Univerzita Karlova v Praze
1.lékařská fakulta
Charles University in Prague
1st Faculty of Medicine

Studijní obor: Biomedicínská informatika
Study domain *Biomedical informatics*

Mgr. Tomáš Kulhánek

Využití technologie GRID při zpracování medicínské informace

Utilization of GRID technology in processing of medical information

Dizertační práce
Dissertation

Školitel: Ing. Milan Šárek, CSc.
Supervisor

Konzultant: Doc. MUDr. Jiří Kofránek, CSc.
Consultant

Praha 2015

Prohlášení

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem řádně uvedl a citoval všechny použité prameny a literaturu. Současně prohlašuji, že práce nebyla využita k získání jiného nebo stejného titulu.

Souhlasím s trvalým uložením elektronické verze mé práce v databázi systému meziuniverzitního projektu Theses.cz za účelem soustavné kontroly podobnosti kvalifikačních prací.

V Praze, 2015 Mgr. Tomáš Kulhánek

Podpis

Identifikační záznam

KULHÁNEK, Tomáš. *Využití technologie GRID při zpracování medicínské informace*. Dizertační práce. Praha, 2015. 48 stran. 7 příloh. Univerzita Karlova v Praze, 1. lékařská fakulta, Ústav patologické fyziologie. Školitel Šárek, Milan. Konzultant Kofránek, Jiří.

Identification record

KULHÁNEK, Tomáš. *Utilization of GRID technology in processing of medical information*. Dissertation. Prague, 2015. 48 pages. 7 appendices. Charles University in Prague, 1st Faculty of Medicine, Institute of Pathological Physiology. Supervisor Šárek, Milan. Consultant Kofránek, Jiří.

Dedication

To my beloved children Karla and Matěj and my dearest lovely wife Marie,
without whom this work would not have been completed.

Acknowledgements

I would like to express my gratefulness to my supervisor Ing. Milan Šárek, CSc. for his support and guidance throughout the beginning of my research. I would also like to thank the staff of association CESNET, especially Ing. Jiří Navrátil, CSc., who gave me an opportunity to work with computing resources connected via high speed academic network and to follow the development of the networking technologies. Further thanks go to the staff of METACENTRUM, department of CESNET, for giving support and access to powerful computing infrastructure for scientific purposes. I would like to thank to the staff of EGI.eu, who gave me an opportunity to be EGI Champion, selected users of scientific European grid-computing and cloud-computing infrastructure. This made me possible to meet the most inspiring people in the technology domain as well as scientific domain face to face.

I would like to thank the fellows from Musical Acoustic Research Center of Academy of Performing Arts, especially RNDr. Marek Frič, PhD. and Ing. Zdeněk Otčenášek, Phd. for their inspiring atmosphere in the research domain of voice science.

I would like to thank my colleagues from Laboratory of Biocybernetics and Computer Aided Teaching within Institute of Pathological Physiology including Mgr. Marek Mateják, Ing. Jan Šilar, Ing. Filip Ježek, Ing. Martin Tribula and MUDr. et Mgr. Pavol Privitzer, who provided me pleasure and inspiring atmosphere for my research and development work during the second part of my research. Especially, I would like to thank to my consultant Doc. MUDr. Jiří Kofránek, CSc. and a fellow MUDr. Stanislav Matoušek, PhD., who taught me the subject of medical physiology and who guided me within the topic of mathematical models and simulation of integrative physiology. Last but not least, thanks to Veronika Sýkorová, DiS and Klára Ulčová, DiS for their help with graphical schemas.

The research projects were solved within the cooperation between association CESNET, First Faculty of Medicine of Charles University in Prague and Academy of Performing Arts in Prague and was partially supported by the project of Ministry of Education, Youth and Sport of the Czech Republic MSM6383917201, by the projects of Development Fund of CESNET 2009/361, 2010/384, 2011/423, 2011/431 and by the project of Ministry of Industry and Trade of the Czech Republic MPO FR-TI3/869.

Abstrakt (česky)

Název práce: Využití technologie GRID při zpracování medicínské informace

Autor: Mgr. Tomáš Kulhánek

Katedra,ústav: Ústav patologické fyziologie 1.LFUK

Školitel: Ing. Milan Šárek, CSc., CESNET z.s.p.o.

Konzultant: Doc. MUDr. Jiří Kofránek, CSc.

Práce se soustředí na vybrané oblasti biomedicínského výzkumu, které mohou profitovat ze současných výpočetních infrastruktur vybudovaných ve vědecké komunitě v evropském a světovém prostoru. Teorie výpočtu, paralelismu a distribuovaného počítání je stručně uvedena s ohledem na počítání v gridech a cloudech. Byla studována oblast výměny medicínských snímků a Gridový PACS systém byl propojen s existujícími distribuovanými systémy pro sdílení DICOM snímků. Další studovanou doménou byla věda týkající se lidského hlasu. Vzdálený přístup k aplikaci pro analýzu hlasu v reálném čase byl představen zároveň s úpravou protokolů pro vzdálenou plochu pro přenos zvukových nahrávek. To přináší možnost využití stávajících aplikací na dálku specialisty na hlas.

Byl studován přístup tzv. systémové biologie v oblasti lidské fyziologie a patofyziologie. Bylo přispěno k metodologii modelování lidské fyziologie pro tvorbu komplexních modelů založených na akauzálním a objektově orientovaném modelovacím přístupu. Byly představeny metody pro studium parametrů pomocí technologie počítání v gridech a v cloudech. Proces identifikace parametrů středně komplexních modelů kardiovasculárního systému a komplexního modelu lidské fyziologie lze významně zrychlit při použití cloud computingu a dobrých výsledků lze dosáhnout v rozumném čase. Tato metoda umožňuje aplikovat parametrické studie ve fyziologickém a biologickém výzkumu. Toto může zlepšit praktické použití matematických modelů a identifikaci parametrů ve zdravotní péči.

Klíčová slova: gridové počítání, počítání v cloudu, výpočetní fyziologie, systémová biologie, odhad parametrů, výměna medicínských snímků, analýza hlasového signálu

Abstract

Title: Utilization of GRID technology in processing medical information

Author: Mgr. Tomáš Kulhánek

Department: Institute of pathological Physiology

Supervisor: Ing. Milan Šárek, CSc., CESNET z.s.p.o

Consultant: Doc. MUDr. Jiří Kofránek, CSc.

This thesis focuses on selected areas of biomedical research in order to benefit from current computational infrastructures established in scientific community in european and global area. The theory of computation, parallelism and distributed computing, with focus on grid computing and cloud computing, is briefly introduced. Exchange of medical images was studied and a seamless integration of grid-based PACS system was established with the current distributed system in order to share DICOM medical images. Voice science was studied and access to real-time voice analysis application via remote desktop technology was introduced using customized protocol to transfer sound recording. This brings a possibility to access current legacy application remotely by voice specialists.

The systems biology approach within domain of human physiology and pathophysiology was studied. Modeling methodology of human physiology was improved in order to build complex models based on acausal and object-oriented modeling techniques. Methods for conducting a parameter study (especially parameter estimation and parameter sweep) were introduced using grid computing and cloud computing technology. The identification of parameters gain substantial speedup by utilizing cloud computing deployment when performed on medium complex models of cardiovascular system and complex models of human physiology. This makes such kind of study applicable in order to perform identification of physiological system in reasonable time for physiological and biological research and good results are available in a reasonable time. This can improve practical usage of mathematical models in healthcare.

Keywords: grid computing, cloud computing, computational physiology, systems biology, parameter estimation, medical image exchange, voice signal analysis

Contents

Chapter 1

Introduction

Grid computing is usually defined as sharing computational and data storage resources across organizational boundaries. In recent years, the development of virtualization technologies has enhanced the availability of services that are provided by grid computing. It has additionally enabled an evolution of the so-called *cloud computing*, which also utilizes a virtual environment on powerful computing infrastructures. Based on the development of technologies and the philosophy of providing them to end users, this thesis focuses on the multidisciplinary research related to grid computing, as well as to cloud computing. It discusses its utilization in biomedical research and its application in relation to the processing of medical information.

The term "medical information" is too broad and further work in this thesis focuses on the following selected areas: (1) the exchange and processing of medical images, (2) the analysis of human voice and (3) the modeling and simulation of human physiology.

The author's work was published in a series of peer-reviewed papers of international journals and peer-reviewed conference proceedings [1, 2, 3, 4, 5, 6, 7] which are included in this work as appendices. The author's work and contribution was also presented in international conferences and published in the respective proceedings and transactions [8, 9, 10, 11]. The work was also popularized in local and regional conferences and their respective proceedings [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. The author contributed to the utility model, which was registered by the Czech Industrial Property Office [24].

1.1 Thesis Goal

The hypothesis of this thesis is that the technologies that relate to grid computing and cloud computing may improve the processing of medical information in order to perform demanding tasks that are almost impossible or require onerous effort to achieve, using classical local or institutional resources. The particular goals of this thesis were:

- To study the latest achievements in the field of exchanging medical images and possible improvements using the grid computing and cloud computing technology.

- To identify use cases in other fields of biomedicine which are suitable to utilizing the power of grid computing and cloud computing infrastructure.

- To develop and test the prototype application that utilizes grid or cloud technologies.

This thesis tries to discuss the hypothesis in different areas of biomedical research and its application. It tries to find answers to the following additional questions:

- *Is it beneficial to utilize grid computing and cloud computing technology for the processing of medical information and how do we do this?* When work on this thesis begun, grid computing was believed to be an answer to scalability issues, e.g., for exchanging large amounts of data or carrying out demanding long-term computation.

- *What are the limitations of processing medical information in grid or cloud?*

- *How can the grid computing and cloud computing influence the direction of biomedical research?* There was an idea that grid computing technology inspires the current architecture of distributed systems, e.g., exchanging medical images (explained in section 3.1) and influences the direction of information systems in hospitals.

Answers to these questions are summarized in section 5.1.

1.2 Thesis Contribution

The author claims that the following contribution was made to state-of-the-art biomedical informatics and computational biology.

- Proposal of a grid infrastructure and pilot implementation of a grid-based system that exchanges medical images that are integrated with an existing distributed systems. The results were published by Kulhánek and Šárek in [1] and popularized in [12, 13, 14]. The author of this thesis customized the existing project, Globus MEDICUS, and deployed it in servers that are networked via the academic network, CESNET, and integrated with the existing regional PACS system. Other co-author coordinated the work with selected hospitals and operators of the PACS system.

- Pilot implementation of a more generic infrastructure as a service for the community within the biomedical research published by Kulhánek in [3, 19]. The author of this thesis proposed the idea of consolidating and sharing physical resources in order to provide a virtual environment for the specific needs of particular use-cases. The pilot infrastructure was tested on examples of selected research projects.

- Proposal of a system and implementation of a remote service for the real-time analysis of a human voice. The results were published by Kulhánek et al. in [2] and popularized in [16, 20]. The author of this thesis designed and customized the existing network protocol in order to transfer voice signal losslessly. The author also deployed the application on a remote virtual server. Other co-authors implemented the algorithms and application in order to analyze voice signal.

- Improved methodology for the modeling of complex physiological systems published by Kulhánek, Mateják, Kofránek et al. in [5, 6, 10, 11]. The author of this thesis contributed to the idea of building complex mathematical models from the basic components and keeping them in an understandable and maintainable form. Additionally, the author implemented several basic blocks and models of a pulsatile cardiovascular system in Modelica language. The other co-authors implemented the library in order to model physiology, using an integrative approach. They also implemented the complex models, which integrated different domains together.

- Design and implementation of a system to estimate the parameters of complex mathematical models in order to validate and calibrate models of the human physiology were published by Kulhánek et al. in [4] and gradual development of related technologies were published and popularized in [8, 9, 18, 22]. The author of this thesis designed architecture for a distributed parameter estimation, integrated models and implemented a pilot deployment, which utilized a scientific cloud computing infrastructure. Other co-authors implemented complex models of the human physiology in Modelica language and tested several algorithms for parameter estimation.

- Improved mathematical model of oxygen, carbon dioxide and hydrogen ion binding to hemoglobin published by Mateják et al. in [7]. The author of this thesis implemented this model in Modelica language and identified its parameters. Other co-authors analyzed and proposed a new mathematical model, based on the basic physical and chemical laws and in relation to previously published studies.

- Simulation of complex models of the human physiology as part of a virtual simulator on portable and mobile devices, utilizing cloud computing published by Kulhánek et al. in [9, 21]. The author of this thesis contributed to the idea of a hybrid architecture of web simulators. This utilizes the infrastructure for parameter estimation in order to simulate complex models remotely and process/visualize the results locally. Other co-authors implemented complex models of the human physiology and simulation scenarios for educational purposes.

- Virtual patient simulator prototype was registered as a utility model by the Industrial Property Office in the Czech Republic [24]. The author of this thesis designed and developed a specific module to control the multiple instances of a virtual simulator within a virtual classroom via a web server application. Other co-authors designed and implemented models of the human physiology and clinically relevant educational scenarios. They also implemented a 3D visualization of selected scenarios using game engine Unity 3D[1].

[1] http://unity3d.com/ accessed March 2015

1.3 Thesis Structure

This thesis is interdisciplinary and, therefore, the following chapters cover the topics from a technical and computer-science point of view. They also touch on some topics that are related to medical science. Chapter 2 provides an overview of the state-of-the-art theory of computation, parallel computation and distributed computing especially grid computing and cloud computing.

Introduction to selected areas of biomedical research domains and related particular methods are in chapter 3. Sharing medical images is introduced in section 3.1, voice science is introduced in section 3.2 and computational biology is introduced in section 3.3.

Chapter 4 summarizes the general results that were obtained by the research methods in specific areas of biomedical research and applications. Chapter 5 discusses the achievements and answers of the hypothesis, as well as the questions thate were stated at the beginning of the work. It also recommends further areas for research.

The appendices contain selected papers [1, 2, 3, 4, 5, 6, 7] that are most relevant to the topic of this thesis. These were published in international peer-reviewed journals or in peer-reviewed conference proceedings:

Appendix A is the paper [1] *Processing of Medical Images in Virtual Distributed Environment*, published by ACM as part of the proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to Increase Digital Citizenship.

Appendix B is the paper [2] *Remote Analysis of Human Voice – Lossless Sound Recording Redirection*, published in Analysis of Biomedical Signals and Images, Proceedings of 20th International EURASIP Conference (BIOSIGNAL).

Appendix C is the paper [3] *Infrastructure for Data Storage and Computation in Biomedical Research*, published by Euromise s.r.o. in the European Journal of Biomedical Informatics.

Appendix D is the paper [4] *Parameter Estimation of Complex Mathematical Models of Human Physiology Using Remote Simulation Distributed in Scientific Cloud*, published in the IEEE Xplore Digital Library as part of the proceedings of the 2014 IEEE-EMBS International Conference on Biomedical and Health Informatics.

Appendix E is the paper [5] *Modeling of Short-term Mechanism of Arterial Pressure Control in the Cardiovascular System: Object-oriented and Acausal Approach*, published by ELSEVIER in Computers in Biology and Medicine 2014, **IF(2013): 1.475**.

Appendix F is the paper [6] *Simple Models of the Cardiovascular System for Educational and Research Purposes* published in Mefanet Journal 2014.

