficient that the development documentation was recorded in writing, such as: data flow plans, designs of commands and information cycles, exhibits of scientific or technical art, expressed in any form, including mathematical, or technical or graphic symbols [33].

It follows, in relation to copyright in the kind of work occurring in CHIC, that modelling work (a) documented in writing, such as laid down in flow charts, exhibits, etc., and (b) attributable to a specific model, which if necessary may be rewritten from these materials, has a good chance of being protected by software copyright along with the model code.

VI. Conclusions

From the above observations it follows that various elements in cancer modelling, which express original creative input, may be protectable by copyright. If protection is not achievable by software copyright, then protection under the ordinary law of copyright may be another option.

First, the models may enjoy copyright protection as computer programs. Second, copyright in the model may extend to the modelling work, underlying the model code. Third, hyper-models, structured in an innovative and original way, may count as compilations in terms of copyright and be protected as such.

Dealing with the computer models first. Where the models are encoded into computer programs and exposed either in source code or as compiled executables, they shall be protectable by software copyright. Apart from the code, also the modelling documentation can be copyrightable along with the model code, provided it records the steps, conducted in the course of developing a model and may be used as a basis for reproducing the model concerned.

Whereas, the model code, the outline of a model, the flowcharts and exhibits, produced by the modelling, are protectable by copyright, the programming language, in which models are written, the processes and functionality, which cancer models implement, the data files and algorithms, utilized by the models in performing their functions, may not be covered by copyright subsisting in the model.

Hyper-models, which are composed of single models, arranged in specific relations to each other, may also be protectable by copyright as compilations. For this, a hyper-model must show an original and creative structure. Copyrightability of a hyper-model as a compilation is to a large extent dictated and can be measured by the degree of originality and creativity, deployed in a hyper-model structure. Thus, in cases where a hyper-model is structured in a particular way because there have been no other ways of arranging models into it (such as when the model arrangement is pre-determined by technical requirements), it will not attract copyright. The same goes for cases where the structure relies on elements taken from the public domain. In contrast, if the modeler was not so restricted and expressed creativity stamping a hyper-model structure "*with his personal touch*", such investment shall not remain unrewarded and copyright in his work will likely be recognized.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement No 600841.

REFERENCES

[1] I. Lishchuk and M. Stauch, "Copyright in Multiscale Cancer Modelling," The Fifth International Conference on Advanced Communications and Computation (INFOCOMP 2015) IARIA, June 2015, pp. 117-122, ISSN: 2308-3484, ISBN: 978-1-61208-416-9.

[2] T. Deisboeck, Z. Wang, P. Macklin, and V. Cristini, "Multiscale Cancer Modelling," Annual Review of Biomedical Engineering, vol. 13, pp. 127-155, August 2011, doi: 10.1146/annurev-bioeng-071910-124729.

[3] "Computational Horizons In Cancer (CHIC): Developing Meta- and Hyper-Multiscale Models and Repositories for In Silico Oncology", CHIC, Project, Objectives, <http://chic-vph.eu/project/objectives> 2016.05.30.

[4] J. A. Grogan, P. K. Maini, J. Pitt-Francis, and H. M. Byrne, "Simulating Tumour Vasculature at Multiple Scales," In G. Stamatakos and D. Dionysiou (Eds): Proc. 2014 6th Int. Adv. Res. Workshop on In Silico Oncology and Cancer Investigation – The CHIC Project Workshop (IARWISOCI), Athens, Greece, Nov.3-4, 2014 (www.6thiarwisoci.iccs.ntua.gr), pp. 23-26, (open-access version), ISBN: 978-618-80348-1-5.

[5] Agreement on Trade-Related Aspects of Intellectual Property Rights, the TRIPS Agreement, Annex 1C of the Marrakesh Agreement Establishing the World Trade Organization, Marrakesh, Morocco, 15 April 1994.

[6] WIPO Copyright Treaty (WCT), Geneva, 20 December 1996.

[7] Berne Convention for the Protection of Literary and Artistic Works of September 9, 1886.

[8] Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs, Official Journal of the European Union (OJEU), L 111/16 – 111/22, 5 May 2009.

[9] CJEU, Judgment of 16 July 2009, Case C 5/08, Infopaq International A/S v Danske Dagblades Forening, ECLI:EU:C:2009:465.

[10] CJEU, Judgment of 2 May 2012, Case C 406/10, SAS Institute Inc v World Programming Ltd, ECLI:EU:C:2012:259.

[11] A. M. St. Laurent, "Understanding open source & free software licensing," O´Reilly, 1. Edition, 2004, p. 2.

[12] CJEU, Judgment of 22 December 2010, Case C 393/09, Bezpečnostní softwarová asociace (BSA) – Svaz softwarové ochrany v Ministerstvo kultury, ECLI:EU:C:2010:816.