Appendix G is the paper [7] *Adair-based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity*, published in Scandinavian Journal of Clinical and Laboratory Investigation 2014, **IF(2013): 2.009**.

Chapter 2

State-of-the-art

The processing of medical information deals with methods that connect different scientific domains, computer science, biomedical engineering and medicine, together with a common goal.

From a computer science (informatics) point of view, it is assumed that the processing of medical information is, in general, a computational problem, which is understood as a task that can be solved by a computer.

As some computationally hard problems ar discussed later in this thesis, the next sections briefly introduce the theoretical and practical aspects, as well as the consequences, of theory of computation, parallelism and distributed computing. Section 2.1 introduces some of the important complexity classes of problems from the view of the computational complexity theory.

Parallel computation can introduce speedup of computation when specific conditions are fulfilled. This theory is briefly covered in section 2.2.

The distribution of a parallel task via a computer network to other computers, servers and cluster of servers is covered in section 2.3, with a focus on grid computing and cloud computing.

2.1 Computational Complexity

An algorithm is a set of operations that is used to accomplish tasks and solve problems. There are several ways of expressing algorithms, e.g., in text, in programming language, pseudo-code or flowcharts. Later in this thesis kopenograms will be used as a graphical language for structured algorithms in order to supplement the Unified Modeling Language (UML) diagrams, as proposed by Kofranek et al.[25][1].

The computational complexity theory classifies problems into several classes, according to the time or space needed by the algorithm in order to solve the problem. The time complexity of an algorithm is usually denoted by a big O notation and size of input problem n

[1] http://www.kopenogram.org accessed March 2015

meaning that a time complexity denoted by $O(g(n))$ is not growing faster than the function g. Formally $f(n) = O(g(n))$, if, and only if, there is a constant c and positive integer n_0 that for each $n \geqslant n_0$: $f(n) \leqslant c \times g(n)$.

$O(1)$ denotes algorithms that take constant time, regardless of the input size. $O(n)$ denotes linear time algorithms, i. e., time is linearly dependent of the input size. For example, figure 2.1 shows the sequential search algorithm in a pseudo-code and kopenogram, which needs to compare each record with a given key. This is used to find some items in an unsorted list or array. For example, if a single comparison takes 0.03 seconds and the list has n records, then the algorithm will take, at worst, n steps and the time complexity will be $f(n) = 0.03 \cdot n = O(n)$. [2]

```
int function SequentialSearch(Array Records, int Key)
{
        for (int index=0;i<Records.Length;i++)
        {
                if (Records[index]=Key)
                {
                      return (index)
                }

        }

        return(not found)
}
```

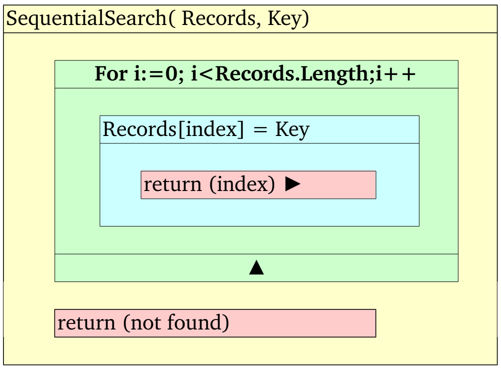

SequentialSearch(Records, Key)
For i:=0; i<Records.Length;i++
Records[index] = Key
return (index) ▶
▲
return (not found)

Figure 2.1: Pseudo-code (left) and kopenogram (right) of a sequential search algorithm with the time complexity $O(n)$. The red blocks are commands, i.e., setting the index or returning results. The green block represents the loop with entry condition. The index is incremented and in programming languages this is achieved by, e.g., *for* cycle statement. The blue block is the condition (*if* statement). When this is fulfilled, then the inner blocks are executed and, in our case, the found index is returned. If no record is found, the loop will end with the last index within *Records* and *not found* sign is returned.

Polynomial time algorithms are defined as the ones that time complexity is bound by $O(n^k)$ for some constant k. The class of problems, solvable by polynomial algorithm, is denoted as *class P* and is recognized as tractable – first noted by Cobham and Cook [28, 29]. Other algorithms, whose time cannot be bound by any polynomial functions, are called exponential. These are recognized as generally intractable, as published by Garey and Cook [30, 29]. For a relatively small input data, the exact solution can be found using the known exponential algorithm with a current computation power. However, for bigger input data, the time needed to solve the problem is far a beyond reasonable amount, as seen from Table 2.1.

[2] A search problem can be solved by the sequential search algorithm, which is a brute force approach that tries all of the values. There are better approaches for a search problem, e.g., a binary search algorithm on a sorted list, taking logarithmic time complexity $O(\log(n))$, which outperforms the sequential search. B-trees are the most used structure for holding the sorted list of elements in production application or databases[26, 27].

A class *NP-complete* of hard problems was identified where it is not known whether a polynomial time algorithm exists, i.e., whether they belong to *class P*. To find a solution for such problems, the current best-known class of algorithm is based on a brute-force search of all of the possible values, which suffers with exponential time complexity $O(k^n)$. The solution founded by the previous algorithm, can be then verified in polynomial time. Additionally, the class of *NP-complete* problems are related; if some polynomial algorithms are found for one of the problems, then a derived algorithm will solve other problems of this class in polynomial time too. This feature of *NP-complete* class of problems was denoted by Cook and Karp [31, 32].

A brute-force search is a general solving technique that generates all of the possible candidates of solution and checks if the problem satisfies the problem statement. All brute-force search algorithms suffer with exponential time complexity $O(k^n)$[3].

input size n time complexity function	10	20	50	100
$O(n)$	00.01 s	00.02 s	00.05 s	00.10 s
$O(n^2)$	00.10 s	00.40 s	02.50 s	10.00 s
$O(n^5)$	01m 40.00 s	53 m 20 s	14h 48m 20s	116 days
$O(2^n)$	01.02 s	17 m 28 s	35702 years	4.02×10^{19} years
$O(3^n)$	59.05 s	40 days	2.28×10^{13} years	1.63×10^{37} years

Table 2.1: The computation time of algorithms with different time complexity functions, where one step of algorithm takes 1 millisecond. Examples of algorithm with polynomial time complexity $O(n^k)$ are compared to algorithms with exponential time complexity $O(k^n)$. It is important to note that, for the problems with an input size of 50 and greater, the exponential algorithm runs far beyond the reasonable time, compared to, e.g., the age of the universe, which is currently estimated to be 13.8×10^9 years [35].

If we presume that the technological update and computation speed will increase, the effect of the technological speedup is visible in Table 2.2. The effect on the polynomial algorithm is multiplicative. However, for the exponential algorithm, the technological speedup will only slightly increase the size of the computable problem. This is the reason why the problems solvable by only exponential algorithm are denoted as intractable.

NP-complete problems are covered in the published works of Garey and Johnson [30]. The whole complexity theory is also covered in published works of Papadimitriou [36] or Sipser [37].

The technological speedup will mainly impact the class of problems, which are solvable by polynomial algorithm. Other non-exact methods are used to find at least some solution for the problems that are only solvable by exponential algorithms. Examples of these methods are:

- The *heuristic method* tries to eliminate the number of steps of computation by some

[3] For example, depth-first iterative-deepening algorithm for a brute-force search was shown to be optimal, compared to other standard brute-force search algorithms (depth-first search or breadth-first search) [33, 34].

	present computer	10 times faster	100 times faster	1000 times faster
$O(n)$	3 600 000 $1\times$	36 000 000 $10\times$	360 000 000 $100\times$	3 600 000 000 $1000\times$
$O(n^2)$	1 897 $1\times$	6 000 $3.16\times$	18 973 $10\times$	60 000 $31.6\times$
$O(n^5)$	20 $1\times$	32 $1.59\times$	51 $2.51\times$	81 $3.98\times$
$O(2^n)$	21 N_{2^n}	25 $N_{2^n} + 3.32$	28 $N_{2^n} + 6.64$	31 $N_{2^n} + 9.97$
$O(3^n)$	13 N_{3^n}	15 $N_{3^n} + 2.09$	17 $N_{3^n} + 4.19$	20 $N_{3^n} + 6.29$

Table 2.2: Effect of computation speedup. The first value is the input size of data that is computable in one hour and the second value is the speedup that is achieved, compared to the value in first column.

implicit or explicit knowledge of the specific problem itself, e.g., eliminating solution classes that seem to not go to optimal solution. With a combination of a brute-search, the heuristic method reduces the size of all of the possible solution candidates to check. More can be found in the published works of Russel et al. [38]

- The *randomization method* uses a non-deterministic methods in some level of computation. For example, the Monte-Carlo method is used to compute problems using pseudo-random generated values and, after several iterations, statistical methods are used to compute the expected value and standard deviation [39].
- *Restriction on input data* – this is another form of using the explicit knowledge of the problem instance and it may reduce all of the possible values to be checked by brute-force search.
- *Approximation algorithm* – this not only finds some good solutions, but also, it quantifies how far from the optimal solution the result is, with some degree of probability.

2.2 Parallelization

If a sequence of instructions can be divided into parts, which can be computed independently in parallel by multiple processors, then it is possible to achieve some computation speedup using current computational technology.

A speedup of a computation on P processors can be defined as:

$$S(P) = \frac{\text{time on 1 processor}}{\text{time on P processors}} \tag{2.1}$$

Figure 2.2 shows serial and parallel execution computations of the same algorithm.

Assume $\alpha \in (0,1)$ as a fraction of the computation in one processor, which cannot be parallelized, $(1-\alpha)$ is a fraction of the computation in one processor, which can be parallelized by P processors, and t is the time needed to compute the process on one

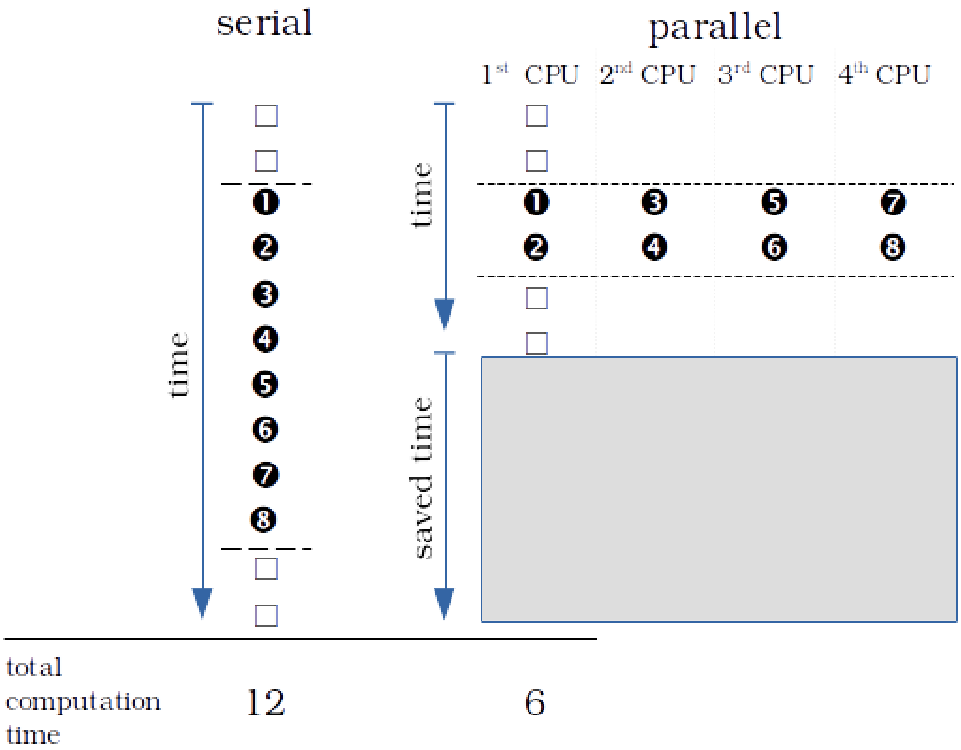


Figure 2.2: Comparison of serial and parallel executions of instructions. The instructions with numbers can be executed in parallel. In this case, the serial computation takes 12 cycles, parallel computation on four central processing unit(CPU) takes six cycles and the achieved speedup is two. If we have eight CPUs, then the computation will be finished in five cycles (achieved speedup will be 2.4 times).

processor. Assume that the overhead[4] of parallelization is small and can be disregarded. Then, the speedup can be computed as:

$$S(P) = \frac{t \times \alpha + t \times (1-\alpha)}{t \times \alpha + \frac{t \times (1-\alpha)}{P}} = \frac{1}{\alpha + \frac{1-\alpha}{P}} \tag{2.2}$$

On an unlimited number of processors, a theoretical upper bound of speedup can be formulated, which depends on α only, denoted as Amdahl's law [40]:

$$S = \lim_{P \to \infty} \frac{1}{\alpha + \frac{1-\alpha}{P}} = \frac{1}{\alpha} \tag{2.3}$$

For example, when a 33% of a computation cannot be parallelized ($\alpha = 0.33$), then the speedup on eight processors can be theoretically $S(8) = \frac{1}{0.33+0.7/8} \doteq 2.4$ and theoretical speedup on unlimited number of processors is $S = \frac{1}{0.33} \doteq 3$. See more in Figure 2.3.

However, α can sometimes be hard to estimate. Additionally, the computing of the fixed size problem on a high number of processors can misrepresent the speedup expec-

[4] Overhead is time spent on communication and synchronization among parallel processes rather than on solving the problem

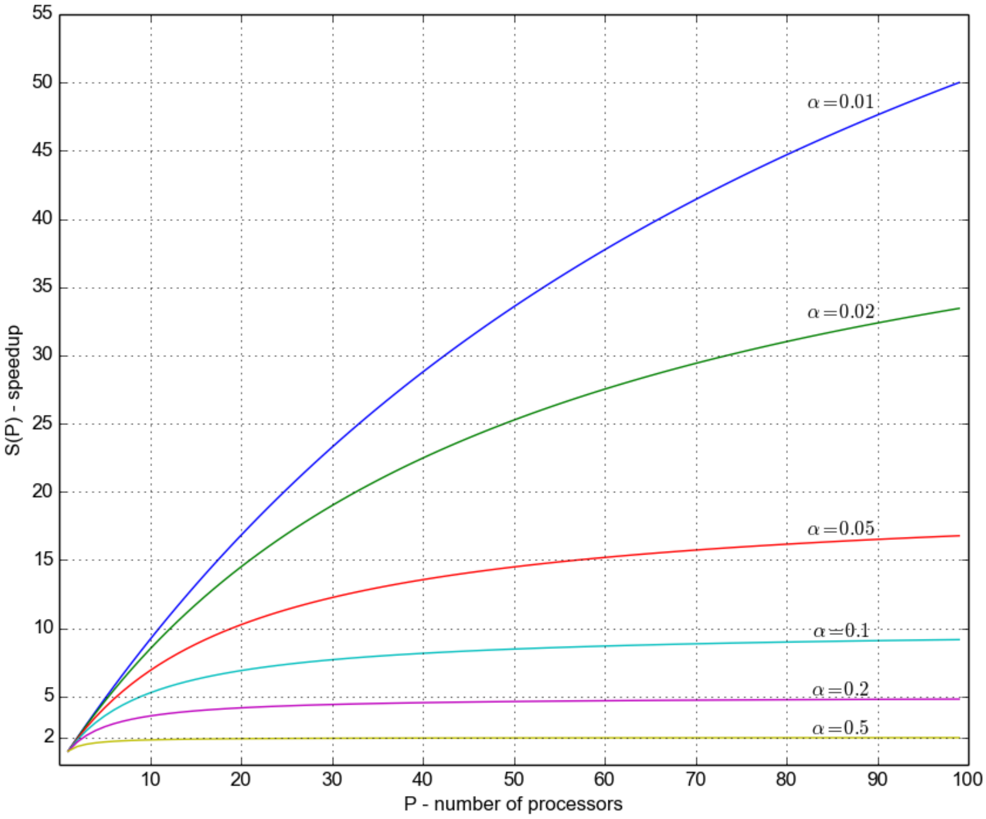


Figure 2.3: Speedup gained on 1 to 100 processors per Amdahl's law for different α values.

tation. Therefore, Gustafson reformulated the law and described another approach – to measure the fraction of the computation, which cannot be parallelized from computing on P processors. This approach estimates the time of how long will such computation take on single processor and speedup is determind from this estimation. In this case, assume that the overhead of parallelization is small and can be disregarded. The β is the "scaled fraction" of the computation on P processors, which cannot be parallelized [41]:

$$S(P) = \frac{t \times \beta + t \times (1-\beta) \times P}{t \times \beta + t \times (1-\beta)} = \beta + (1-\beta) \times P \tag{2.4}$$

This law presumes that the fraction β will not change on different number of processors, as seen in Figure 2.4.

Both laws disregard parallelization overhead, however, if there is a significant one, the speedup of parallelization will be degraded by the overhead and can eventually lead to slow down of computation. Amdahl's law (2.3) is argument that maintains the speedup limit for current fixed-sized problems. Gustafson's law, however, is more suitable to estimate speedup for bigger problem size and arguments that problem size scale with parallelization.

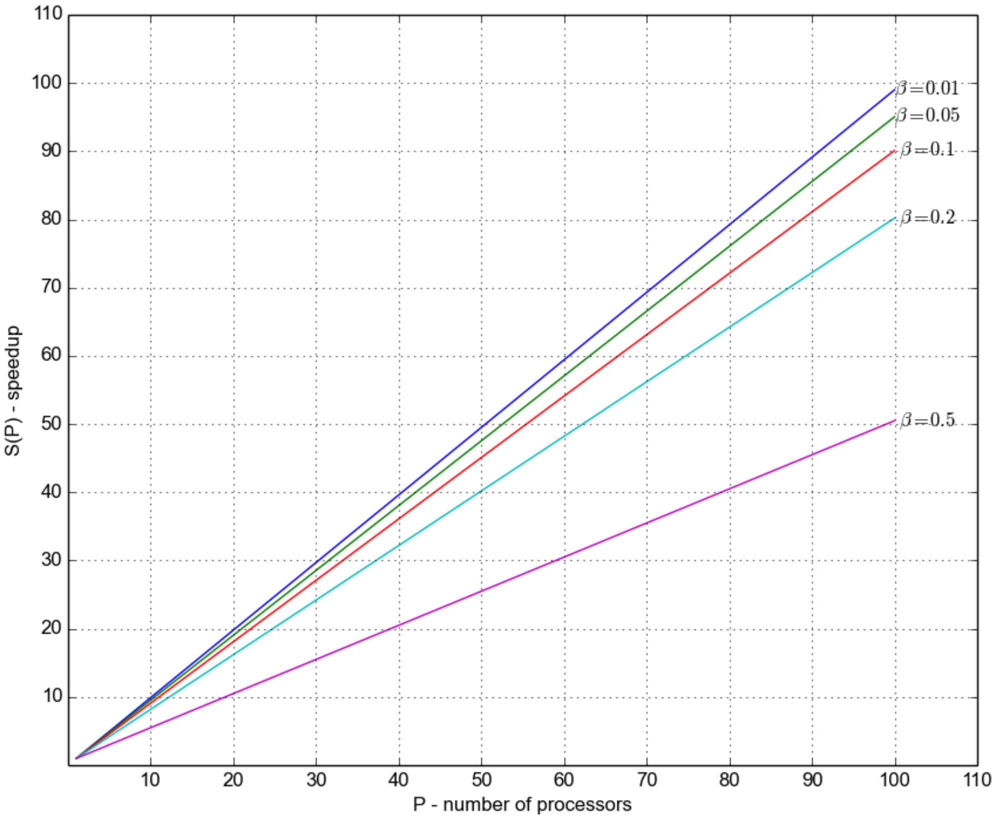


Figure 2.4: Speedup gained on 1 to 100 processors per Gustafson's law for different β values.

2.2.1 Programming Model

There are several levels of how parallelism is realized:

- *Instruction level parallelism* – if the instructions are independent, they can be executed at the same time by a multiple central processing unit(CPU), e.g., by several cores of the multi-core processor. Programs are usually written as a sequence of instructions. Instruction parallelism depends mainly on the compiler's capabilities to recognize or reorder the instruction to execute them in parallel. In the multi-core processor era, instruction parallelism started to be systematically utilized.

- *Data parallelism* – the same operation is performed on multiple data, usually arrays. The instruction is distributed into multiple processors or processor cores and is executed on elements of data structure in parallel. This is currently the characteristic feature of a computing on *General purpose graphical processing unit*(GPGPU) and programming application interface (API) such as CUDA[5] and OpenCL[6]

[5] https://developer.nvidia.com/cuda-zone accessed February 2015

[6] https://www.khronos.org/opencl/ accessed February 2015

- *Loop parallelism* – the computation may contain iteration on some large data structures. Such iterative processing is usually programmed as a loop and if i^{th} iteration is independent on previous $(i-1)^{th}$, then the iteration can be executed in parallel by different processors.
- *Task parallelism* – the computation contains parts that are independent of each other. The computation of such parts can be scheduled and distributed into multiple processors and can be computed concurrently. For example, master/worker pattern is realized by a "master" process which sets up a pool of "worker" processes and a set of tasks is distributed to them. Other example, fork/join pattern is realized by the main process, which forks into several concurent processes doing computation and at some point, they join back into a single process, which may, after some computation, fork again.

Looking at the way the processes interact, these are the most common forms:

- The *threads* are several concurrent execution paths that are independent but, in general, share the same memory. They are standardized, e.g., as POSIX threads (Pthreads) and implemented in many platforms. Further reading about Pthreads can be found in the published works of Butenhof [42]. There are also other implementations of the threads going beyond the POSIX standard, which have been introduced in other languages and programming environments.
- *Shared memory*. The *OpenMP*[7] is a shared memory application interface, which is standardized and implemented by several compilers for programming languages. It also uses a multithreaded model, however, programming is task-oriented and more abstract than using threads, as described by Chapman et al. [43].
- *Message passing*. The *Message Passing Interface* (MPI) is a specification for performing task communication by passing messages between tasks. Further reading about MPI can be found in the published works of Pacheco [44].

More information about parallel programming models can be found in a survey by Diaz et al. [45].

Some algorithms can be easily divided into independent tasks, which can be computed in parallel. If there is minimal or no need to communicate among the parallel tasks, such algorithms are called embarrassingly parallel. For example,

- Operation on matrices [46] are currently used to render 2D and 3D graphics.
- Parameter studies, where the same computation is performed using different sets of input parameters [47].
- Brute-force search algorithm, where a subset of possible candidates for solution are generated and checked in parallel.

[7] `http://openmp.org/` accessed February 2015

- Genetic algorithm and other evolutionary algorithms [48].

In contrast to embarrassingly parallel problems, there are problems, which are inherently sequential. Algorithms solving such problems cannot be significantly speeded up by parallel computing.

Both aspects of scalability(speed up gained by parallel computing) and effectivity (time demand based on the size of input data = time complexity) should be considered. Highly scalable algorithms can be outperformed by a sequential algorithm, which solve the same problem with a better time complexity, as noted, e.g., by Madden [49].

Other of parallel computing and the design and build of parallel programs were published in the earlier works of Foster [47], D.Culler et al [50] or Rauber and Räuder [51].

2.2.2 Summary

To summarize this section: parallel computing can introduce speedup on current computational technology and some computation problems may become feasible. Additionally, parallelization overhead and fraction of non-parallelizable parts should be considered as it may degrade the expected speedup. In the case of exponential algorithm (e.g., for *NP-complete* problems), the speedup will only increase the size of the solvable problem slightly (see Table 2.2) and some problems cannot be (or it is believed that they cannot) significantly speedup by parallel computing. Task parallelism and distributed computing will be considered in later text of this thesis.

2.3 Distributed computing technologies

Distributed computing is based on the idea of spreading a computation task into a set of computers, which are connected via a computer network.

The main motivations of using distributed computing technologies are to:(1) share, store and exchange resources (2) provide and consume computational services (3) access a much higher capacity of storage and computation than is available locally and (4) connect people.

To manage distributed computing, several challenges are maintained such as synchronization (the exchange of messages in a computation workflow) to achieve, e.g., mutual exclusion (when a task needs exclusive access to some resource), prevent deadlock (no progress is possible) or resource starvation (when resources – such as processor time – are not scheduled for a particular task for some reason and the task cannot finish computation). Distributed systems offer some sort of fault tolerance (managing fault of a node during computation) or security (encryption of communicating channels and stored data, authentication and authorization to access some resources or data) etc. The topic of distributed computing is covered in the published works of Tannenbaum [52].

An extreme example of distributed computing is the Internet. Here, computers are interconnected via the family of TCP/IP protocols. For example, the World Wide Web (WWW) is based on HTTP protocol and web server located by it's Uniform Resource Locator (URL) (is specific type of URI) returns a hypertext document usually in HTML format and other related format and technologies for text and multimedia. The standards and protocols of WWW are primarily maintained and developed by W3C consorcium[8].

For scientific purposes, distributed computing infrastructures evolved into sets of clusters, computing centers or individual computing resources that are owned by different subjects. A continuous effort is being made to join such resources into a federation of computational capacity via high-speed computer networks. This allows a better virtual capacity to be obtained. Some minimal requirements were formulated, as well as defined standards for network protocols and services, which a distributed infrastructure should fulfill and provide. Such infrastructures are currently distinguished as grid computing or cloud computing and their users can gain access to a much higher virtual capacity than accessible locally. Users can also access remotely specialized devices, which are not available within their institution.

2.3.1 Programming Model

A parallel programming model (section 2.2.1) is used to realize the distributed computing in a local computer or server. An additionally higher level of task interaction is realized via a shared distributed file system or by messages that are passed over a computer network. Looking at software layers, distributed computing usually incorporates one or several new layers. For example, middleware is a layer of software, which delivers defined services and application interface (API) hiding specific platform dependent implementation.

Software Architecture

As algorithms and programs are needed to solve an increasing number of problems and changing requirements, a new view on program and algorithms – software architecture – is needed in order to construct and order several programs and algorithms into a more robust system, aiming to solve a broader set of problems.

Within distributed computing, the major software architecture is based on client-server architecture (Figure 2.5), peer-to-peer architecture or more layered architecture patterns.

Service-oriented Architecture (SOA) is a high-level programming model, which is based on self-contained units of functionality – service – and wrapped with documented interfaces. SOA introduces a new layer – service layer – in client-server architecture, which separates service interface from its implementation. T. Erl describes further SOA principles and paradigms [53].

[8] http://www.w3.org accessed April 2015

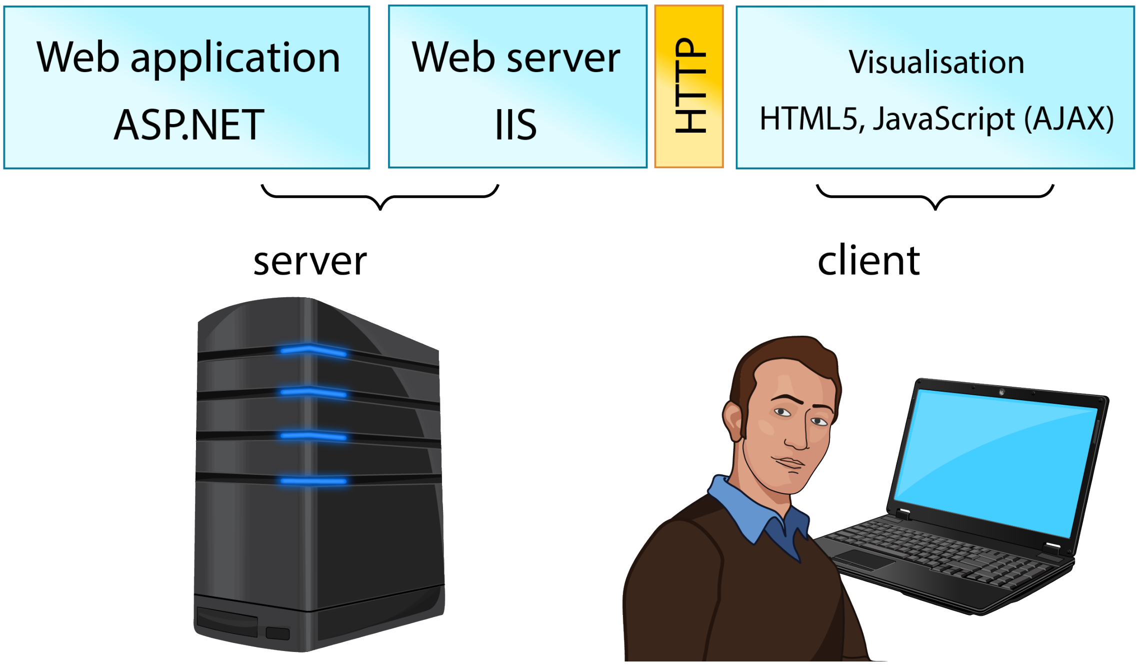


Figure 2.5: Example of client-server architecture involving a web server, which is middleware between web application and server platform. A hypertext document is retrieved from a server via HTTP protocol. Client visualizes a hypertext document in HTML format and may use some interactivity written in Javascript code. Server implementation is in this case realized by Microsoft's Internet Information Server (IIS) and web application is written using ASP.NET server-side Web application framework(`http:\www.asp.net`).

Another approach represents the objects and data of a system as resources with a standard set of operations: create, read, update, delete (CRUD). Representation State Transfer (REST) specifies several architectural constraints that help scalability and performance. It presents functionality via a fixed number of operations and uniform resource locations (URLs), as proposed by Fielding [54]. The constraints of REST are: application is statelessness, resource-orientated with uniform interface (CRUD) and hypermedia-driven, which should facilitate and optimize the processing of resources via current web based technologies, mainly HTTP protocol.

While SOA focuses on application design and easily turning application objects into distributed services, REST is rather a set of constraints on the architecture, which are used to handle the issues of distribution within the web, as noted by Vinoski [55].

The software architecture of enterprise applications, distributed systems and some repeating patterns are cataloged in published works, e.g., by Fowler et. al[56] or Nilsson [57]. Furthermore, Hohpe et al. disusses integration patterns, with a focus on the ways of connecting heterogeneous parts of the system [58].