[13] U.S. Court of Appeals, Third Circuit, Ruling of August 4, 1986, Whelan Associates Inc. v. Jaslow Dental Laboratory, Inc., et al, 797 F.2d 1222, 230 USPQ 481.

[14] U.S. District Court for the Northern District of California, Ruling of 31 May 2012, Case C 10-03561 WHA, Oracle America, Inc., v. Google Inc.

[15] Ibcos Computers Ltd v Barclays Mercantile Highland Finance Ltd [1994] FSR 275.

[16] S. Stokes, "Digital Copyright, Law and Practice," 4. Edition, Hart Publishing, Oxford and Portland, Oregon, 2014, p. 121.

[17] Report by the International Bureau, "Measures to enhance International Cooperation in the field of Legal Protection of Computer Software," WIPO expert group on the legal protection of computer software, First Session, Geneva, 27-30 November 1979, LPCS/I/2, 30 September 1979.

[18] U.S. Court of Appeals for the Second Circuit, Ruling of 17 December 1992, Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 2d Cir.1992.

[19] Oracle, Java Platform, Standard Edition 6, API Specification, Overview, <http://docs.oracle.com/javase/6/docs/api/> 2016.05.30.

[20] U.S. Copyright Office, Circular 34; "Copyright Protection 'Not Available for Names, Titles or Short Phrases," rev. January 2012.

[21] U.S. Court of Appeals for the Federal Circuit, Ruling of 09 May 2014, Oracle America, Inc., v. Google Inc., Appeals from the United States District Court for the Northern District of California in No. 10-CV-3561.

[22] J. Mullin, "Google beats Oracle—Android makes "fair use" of Java APIs," Ars Technica, 26 May 2016, <http://arstechnica.com/tech-policy/2016/05/google-wins-trial-against-oracle-as-jury-finds-android-is-fair-use/> 2016.05.31.

[23] M. Viceconti, "A tentative taxonomy for predictive models in relation to their falsifiability," Philos. Transact A Math Phys Eng Sci, vol. 369, pp. 4149-61, 1954.

[24] CHIC, Deliverable No. 7.1, "Hypermodelling specifications," CHIC, Downloads, Deliverables, Public Deliverables, <http://chic-vph.eu/uploads/media/D7-1_Hypermodelling_specifications.pdf> 2016.05.30.

[25] D. Tartarini, K. Duan, N. Gruel, D. Testi, D. Walker, and M. Viceconti, "The VPH Hypermodelling Framework for Cancer Multiscale Models in the Clinical Practice," In G. Stamatakos and D. Dionysiou (Eds): Proc. 2014 6th Int. Adv. Res. Workshop on In Silico Oncology and Cancer Investigation – The CHIC Project Workshop (IARWISOCI), Athens, Greece, Nov.3-4, 2014 (www.6thiarwisoci.iccs.ntua.gr), pp. 61-64, (open-access version), ISBN: 978-618-80348-1-5.

[26] M. Viceconti, "A tentative taxonomy for predictive models in relation to their falsifiability," Philos. Transact A Math Phys Eng Sci, vol. 369, pp. 4149-61, 1954.

[27] P. A. Fishwick, "Hypermodelling: an intergrated approach to dynamic system modelling," Journal of simulation, vol. 6, pp. 2-8, February 2012.

[28] R. Guha, "Flexible Web Service Infrastructure for the Development and Deployment of Predictive Models," Journal of Chemical Information and Modellling, vol. 48 (2), pp. 456-464, 2008.

[29] F. Rikhtegar, E. Kolokotroni, G. Stamatakos, and P. Büchler, "A Model of Tumor Growth Coupling a Cellular Biomodel with Biomechanical Simulations," In G. Stamatakos and D. Dionysiou (Eds): Proc. 2014 6th Int. Adv. Res. Workshop on In Silico Oncology and Cancer Investigation – The CHIC Project Workshop (IARWISOCI), Athens, Greece, Nov.3-4, 2014 (www.6thiarwisoci.iccs.ntua.gr), pp.43-46 (open-access version), ISBN: 978-618-80348-1-5.

[30] G. Stamatakos, et al., "The Technologically Integrated Oncosimulator: Combining Multiscale Cancer Modelling with Information Technology in the In Silico Oncology Context," IEEE J Biomedical and Health Informatics, vol. 18, no. 3, pp. 840-854, May 2014.DOI:10.1109/JBHI.2013.2284276.

[31] Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, OJEU, Nr. L 77/20-77/28, 27 March 1996.

[32] CJEU, Judgment of 1 March 2012, Case C 604/10 Football Dataco Ltd. and Others v Yahoo! UK Ltd. and Others, ECLI:EU:C:2012:115.

[33] Federal Court of Justice of Germany, Judgment of 09 May 1985, Case I ZR 52/83, BGHZ 94, pp. 276 – 292.