Types of computing infrastructure

When we focus on the architecture of middleware and the philosophy of building a computing infrastructure, these main types of distributed infrastructures are distinguished for scientific computing and are relevant to the rest of this work:

1. *Service grid computing* is based on the idea that computing resources (servers, clusters and special hardware) are owned by some organizations but may be maintained by some collective organizations and shared with an effort to provide a collection of services in a best effort approach. See section 2.3.3.

2. *Desktop grid computing* is based on the idea of connecting generic desktop computers and providing the idle computation time, e.g., as a screen saver or background process to the projects. See section 2.3.4.

3. *Cloud computing* provides access to software, platform or whole infrastructure as a service in an elastic way via Internet. The resources are realized using virtualization and provisioned based on the current requirement with minimal administration intervention. See section 2.3.5 and 2.3.6.

2.3.2 Research and education network

The fundamental part of any distributed computing infrastructure is the computer network, which connects resources that are distributed in different geographical locations, generally on the Internet.

The national grid initiative in the Czech Republic, METACENTRUM[9], is interconnected via the high-speed network, CESNET 2, which utilizes the technology of transferring data over optical cables using Dense Wavelength Division Multiplexing (DWDM) [59], as seen in Figure 2.6.

2.3.3 Service Grid Computing

Service grid computing is based on a basic set of services, which are implemented by a specific grid middleware. They provide uniform interface for job scheduling and execution within the computing infrastructure. The term *grid* is used to emphasize the analogy with an electric power grid, providing access to electricity [60]. Foster et al.[61, 60] and Chervenak et al. [62] describe "data" and "computational" grids as shared hardware and software resources, which provide reliable, consistent, pervasive and cheap access to high performance computational capacities. They also provide the effective and reliable execution of requests over data, which needs sensitive controlling of terabyte storage, data transfers to gigabits per second over global computer networks and the scheduling of such data transfer, with respect to computational needs. The services provided by grid are either tools or

[9] http://www.metacentrum.cz

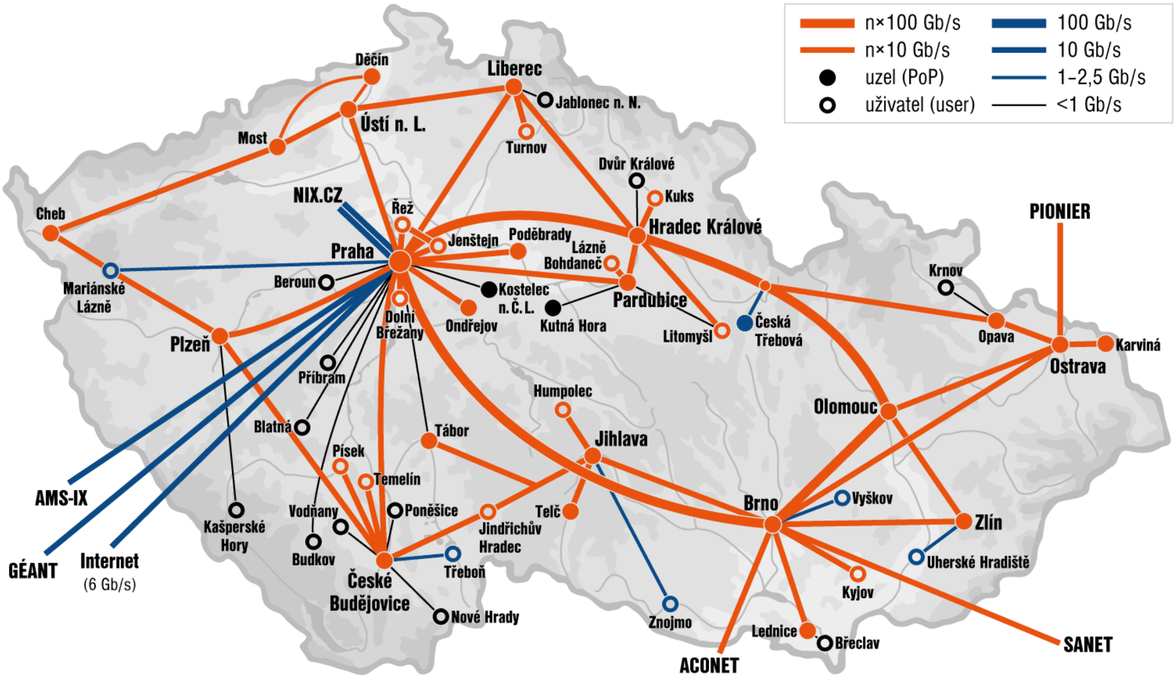


Figure 2.6: CESNET2 network topology in December 2014. It has been maintained by association CESNET (association of universities and academy of science in the Czech Republic to operate research and educational network). It mainly interconnects departments of universities and academy of sciences via a rented physical network. It provides connection to the general Internet via the Czech NIX.cz (Neutral Internet eXchange), AMS-IX (Amsterdam Internet eXchange) and European research and education network GÉANT. Sources: http://www.cesnet.cz

web services, following *Service Oriented Architecture* (SOA) for grid computing – Open Grid Service Architecture (OGSA) [63]. The security model and access to grid infrastructures are mainly proposed and implemented by a mutual authentication between users and resources via a public key infrastructure, using the X.509 certificate [64]. The administration and maintenance of such networked infrastructures are not trivial tasks and are performed by experts of institutional computing centers or national laboratories. Furthermore, interconnected sites are managed and coordinated at a national level or an international level. Such national organizations cooperate with similar national grid infrastructures in other countries. An umbrella organization in Europe – the European Grid Infrastructure (EGI)[10] – was established during 2010. It supports the integration and coordination of activities of national grid initiatives (NGI). Countries connected to the EGI and approximate number of CPUs per country in 2014 are presented in Figure 2.7.

In the last decade, there has been an acceleration in the growth of several production grid infrastructures for science. This is mainly due to the need for experiments of high-energy physics in order to process a large number of observed data in a reasonable time [65]. The Worldwide Large Hadron Collider Computing Grid (WLCG) was designed to store and process almost 30 PetaBytes of data per year in the period of 2009-2013 [66]. It is one of the largest grids to be deployed in a grid infrastructure. As the hardware infras-

[10] http://www.egi.eu

tructure is built with a philosophy of federated access to resources, which are owned by research institutions, universities, etc. Other scientists from different scientific disciplines can also become users of this powerful infrastructure. Due to the development of virtualization, the infrastructure can employ a larger set of applications and services and could be attractive for smaller scientific collaboration with different requirement on platform, shared libraries, etc.

Several grid infrastructures were established, based on different grid middlewares. Condor is one of the earliest efforts to provide access to underutilized computers, while preserving the rights of the owners [67]. Major grid middlewares that are operational in EGI are Globus [68][11], gLite [69][12] and ARC[13].

Efforts are currently being made to maintain interoperability in order to connect different applications with different resources from different grid infrastructures. Riedel et al. reports on the efforts being made to involve technology providers, as well as deployment teams, in order to participate in open standards of security, data management, etc. [70].

2.3.4 Desktop Grid Computing

Joining desktop computers from an individual user to form a voluntary or desktop grid was popularized by a project that tries to identify uncommon signals from space to search for extraterrestrial intelligence (SETI@Home)[14]. It is based on the idea that a volunteer downloads a small client program, which executes in the background or instead of a screen saver. It downloads additional data from a server on the Internet in order to be analyzed. It then sends the results back to the server. In contrast to service grids, the authorization of users cannot be so strong for volunteer individuals and some other policies, e.g., redundancy and validation, are implemented to eliminate bad or cheating results [71]. After the success of the SETI@Home, general-purpose frameworks were built in order to facilitate the development of projects that use a similar philosophy of computing on desktop computers, which are connected via the Internet such as BOINC [72], SZTAKI extension to BOINC [73, 74], XtremWeb [75] and others. Currently, there are a lot of similar projects that gain the same computer power as the SETI@Home project, e.g., the LHC@Home and it's successor, LHC@Home 2. These projects were established and used to execute selected tasks to analyze data from the Large Hadron Collider (LHC) project [76, 77].

The average performance of BOINC projects is 8.073 PetaFLOPS, with 294 764 volunteers computing on 502 238 computers (March 18th 2015). For example SETI@Home's 24 hour performance is 1.95 PetaFLOPS. Although desktop grids and service grids are two different approaches that are used to gather computing power from a large number of

[11] http://toolkit.globus.org/toolkit/ accessed February 2015

[12] http://glite.cern.ch accessed February 2015

[13] http://www.nordugrid.org/arc/ accessed February 2015

[14] http://setiathome.ssl.berkeley.edu/

computing resources, efforts are being made to interoperate and share the capacity among infrastructures, e.g., the EdgE project, published by Kacsuk et al.[78] and Urbah et al. [79].

2.3.5 Virtualization

Virtualization technology separates the physical hardware layer from the software environment, emulating a new virtual hardware layer. The hypervisor (or virtual machine manager) manages guest virtual machines and translates I/O operations between virtual device and physical device. It also translates instructions from virtual CPU to physical CPU. This introduces some overhead and performance degradation of virtual system compared to physical. However, recent virtualization technology has introduced several techniques that reduce overhead and eliminate specific hardware features and instructions which are hard to virtualize, as reported by Barham et al. and Youseff et al. [80, 81]. Thanks to them, a virtual environment has been fine-tuned for an application that can be executed on almost any hardware and platform. Here, virtualization becomes part of the solution to execute jobs of desktop grid or service grid projects on different physical platforms, as published, e.g., by Ruda et al. [82]. Currently, there are several commercial, free or even open-source virtualization implementations, which are provided by different vendors and hypervisors - VMWare[15], XEN[16], KVM[17], VirtualBox[18] etc.

2.3.6 Cloud Computing

In contrast to grid computing, where a user schedules jobs in order to access shared environment and may be influenced by other users or by the environment, cloud computing provides access to a virtual software, platform or whole infrastructure. Consequently, the user or process is given the impression of sole use. Virtualization techniques have enabled expansion of cloud computing, mainly on infrastructures that were built for another purpose. These can be rented in times when the primary infrastructure is not fully utilized [83]. Typically, the access to shared computing resources can be rapidly provisioned and released with minimal management effort or service provider interaction. This implicates other important feature of cloud-computing – elasticity – ability to scale up and down computing resources when required. These and other characteristics were collected and published by Mell and Grance in the NIST definition of cloud computing [84].

Cloud computing makes computational power and storage as utilities or commodities that can be rented. With current commercial clouds, the commercial area has evolved

[15] http://www.vmware.com/ accessed March 2015
[16] http://www.xenproject.org/ accessed March 2015
[17] http://www.linux-kvm.org/ accessed March 2015
[18] https://www.virtualbox.org/ accesse March 2015

in order to facilitate scaling up per the business needs and computational demand, e.g., Amazon EC2[19], Microsoft Windows Azure [20], Google cloud[21] and others.

Cloud computing is provided in several fundamental models or categories. Infrastructure as a Service (IaaS) offers the whole virtual infrastructure including virtual machine and network. User can install operating system images and application software. Platform as a Service (PaaS) offers a specific computing platform including development tools, programming language, libraries and databases. User can use PaaS to develop and execute his own domain-specific application. Software as a Service (SaaS) offers access to a final software, usually in a form of web application. User can use SaaS with web-browser or with some specific client software. Cloud computing in research infrastructures is being deployed next to the already existing grid infrastructures and can utilize the same hardware resources. Some methods to integrate grid computing and cloud computing was proposed, e.g., with PaaS approach by Anjum et al. [85], or with a plugin for existing grid portal by Kacsuk [86]. Currently, the most used platforms are OpenNebula [87] and OpenStack [88] in research infrastructures. Interoperability among cloud providers and a standardization of cloud computing, virtualization and related technologies are important as these would keep users from being locked into a specific cloud provider, as noted by Otiz [89].

2.3.7 Application Model

Applications that are computed within a grid or cloud infrastructure can be characterized by the quantity of tasks being performed, the size of the input data and the communication that needs to be carried out between concurrent tasks. Grid computing infrastructures are primarily utilized for computation, in which tasks take a long time. These are relatively loosely coupled and resources are used over a long period of time. Performance or capacity is usually mentioned in operations or CPUs per month or year and for such computations, the term High throughput computing (HTC) is used.

The High Performance Computing (HPC) is usually characterized by computing problems that have a small number of tasks. These are relatively tightly coupled and can take shorter time than HTC. Performance is measured in operations per second (FLOPS) [90, 91]. The grid infrastructure can involve HPC servers or clusters.

Many Task Computing (MTC) aims to bridge HTC and HPC. While the computation usually takes a shorter amount of time, the data exchange is in MB rather than in GB. Performance is measured in tasks per seconds rather than jobs per months or years and it involves computing much more heterogeneous problems, which are not "happily" parallel. Therefore Raicu et al. proposed and implemented a prototype of task execution framework, which is suitable for MTC to prevent some shortcomings of HTC or HPC [92, 93, 94].

[19] http://aws.amazon.com/ec2/ accessed February 2015

[20] http://azure.microsoft.com/ accessed February 2015

[21] https://cloud.google.com/ accessed February 2015

2.3.8 Workflows and Gateways

A workflow is an abstract description of the process of computation and data manipulation, which is specified by an expert in order to express what should be done within a distributed system. It automates the process of computation by composing data manipulation steps and tasks, as well as solving failures.

The workflow can be encoded in any programming or scripting language, however, some higher level languages have evolved. In the business domain, a Business Process Execution Language (BPEL) has become one of the most used languages for describing the workflow of orchestration of web services and transaction steps [95]. In the scientific domain, different workflow systems are operational, including BPEL, with different capabilities. Yu, Zhao et al. published the taxonomies of some of the existing workflow systems [96, 97, 98]. Workflows in cloud computing can be covered also by web technologies programming languages, e.g., Javascript [83].

The workflow system, which implements concrete workflow language, is usually tightly coupled with a specific grid computing or cloud computing infrastructure.

To connect the different grid infrastructures, a mutual workflow management system can be used to integrate them, as proposed by Kacsuk et al. [99, 86].

Scientific gateways incorporate a higher level of services for specific scientific communities, e.g., a web portal or desktop application in order to control the process of computation via a workflow[100]. Several frameworks were developed for building scientific gateways, e.g., Apache Airavata [101, 102] or WS-PGRADE/gUSE[103]. Furthermore, the concrete instances are available for a broader area of the scientific community.

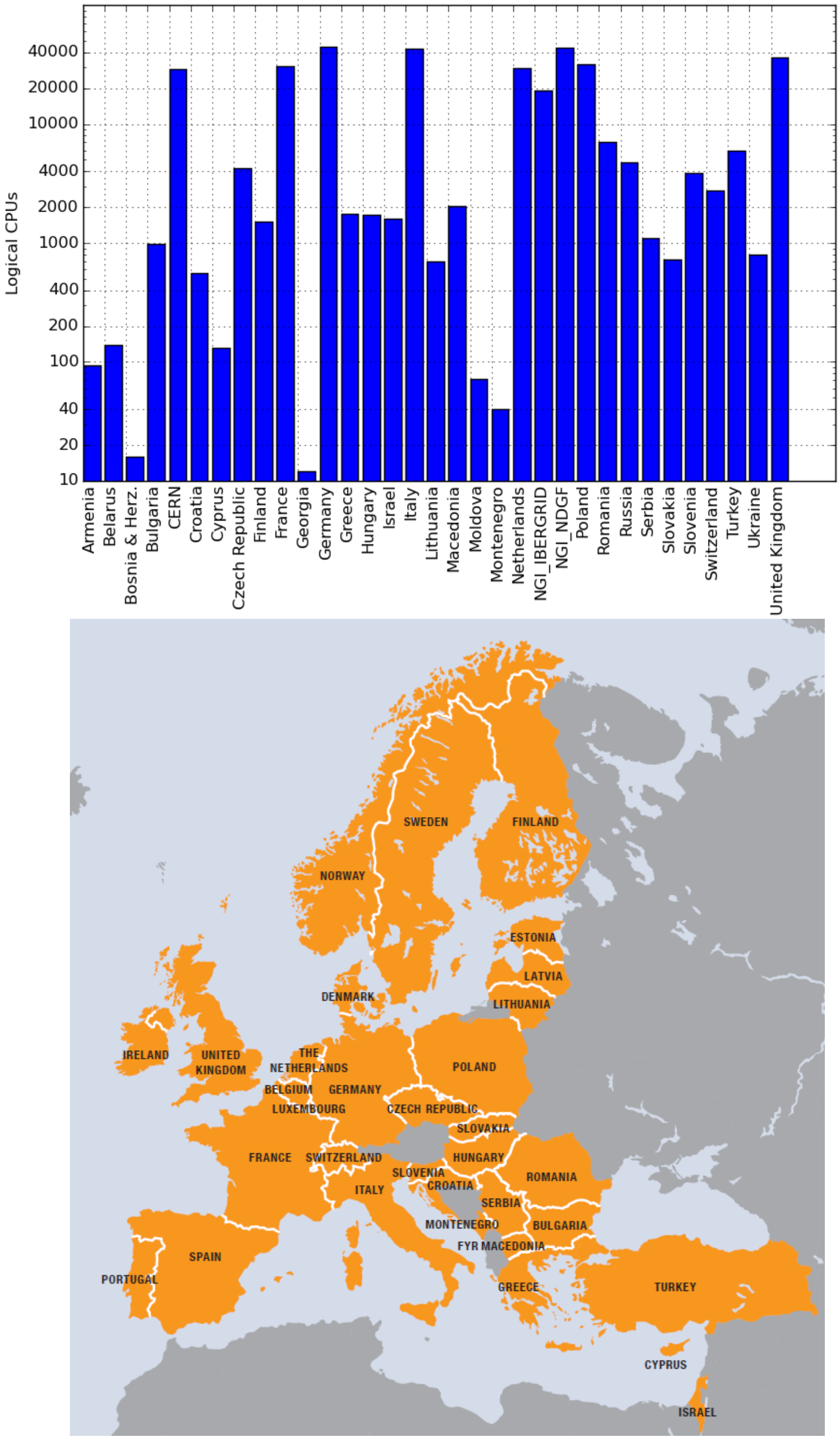


Figure 2.7: Countries involved in EGI and number of CPUs within EGI infrastructure in 2012, the total logical CPU capacity at the end of 2012 was 349 720 cores(in graph per countries). CPU capacity was 433 957 cores in March 2014. Sources: EGI Compendium 2013, EGI statistics at http://www.egi.eu.

Chapter 3

Methods

The general issue of utilizing grid or cloud computing infrastructures is the selection of the appropriate method that is used to integrate domain-specific computations into the grid or cloud infrastructure of a concrete provider.

Various tools are already available within current grid infrastructures, including open-source and licensed software for computation. The local scientific grid provider may give a list of the available applications[1]. Alternatively, application databases are available in a broader environment, e.g., in the EGI.eu application database[2]. Additionally, the workflow systems and scientific gateways that are mentioned in section 2.3.8 try to hide the complexity of grid or cloud computing infrastructures and may also be used to integrate specific domains. In designing a new application, the programming model of parallel computing and/or distributed computing (in section 2.2.1 and 2.3.1) needs to be followed, utilizing the benefits of grid computing and cloud computing.

The general approach to port applications to a grid infrastructure is to automatize what can be automatized, i.e., make scripts, configure system, prepare some user interface, integrate with existing applications, utilize protocol compatibility, etc. Additionally, the prepared template, script or application should be reused for further similar computational requests.

3.1 Sharing Medical Images

Use cases that relate to digital medical images involve image acquisition, preprocessing, storing and searching. Clinicians mainly use patient images for visualization and diagnostic purposes. Computer assisted methods facilitate the diagnostic process and involves image enhancement (to reduce the image noise and to increase the contrast), image segmentation (to separate different types of structures from the background and from each other), quantification methods (to determine the structure shape, size and volume) and

[1] applications available in CESNET METACENTRUM https://wiki.metacentrum.cz/wiki/Kategorie:Applications accessed February 2015

[2] https://appdb.egi.eu/ accessed February 2015

registration methods (to process and join multiple different images into one). Comprehensive concepts and digital techniques in medical imaging are presented in published works edited by I.N.Bankman[104].

The acquisition of a medical image is performed with different modalities (different types of equipment and sensors) by radiologists or other specialists. The DICOM[3] format and protocol has become an industrial standard for exchanging medical images electronically and in Picture Archiving Communication Systems (PACS). PACS holds the acquired DICOM images with metadata and description, which are noted by experts. PACS is usually part of the information systems in hospitals. Figure 3.1 shows the typical workflows of a medical image in a hospital.

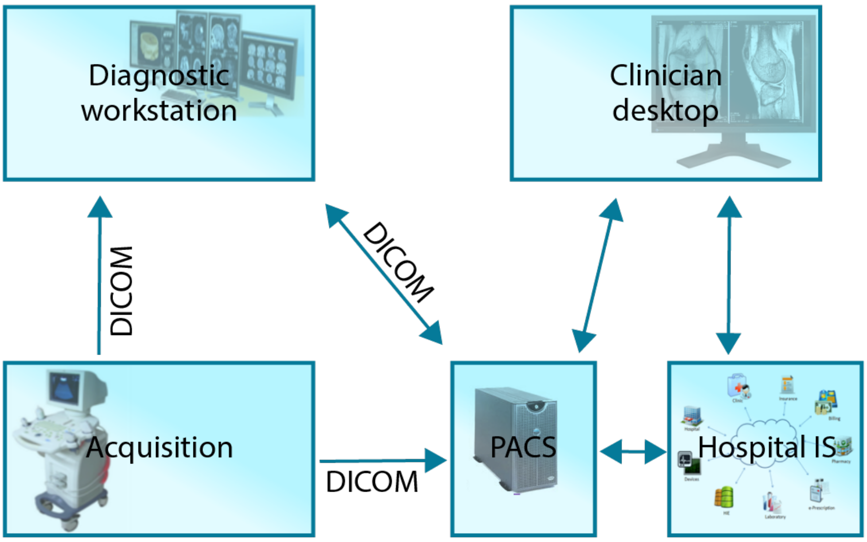


Figure 3.1: The typical workflow of a medical image in a hospital. Data acquisition is made by modalities (magnetic resonance, ultrasonography, X-ray radiography, etc.). By using the DICOM format and protocol, it can be directly transferred and visualized by diagnostic workstation. With the metadata filled by an expert physician, the image is stored in PACS. Other desktops within the hospital can retrieve the image and review the report. The hospital information system may be involved in other workflows and communicate with other formats and standards (e.g., HL7).

As the data that are processed in hospital information systems contains sensitive information of real patients, they are protected and manipulation with such data are regulated per national or international laws or agreements. The development of telecommunication and network technologies has enabled telemedicine – providing healthcare over a remote distance. This requires the sharing and exchanging of sensitive data of real patients among different healthcare providers and such data may be very valuable for further research. Security and encryption should be addressed, because the DICOM standard itself does not appropriately solve security issues. Thus, encryption during the transferring of data over to a computer network must be ensured. In the Czech Republic, there are several projects

[3] DICOM: http://dicom.nema.org/ accessed January 2015

in production that interconnect different hospitals, clinics and other healthcare organizations in order to exchange medical images. Project ePACS allows the interconnection of each participant's PACS system via a dedicated Virtual Private Network (VPN) channel to the central node. The exchange of medical images is realized via DICOM protocol by routing the communication from one VPN channel to the other[4]. Another approach is used in the project MEDIMED, which is held by Masaryk University in Brno. Instead of a dedicated VPN channel, they use Secure Sockets Layer (SSL) encryption over the standard DICOM protocol going on top of TCP/IP communication. Regional hospitals and healthcare providers are interconnected via the MEDIMED servers, as presented by Slavicek et al. [105].In other countries, cross-border teleradiology was tested in projects of Baltic e-health, R-Bay and others, which were published by Ross et al. [106] and Saliba et al. [107]. These projects focused on the sharing of medical images, as well as other knowledge and information.

Access to a wide range of medical images is needed for the research of new processing and diagnostic methods, rare diseases, developing new detection algorithm, etc. DICOM records are "de-identified" (identification of patient records are deleted and only the date of bird and other non-personal data are kept) or anonymized (additional information is manipulated to prevent potential disclosure) in order to protect sensitive personal data, but keep important information for research purposes. The Globus MEDICUS project, published by Erberich et al. [108, 109], is based on Globus Toolkit middleware in order to federate clinical and research application via a grid computing infrastructure. The project has been in hibernation since 2008 and no further development has been published[5]. Similar efforts were made by the Medical Data Manager project, which used gLite grid middleware deployed within european grid infrastructure EGEE (predecesor of EGI) and was published by Duque, Montagnat et al.[110, 111] [6]. The MediGRID project, which was published by Krefting et al.[112], added additional processing of images within selected use cases that are supported by a grid computing infrastructure [113]. The Health-e-Child project aimed to interconnect research institutions and hospitals in United Kingdom, France and Italy for the purpose of a grid-based healthcare platform for pediatric health-care [114]. The Neurist project developed system, connecting clinicians and researchers in order to improve research and the treatment of cerebral aneurysm. This provided the tools to analyze and interpret patient data and gave researchers access to a set of aneurysm data, published by Benkner et al.[115]. The SEAGRIN research project aimed to share knowledge, mainly for educational purposes, in semi-formally described semantics. Kuba et al. published this proposal and its implementation [116].

The storing of sensitive medical information, which is even de-identified or anonymized, is usually restricted. This led to the idea of storing such information within trusted institutions, e.g., hospitals, and to move and facilitate the deployment of grid services that store

[4] ePACS:http://www.epacs.cz, accessed January 2015

[5] https://dev.globus.org/wiki/Incubator/MEDICUS accessed February 2015

[6] http://modalis.i3s.unice.fr/softwares/mdm/start accessed February 2015

medical data to that institution. For example, pre-installed virtual machines can contain grid services and are deployed as a sealed grid, as proposed by Kuba et al.[117].

To summarize this section, in past years, digital medical image acquisition, storing, exchanging and processing has become common and it currently uses distributed computing techniques. Several efforts have been made to implement medical data management within grid or cloud infrastructures for research purposes and to integrate them with production infrastructures. Security is solved by authentication and authorization mechanisms, as well as by encrypting data and/or de-identification or anonymization but keeping minimal information that is required for research purposes. A related question is how easily the previously mentioned grid-based technologies can be integrated with current systems in hospitals or institutions that main purpose is common health-care. The following section describes selected methods that are used to integrate a pilot deployment of Globus MEDICUS with the current regional system for exchanging medical images - MEDIMED.

3.1.1 Methods to share medical images in grid

The Globus toolkit belongs to a group of the most used grid middleware (see section 2.3.3). The core service included in Globus Toolkit is GridFTP – grid extension to file transfer protocol(FTP). This implements strategies such as *stripping data* into multiple pieces; the *parallel transfer of data*, utilizing stripped data parts to be transferred via different channels; *partial file transfer*, some applications may not need to access the whole file but rather a smaller portion of it, etc., as described by Foster et al. and Allcock et al. [118, 119]. Other core services are Replica Location Service, which aims to localize data, and Globus Resource Allocation Management (GRAM), which provides web service and proxies to the lower level job scheduler's implementation [118].

Next to core services, the domain-specific services might be implemented for the purpose of an application that uses the Open Grid Service Architecture (OGSA). Globus MEDICUS [108, 109] implements a DICOM Grid Interface Service (DGIS) and integrates the open-source PixelMed™ Java DICOM Toolkit[7] into a web service, communicating via the DICOM protocol. Furthermore, it forwards queries to the underlying services within Globus toolkit.

DGIS acts as a gateway to a grid infrastructure. As communication via the DICOM protocol is not secured, it is recommended that the DGIS be installed on the location of the PACS system or the DICOM ready modality or software. When a DICOM study is uploaded into DGIS, it is anonymized and stored. A record is made into another service's Meta Catalog service, which resides in the same domain or anywhere in the grid that is accessible via the Globus Toolkit. Such an anonymized database of DICOM records can be used to query via the DGIS interface and to, for example, integrate with web-based applications, showing records for research purposes. Furthermore, authentication and authorization can be achieved in this level. To integrate this system with an existing

[7] http://www.pixelmed.com/ accessed February 2015

system MEDIMED project [105] for sharing the medical images, the special client software "RediMed console", needs to be installed next to the DGIS. DGIS behaves as an access point to a PACS system whose records can be exchanged via the RediMed console software to other MEDIMED participants. The results of this particular deployment and integration are presented in section 4.1.

3.2 Voice Science

With the introduction of objective data analysis and laryngoscopy methods, voice science emphasized the cooperation among laryngologists, speech pathologists and voice teachers. The normal human voice ranges from 50 Hz to about 1 000 Hz and there are some individual variations. For the analysis of a digitally recorded voice, either habitual or singing, the Discrete Fourier Transformation (DFT) is used to produce a frequency and amplitude analysis of the recorded input voice samples. One of the most used class algorithm to compute DFT is the class of Fast Fourier Transformation (FFT) with computational complexity $O(n \log(n))$ [120, 121]. The result of the analysis can be visualized in a voice range profile (VRP) and the significant difference between an untrained and trained voice can be seen, as published, e.g., by LeBorgne and Weinrich showing a significant difference of VRP after nine-month training [122]. Furthermore, some disorders can be quantitatively seen, which were published, e.g., by Wuyts et al.[123].

Another method that is used to analyze vocal chords is laryngoscopy. The videostroboscopy and high speed video in laryngoscope methods produce videos showing real movement of vocal chords. The videokymography method, introduced by Švec et al., complements the videostroboscopy. It allows the visualizing and analyzes of the movement of vocal cords. These movements are recorded by a high speed camera and a new image, constructed from selected line of recorded video, is shown on standard TV or monitor using low frequency [124, 125].

In the case of a recorded sound and further analysis, there is a question about how such a service can be integrated in a grid computing or cloud computing environment in order to provide access to a complex application for non-technical voice specialists. Additionally, analytical software was already developed and calibrated for selected types of microphones in the MS Windows platform by Frič et al. [126, 127]. Therefore, it was proposed and implemented a method that provides remote access to the analytical software. Section 3.2.1 describes how the analytical software was customized with a remote desktop protocol (RDP). Results are described in section 4.2. A similar approach can be used for processing video recordings from a laryngoscope, however, the practical limits are discussed in section 5.

3.2.1 Methods for Remote Analysis of the Human Voice

Terminal access to some remote computational capabilities, e.g., remote command-line or remote execution, is another integration strategy that is used for some remote infrastructures. Secure Shell (SSH) is used to establish a secure channel via an unsecured network (e.g., the Internet) from an SSH client to a SSH server. This is a basic method that is used to access a grid computing infrastructure. Remote Desktop Protocol (RDP) is a proprietary protocol that is used for desktop sharing. It was primarily developed in a Microsoft Windows platform, however, today, clients and servers exist for several other platforms.

The software for parameterized Voice Range Profile (ParVRP) and Voice Range Profile in Real time (RealVoiceLab) was already developed and calibrated for selected types of microphones in an MS Windows platform by Fric et al. [126, 127]. Its implementation is carried out in an MATLAB environment, utilizing Signal Processing Toolbox[8]. It is compiled with a MATLAB Compiler and distributed as an executable.

Instead of migrating the application into some compatible platform for grid middleware, a virtual machine was introduced and access to the software was provided via a RDP protocol. RDP itself contains the redirection of several services, e.g., sound recording or drive access. As the default sound recording redirection introduces some sound degradation without control, I proposed, implemented and integrated a custom RDP plugin with the ParVRP and RealVoiceLab software in order to redirect the sound recording without the loss of information. The technical details are in Appendix B.

The computation of frequencies and amplitude from the recorded samples utilizes the effective Fast Fourier Transformation, which has time complexity $O(n \log(n))$. The benefit of deploying such an application in distributed infrastructures is the immediate access to updated software. Additionally, a collection of anonymized records of voice samples and results are very useful for further research and education. The possible disadvantage is the need to access to Internet.

This type of application can be packaged as a virtual machine template and configured within different types of cloud infrastructures. Together with a script or web portal, the on-demand deployment can be automated. The client part (RDP client) needs to connect to the appropriate instance. The results of such a deployment are discussed in section 4.2.

3.3 Computational physiology

A mathematical formalization of the fundamental knowledge and relation among a biological system – a mathematical model - is used as a base abstraction in order to utilize the current discoveries of the genomics and proteomics. It is also used to formalize the knowledge and construct a "Physiome Model". By definition, a model is the simplification of a complex reality.

[8] http://www.mathworks.com/products/signal/ accessed February 2015

Constructing the models and integrating them into a complex entity, which can be used for further purposes, is schematically illustrated in Figure 3.2. The measurements are carried out in laboratories or in hospitals. Lumped parameter models are usually represented as ordinary differential equations and differential algebraic equations. They characterize the reality as a topology of discrete elements. The imaging methods for processing and analysis (section 3.1) are used to construct 3D models from segmentation and generate mesh representations that are connected to physical principles.

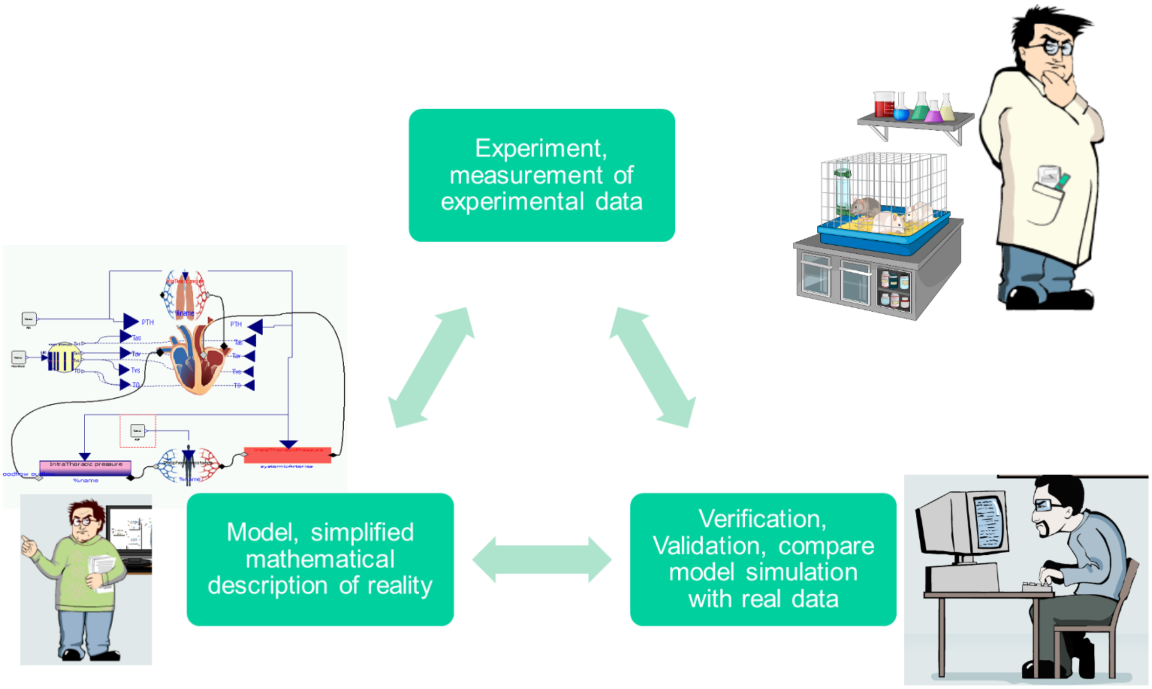

Experiment, measurement of experimental data
Model, simplified mathematical description of reality
Verification, Validation, compare model simulation with real data

Figure 3.2: Schematic illustration of the scientific process. The experiments produce data that are interpreted and a hypothesis is formalized as a model. Validation compares the model simulation with the experiment, if the model satisfies the criteria - if it is in agreement with real experiments, then the validated model can be used for other purposes.

The application of mathematical modeling techniques towards biomedical research is sometimes called systems biology. This approach combines the reductionism and integration, as denoted by Kohl et al.[128]. Application towards clinical practice includes the quantification of the diagnostic index or treatment strategy. It is a goal to develop tools, database models and methods of several Physiome projects, e.g., VPH-Physiome project presented by Hunter et al.[129].

One of the earliest complex and integrative modeling efforts was a model of circulation and its regulation, which was published by Guyton et al. in 1972 [130]. It continues as a "Human Model" or "HumMod" via derivative and technological upgrades, as introduced by Hester et al. [131, 132], with a focus on integration efforts. A different approach of modeling the human physiology is to construct a database of smaller models, which focus on some particular physiological phenomenon. For example, the NSR Physiome project introduces a JSIM[9] Java-based simulation system in order to support modeling in physiology. A repository of several hundred models was published using this system [133]. A similar effort is made by the IUPS Physiome project and repositories of the models are based on XML standard languages CellML and FieldML [134, 135]. The Systems Biology Markup Language (SBML) is used for modeling a biological system at the level of biochemical reaction and regulatory network. Another database collects several hundreds of curated and non-curated models [136, 137].

3.3.1 Modeling Methodology

The methodology of formalizing mathematical models is influenced by the abilities of underlying modeling language that is used. JSIM, CellML, SBML or HumMod are domain specific languages and the tools that are primarily developed within physiological or systems biological communities. Other authors use commercial or industry standard tools for mathematical modeling and computing. For example, Kofranek et al. described Guyton's 1972 model in MATLAB® Simulink [138] and the derivative HumMod in acausal object-oriented Modelica language [139, 10]. Fernandez et al. described models of cardiovascular pulsatile system using MATLAB Simscape [140] and recently, in Modelica [141].

The Modelica language is an object-oriented, equation-based and acausal modeling language standardized and maintaned by the Modelica association[10].

Thus, there is an open debate as to whether in-house domain-specific language and tools, like JSIM, CellML and FieldML, SBML or HumMod, have reached their capabilities for representing complex models. Only the HumMod achieved the integrative approach, building a complex integrative model of human physiology using a lumped parameter approach. However, as the HumMod modeling technology is maintained by a small team of experts, it is not used in broader physiology community.

Therefore, I contributed to the modeling methodology beneficial for complex models, which key features are the acausal modeling technique and object orientation. This keeps the complex model structure decomposed into understandable and maintainable parts and allows the complexity of models like HumMod to be covered.

The paper [5] *Modeling of Short-term Mechanism of Arterial Pressure in the Cardiovascular System: Object-oriented and Acausal Approach* in Appendix E published disputation about causal and acausal approach in using Modelica for modeling pulsatile cardiovascular system (CVS) and possible enhancement for more complex models.

The paper [6] *Simple Models of the Cardiovascular System for Educational and Research Purposes* in Appendix F, published detailed methodology of modeling lumped parameter pulsatile CVS in Modelica.

A common guide to the Modelica language and its capabilities can be found an the published works of Fritzson [142] and in the on-line works of Tiller [143].

[9] JSIM: http://www.physiome.org/jsim/ accessed January 2015

[10] http://www.modelica.org accessed February 2015

3.3.2 Identification of physiological systems

Usually, some knowledge of the system - the structure - is available and unknown coefficients (parameters) remain unknown. Once the model is formalized and constructed, a further problem is to estimate the model parameters so that the model reproduces a real world system. This procedure is sometimes called system identification and the objective of the parameter estimation is usually to minimize the following function (to find the least amount of differences between the predicted and measured values):

$$f(\vec{p}) = \sum_{i=1}^{n} (M(t_i, \vec{p}) - d(t_i))^2 \rightarrow \min \tag{3.1}$$

where $\vec{p}$ is the vector of values of parameters, $M(t_i, \vec{p})$ is model simulated at time t_i with the given parameter values $\vec{p}$ and $d(t_i)$ is the measured experimental value at time t_i. In general, mathematical models of biological systems are, in most cases, non-linear some of them are non-differentiable. Therefore, global optimization methods must be used in general. Algorithmically, the problem of parameter estimation was shown to belong to the *NP-complete* problems, published by Hofmann [144], which implies that the best known exact algorithm solving this problem has exponential time complexity, e.g. based on brute-force search – trying all possible values of the parameters and simulate the model with them, finding the minimum of the objective function (3.1). Further reading about parameter estimation and system identification can be found in published works edited by Eykhoff [145], or in the published works of Khoo [146, p. 159].

The heuristic methods (evolution strategies), randomization methods (Monte-Carlo method) and others are commonly used as global optimization methods in order to find at least some solution in a reasonable time. Evolution strategies have been identified as robust, having the potential to utilize parallel computing methods, as shown by Moles et al.[147]

Parameter estimation and further analysis methods are part of specialized mathematical software. For example, Pruet et al. used Metropolis algorithm to produce a distribution of parameters in order to calibrate a model of the human cardiovascular physiology. This was further tested against the predictive ability of circulatory failure and statistical methods were used with the software Wolfram *Mathematica* [148]. The iterative improvement method in the MATLAB Simulink® was used by Takahashi et al. in their estimation of two parameters of a simple cardiovascular model [149]. Furthermore, Abbas et al compared several methods in their estimation of multiple parameters of a cardiovascular system in MATLAB Simulink®[150].

Maffioletti et al. published a GC3Pie framework, which utilized evolutionary algorithms. They introduced workflow to identify parameters of models for economical predictions, using grid computing infrastructure [151]. Humphrey et al. calibrated hydrology models, utilizing a commercial Windows Azure cloud computing infrastructure and achieved significant speedup [152].

3.3.3 Methods for Parameter Estimation

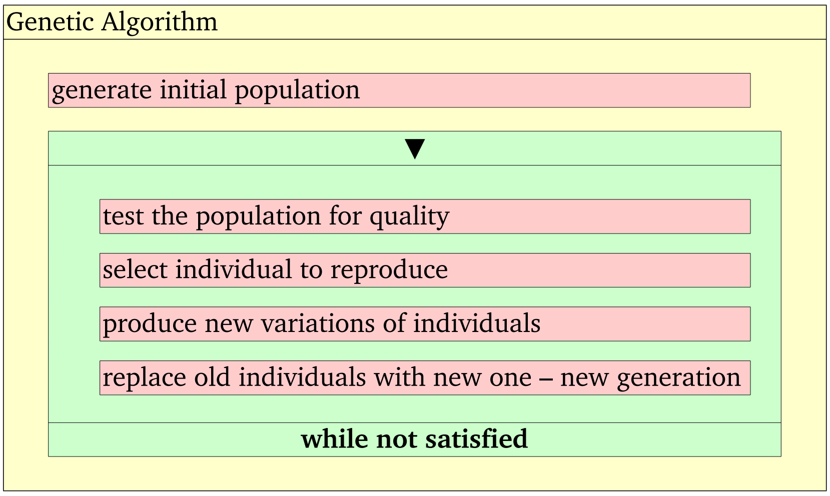

Genetic Algorithm
generate initial population
▼
test the population for quality
select individual to reproduce
produce new variations of individuals
replace old individuals with new one – new generation
while not satisfied

Figure 3.3: Kopenogram of a genetic algorithm.

An evolutionary algorithm can be used as a heuristic strategy for finding global minimum or maximum. It can also be used to estimate the parameters of a model. A genetic algorithm is a type of evolutionary algorithm, which encodes individuals as binary string. It was introduced by the likes of Holland[153]. These algorithm steps are schematically presented in Figure 3.3.

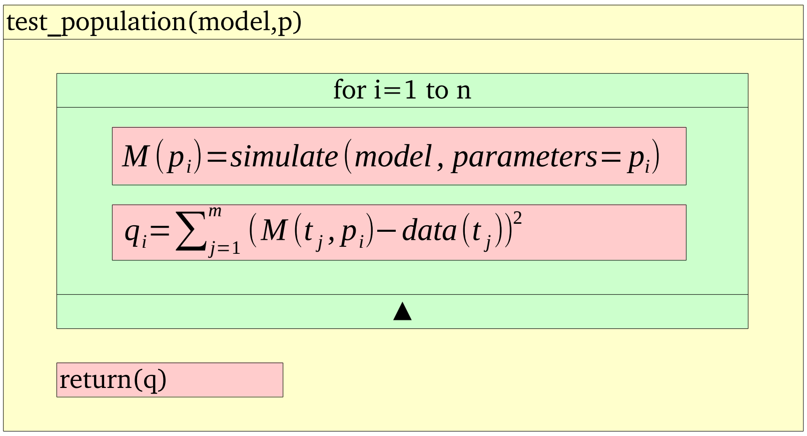

test_population(model,p)
for i=1 to n
$M(p_i) = simulate(model, parameters = p_i)$
$q_i = \sum_{j=1}^{m} (M(t_j, p_i) - data(t_j))^2$
▲
return(q)

Figure 3.4: Kopenogram of the specific test of a population for quality, in the case of parameter estimation used by a genetic algorithm. The model is simulated according to individual i with parameters p_i and the quality q_i is counted per the objective function 3.1.

The iteration within the loop "▼ . . . *while not satisfied*" depends on the previous iteration and, thus, it cannot be parallelized. However the *test the population for quality* has an algorithmical structure in Figure 3.4 for the parameter estimation. Each iteration in the loop "*for i=1 to n*" is independent and, therefore, loop parallelism (section 2.2.1) can be utilized and implemented here.

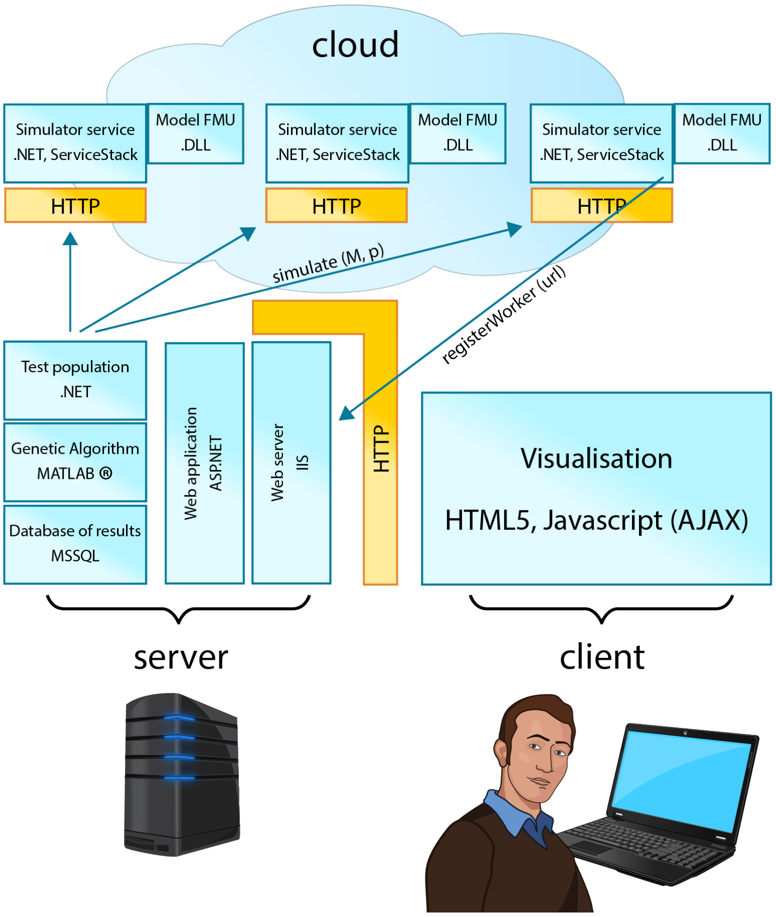


Figure 3.5: Architecture of a system that employs genetic algorithm and distributes the task *simulate* into a cloud computing environment.

Architecture of System for Parameter Estimation

The proposed architecture of the system for parameter estimation (Figure 3.5) was influenced by the need of some interactivity and for the overall accessibility for users, which is fulfilled by the web UI. The key part of the system is the model that is exported into a binary platform-dependent library. The specific model of a studied system that is implemented in Modelica is exported into a standard Functional Mockup Unit (FMU). Functional Mockup Interface (FMI) is standardized XML metadata description, packaged together with a binary library .DLL (or .SO), following a standardized API, published by Blochwitz et al. [154][11]. In the time of writing this thesis, the most stable Modelica tool was Dymola version 2015[12], and most stable export was to FMU for a MS Windows platform.

The parallelization is implemented using threads. For this, in *test_population* method is used, which, within a loop, follows a fork/join pattern – the created threads simultaneously ask for simulation results with a parameter set and the main process waits until all of the results are returned before computing the full vector of quality evaluation q.

[11] https://www.fmi-standard.org/ accessed February 2015
[12] http://www.dynasim.se - Dymola tool, accessed March 2015

Model specific FMU is packaged with a .NET ServiceStack framework[13] and it exposes a simulation functionality as a RESTful web service, which can be accessed and orchestrated by the *test_population* algorithm. The implementation of genetic algorithm is reused from MATLAB ™Global Optimization Toolbox[14] and, with a database of results in a SQL database, is integrated with an ASP.NET web application. This presents a web user interface and functionality to a user. Section 4.3 describes the results of applying the methods and deploying the designed system in a local cluster and cloud computing infrastructure.

3.3.4 Parameter Sweep

After the parameter estimation, further problem arise regarding the structural identifiability and analysis of sensitivity to the estimated parameter values[146, p. 176].

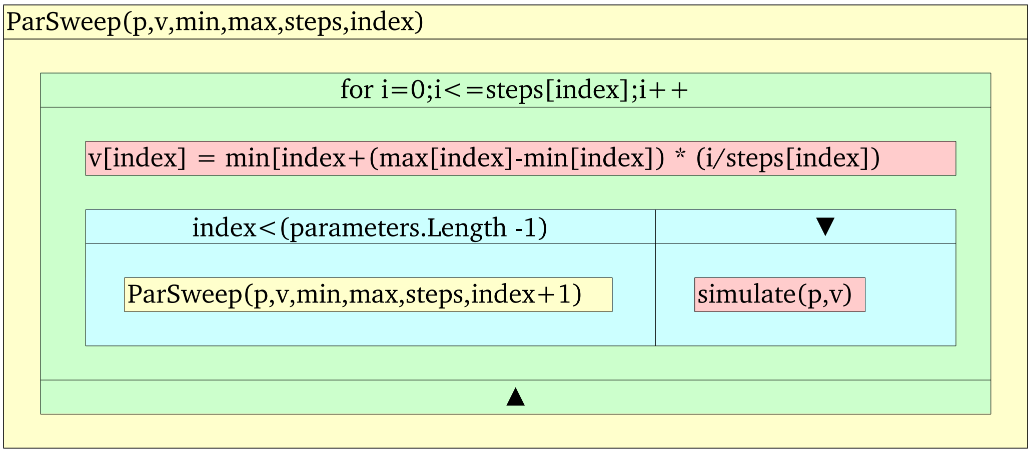

ParSweep(p,v,min,max,steps,index)
for i=0;i<=steps[index];i++
v[index] = min[index+(max[index]-min[index]) * (i/steps[index])
index<(parameters.Length -1)
▼
ParSweep(p,v,min,max,steps,index+1)
simulate(p,v)
▲

Figure 3.6: Kopenogram of a recursive parameter sweep algorithm. p,v,min,max and $steps$ are arrays with the same dimension that hold the parameter name, value, starting and stopping value and number of steps that have to be performed between the starting and stopping value per each $index$.

Parameter sweep (PS) is one of the techniques that is used for a sensitivity and uncertainty analysis, which is based on the changing selected parameters, simulating whole model and quantifying the change on model behavior with different parameters. An uncertainty and sensitivity analysis tries to determine how a change in the value of a parameter contributes to the model output and how the estimation of parameter values is robust against errors of real data measurement. The various methods for carying out an uncertainty and sensitivity analysis have been published, e.g., in reviews by Helton et al. [155] or in publications by Saltelli et al.[156, 157].

The recursive algorithm of a parameter sweep for exploring parameter space (in Figure 3.6) generates a tremendous number of simulations. Presuming that *simulate* operation takes a constant time for any parameters (which, in general, is not true), the time complexity of PS is exponential $O(\prod_{i=1}^{n})\mathrm{steps}_i) \approx O(k^n)$ where $k = \max_{i=1}^{n}(\mathrm{steps}_i)$ and n

[13] https://servicestack.net/ accessed February 2015

[14] http://www.mathworks.com/products/global-optimization/ Matlab Global Optimization Toolbox, accessed March 2015

is number of parameters to be swept. For example, for 1 000 values for each parameter: $O(1000^n)$. The large number of distinct simulation can take a tremendous ammount of time on a single computer. However, in contrast to parameter estimation, each simulation is independent and a PS algorithm is determined as embarrassingly parallel. It is implemented in many grid computing projects and workflows, e.g., P-Grade portal, as published by Kacsuk et al.[86].

A system was proposed with customized BOINC platform[72][15]. The task parallelism and master/worker programming model (mentioned in section 2.2.1) is utilized. The Modelica model exported as FMU for Windows platform is integrated with BOINC wrapper. As a whole, it is integrated into BOINC platform, which is deployed on a server, as seen in Figure 3.7.

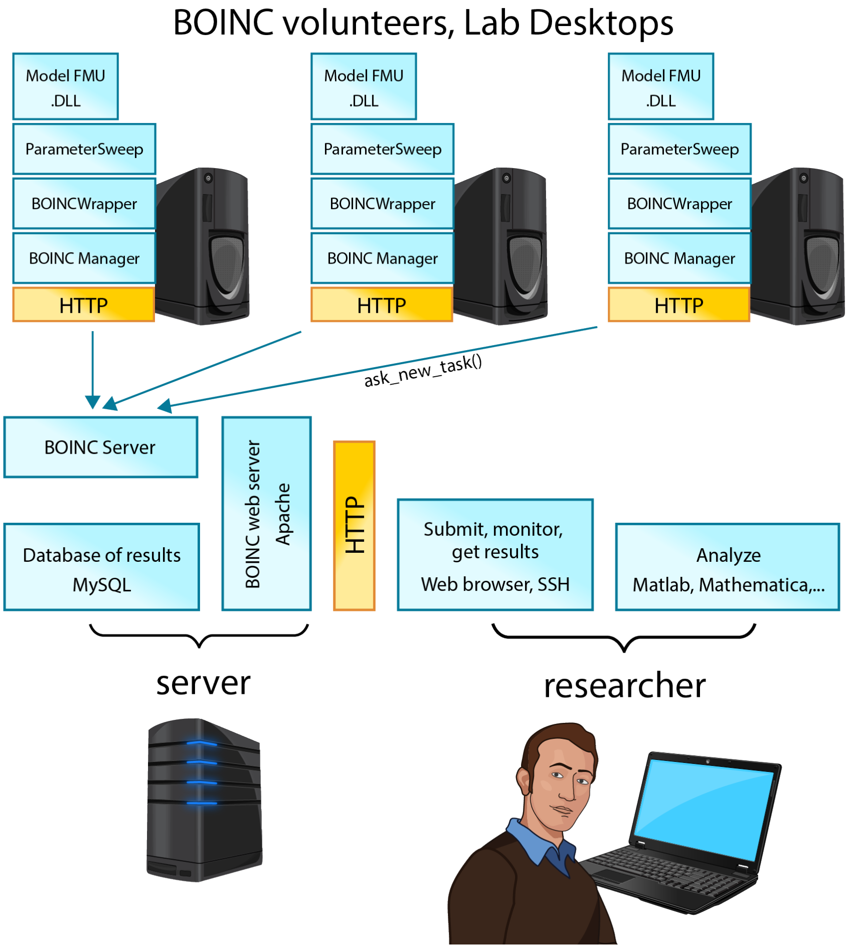


Figure 3.7: Architecture of a parameter sweep integrated into BOINC framework. The whole parameter space is divided into smaller spaces which are resolved by the BOINC workers

The results are described in section 4.3.

[15] http://boinc.berkeley.edu/ accessed February 2015

Chapter 4

Results

In previous chapters, different methods were introduced that are available for selected use cases in biology and medicine research. As each of the use cases and available systems were proposed on different operating system platforms, architecture and/or middleware, the virtualization was utilized to build the virtual infrastructure for the purposes of each pilot application. The paper [3] *Infrastructure for Data Storage and Computation in Biomedical Research* in Appendix C describes the result of establishing virtualization on a physical infrastructure in order to share computational power among different platforms.

4.1 Medical Images

The pilot infrastructure of several servers was installed in several institutions in Prague, Czech Republic. Globus Toolkit and Globus MEDICUS were installed on them. The paper [1] *Processing of Medical Images in Virtual Distributed Environment*, in Appendix A, published details about the integration of Globus MEDICUS with a MeDiMed project. It concludes that such integration via the DICOM protocol is almost seamless. Furthermore, if such a grid-based system is joined with a production system for exchanging clinical DICOM data, it could be beneficial for researchers.

4.2 Remote Access To Voice Analysis

The paper [2] *Remote Analysis of Human Voice–Lossless Sound Recording Redirection*, in the Appendix B, published technical details and results of customizing a RDP protocol for lossless sound recording redirection. It also discusses remote access via a remote desktop feature of Windows platform to an application in order to analyze human voice and produce a voice range profile for further use.

Additionally, the remote application was packaged as a virtual machine template. This was deployed in the pilot virtual infrastructure next to the test instance of Globus MEDICUS. The virtual machine template was also deployed to different cloud computing infrastructures. The first was deployed to the Amazon EC2[1] and the second to the pilot scientific cloud, MetaCloud[2]. In the EGI Technical Forum 2012, such comparison was presented to the user and technical community within CESNET and EGI organization[23].

4.3 Parameter Estimation

The paper [4] *Parameter Estimation of Complex Mathematical Models of Human Physiology Using Remote Simulation Distributed in Scientific Cloud*, in Appendix D, published the architecture and measurement of a speedup that was achieved on estimating parameters of three different types of models - from the non-complex, medium-complex and complex models. It concluded that only medium-complex and complex models may benefit from the architecture as the communication overhead may become major for simple models and decrease the overall performance.

Additionally, a scientific result was published in the paper [7] *Adair-based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity*, in Appendix G, where a mathematical model of hemoglobin integrating O_2, CO_2 and H^+ binding based on theoretical principles, which were verified on the parameter estimation algorithm system[4], together with methods available in Wolfram *MATHEMATICA* 9.0[3].

Thus, the overall performance and speedup estimation were tested against the Modelica implementation of complex physiological model HumMod [139]; the Modelica implementation of a model of hemodynamics of the cardiovascular system, published by Meurs [158]; the model of binding gases to hemoglobin, published by Matejak [7] and the trivial model of a curve $f(x)$ with four parameters a, b, c, d defined as $f(x) = a \cdot \sin(b \cdot (x - c)) + x \cdot d$ and named as "SinusCurve".

complexity	name	T_1 [s]	T_2 [s]	T_3 [s]	T_4 [s]	α	S
high	HumMod [139]	4639	4639	4618	4616	8.858×10^{-5}	11 290
medium	Meurs2011[158]	661.8	661.5	634.7	634.5	0.000 494 1	2024
low	Matejak2014[7]	17.87	17.61	1.399	1.123	0.014 44	69.26
trivial	SinusCurve	0.073	0.020	x	x	0.7260	1.377

Table 4.1: Time spent in different parts of the parameter estimation algorithm for one processor deployment utilizing virtual machine on physical hardware 2x 6-core Intel E5-2620 2GHz. Genetic algorithm works with a population of 120 individuals for 10 generations. T1 – is the whole time of the computation, T2 – is the time of the computation, which can be parallelized, T3 – time spent within the worker node, T4 – time spent in simulation, α – computed as $1 - (T2/T1)$ and S is the theoretical speedup limit per Amdahl's law $(1/\alpha)$ eq.(2.3).

The computation time of a single simulation depends mainly on the model complexity. Based on the findings, the simulations of the models were divided into four groups, depending on its demand to compute 1200 simulations. Speedup is studied for these four

[1] http://aws.amazon.com/ec2/ accessed February 2015

[2] http://www.metacentrum.cz/en/cloud/ accessed February 2015

[3] http://www.wolfram.com/mathematica/accessed February 2015

complexity	name	T_1 [s]	T_2 [s]	T_3 [s]	T_4 [s]	α	S
high	HumMod [139]	6463	6461	6451	6458	0.000 352 8	2835
medium	Meurs2011[158]	699.6	699.2	697.9	696.9	0.000 576 0	1736
low	Matejak2014[7]	2.893	2.373	1.228	1.149	0.1797	5.563

Table 4.2: Same as Table 4.1, but measured on a local server deployment, with reduced communication overhead.

groups and Amdahl's law is appropriate to estimate upper limit of it as discussed in Section 2.2.1. Fraction α and the speedup limit per Amdahl's law are stated in Tables 4.1 and 4.2.

The difference between T_2 and T_3 is an overhead, which was introduced by the network communication between the genetic algorithm and the worker nodes that were deployed in the cloud deployment, provided by CESNET NGI department METACENTRUM[4]. The network overhead can be eliminated in serial implementations by directly integrating the simulation into a genetic algorithm. Therefore, Table 4.3 considers and compares a hypothetical serial execution time, which is estimated without the network overhead.

	distributed in cloud				distributed in local cluster			
	overhead		est. serial		overhead		est. serial	
model name	$T_2 - T_3$ [s]	fraction$_{[\%]}$	T_{es} [s]	S_{es}	$T_2 - T_3$ [s]	fraction$_{[\%]}$	T_{es} [s]	S_{es}
HumMod [139]	20.98	0.4523	4619	1.005	9.858	0.1525	6453	1.002
Meurs2011 [158]	26.80	4.049	635.0	1.042	1.321	0.1888	698.3	1.002
Matejak2014[7]	16.21	90.73	1.657	10.78	1.145	39.58	1.748	1.655

Table 4.3: Comparison in cloud deployment vs. local cluster deployment of communication overhead. Its fraction in the whole computation was introduced by the network transfer speed and latency. The estimated time and speedup, if the worker is replaced by a serial version of computation without communication overhead, is: T_{es} – estimated time of serial version of computation. S_{es} – estimated speedup of serial version of computation against the parallel on one processor.

The speedup was measured on 10 CPUs till 60 CPUs and compared in order to predicted the speedup, as seen in Figure 4.1. Similar measurements with different parameters of genetic algorithm were carried out using 80 and 160 CPUs, as seen in Table 4.4.

complexity	name	$T_1(80)_{[s]}$	$S(80)$	$T_1(160)_{[s]}$	$S(160)$
high	HumMod [139]	544.5	63.98	286.7	121.5
medium	Meurs2011[158]	90.99	55.93	96.00	53.01
low	Matejak2014[7]	11.38	16.69	15.13	12.55

Table 4.4: Scalability on 80 CPUs and 160 CPUs of parameter estimation on cloud computing cluster on 5-10 virtual machines, each 16 CPUs on physical hardware 2x 8-core Intel E5-2670 2.6GHz. Genetic algorithm configured with a population 640 individuals for 20 generations, which increases about 10 times more simulation performed compared to previous tables. Speedup estimated from measuring the computation when using 1 CPU.

To summarize the results from Table 4.4 and Figure 4.1, the low complex model scales up to 40 CPUs with a speedup of 15 (compare to theoretical serial speedup of 11, if eliminating network overhead, Table 4.3). The medium complex model scales up to 80 CPUs

[4] http://www.metacentrum.cz accessed March 2015

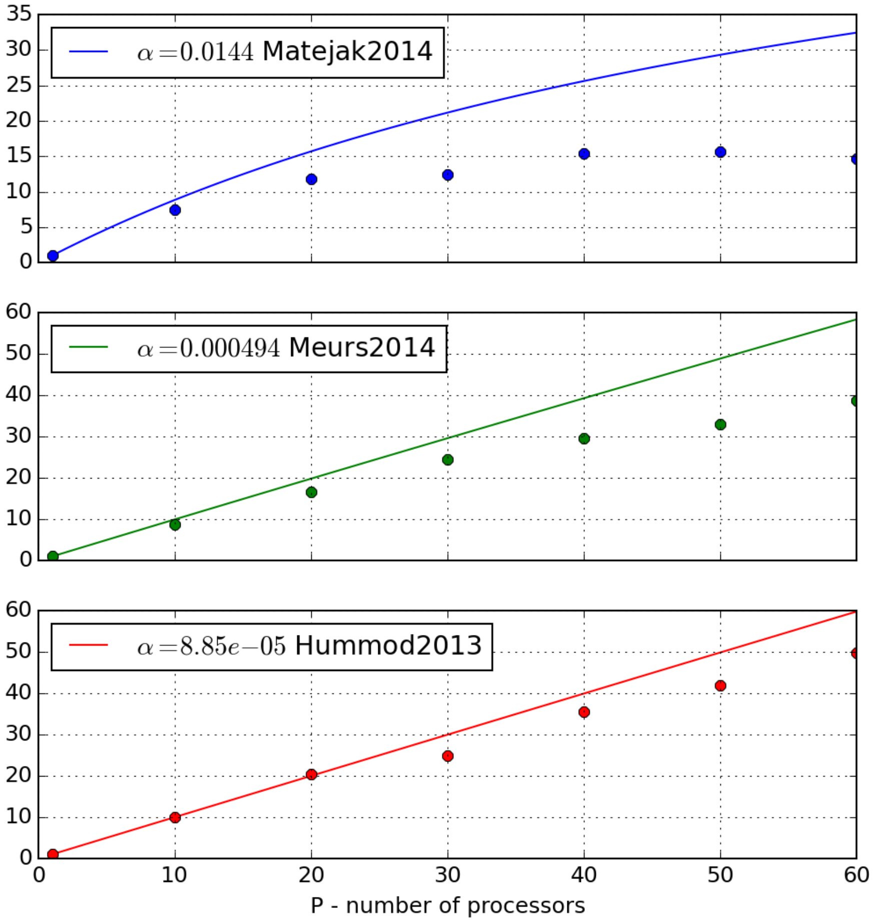


Figure 4.1: Estimated speedup (lines) per Amdahl's law (eq.2.3 [40]) for different α of several Modelica models and real measured speedup (points) on cloud deployed on 1-6 virtual machines on physical hardware (2x 6-core Intel E5-2620 2GHz, 1Gbit/s Ethernet.)

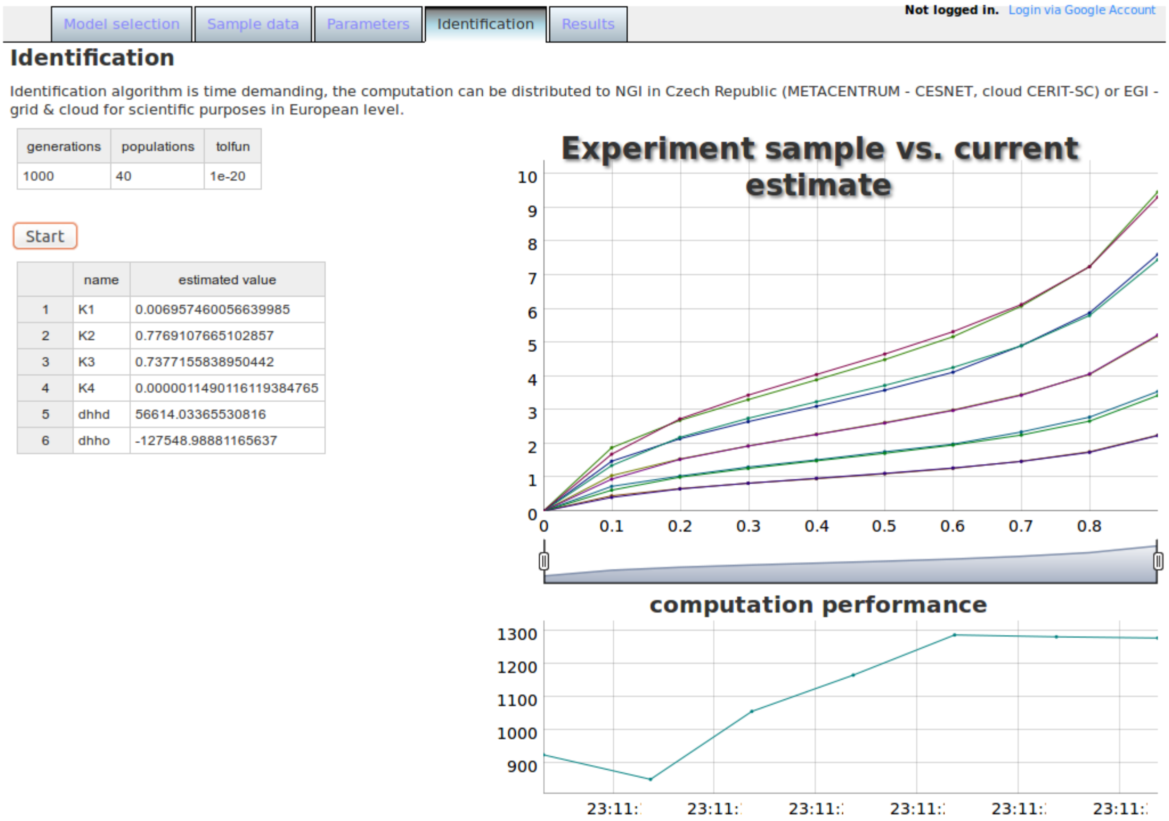
with a speedup of 56 and complex model scales up to 160 CPUs (and probably more) with a speedup of 122. Practically, good parameter estimation was obtained after 200 generations with population of 640, which implicates that the computation time can be reduced from four days to 47 minutes in the case of complex model and from 14 hours to 15 minutes in the case of medium complex model.

The deployment on local cluster reduces the communication overhead. However, in order to compute concurrently, it is limited by available processors. Thus, computing on local cluster should be considered for boundary cases like the simple models. The following statement can be made:

- If the alpha fraction is major, then a serial computation of parameter estimation algorithm without communication overhead, will perform best. This is the case for the trivial function.

- If the alpha fraction is minor, but the network overhead is still major a computation on local cluster or virtual HPC cluster should be considered. This is the case for the low complex model simulation, e.g. Matejak2014[7].

- If the alpha fraction is minor and the network overhead is also minor, then the distributed computation, e.g., in a cloud-computing environment, is worth using. This is the case of the medium and high complex model simulations of "Meurs2011" [158] and "HumMod2013" [139].



Identification

Identification algorithm is time demanding, the computation can be distributed to NGI in Czech Republic (METACENTRUM - CESNET, cloud CERIT-SC) or EGI - grid & cloud for scientific purposes in European level.

generations	populations	tolfun
1000	40	1e-20

	name	estimated value
1	K1	0.006957460056639985
2	K2	0.7769107665102857
3	K3	0.7377155838950442
4	K4	0.0000011490116119384765
5	dhhd	56614.03365530816
6	dhho	-127548.98881165637

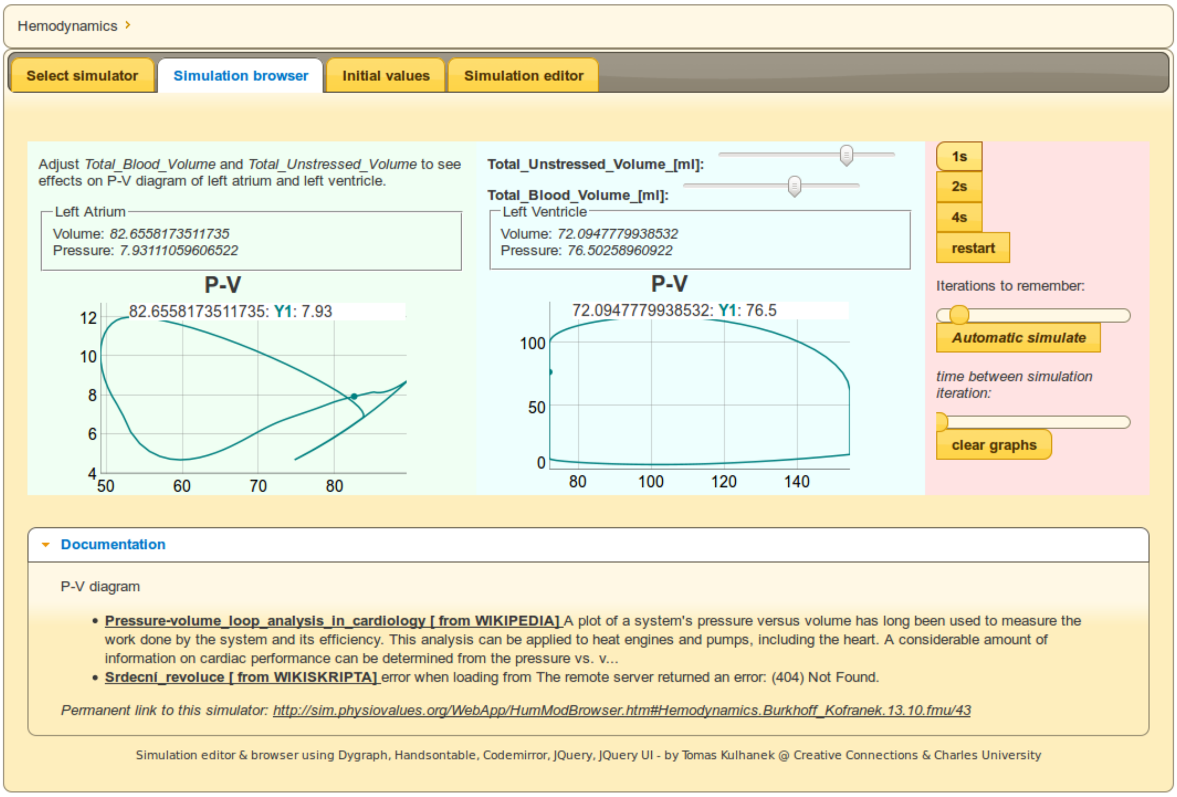
Figure 4.2: User interface of a web application for parameter estimation. In this case, for the model Matejak2014[7]. The top left table lists the parameters for the genetic algorithm(1000 generations with a population size of 40 and a cumulative change in the generation limit, which ends the algorithm earlier). The middle left table shows the model parameters and current best values, which fit the sample data. The chart shows how the sample data fits with the model simulation. The right bottom chart shows the performance of computation in a number of model simulations per second.

4.3.1 Parameter Sweep

The desktop grid BOINC system was established for parameter sweep application. The established project, *Physiome@home*, and it's project web page, http://physiome.lf1.cuni.cz/ident3/physiome, manages workunit tasks which are sent to and executed by BOINC workers. The worker application is a packaged model that is exported as FMU for a Windows platform and wrapper application, which communicates with the BOINC manager on the desired volunteer computer.

4.3.2 Remote Simulation and Local Visualization

An extra outcome of an architecture for parameter estimation is a hybrid web simulator system, where a single instance of a worker node is utilized as a back-end for the simulation engine. The front-end presented as a web application, is implemented using HTML and Javascript libraries in order to visualizes and control the simulation on the worker, as seen in Figure 4.3. The results were published in [9] and popularized in [21].



Adjust *Total_Blood_Volume* and *Total_Unstressed_Volume* to see effects on P-V diagram of left atrium and left ventricle.

Total_Unstressed_Volume_[ml]:

Total_Blood_Volume_[ml]:

Left Atrium

Volume: *82.6558173511735*
Pressure: *7.93111059606522*

Left Ventricle

Volume: *72.0947779938532*
Pressure: *76.50258960922*

Iterations to remember:

time between simulation iteration:

P-V diagram

- **Pressure-volume_loop_analysis_in_cardiology [from WIKIPEDIA]** A plot of a system's pressure versus volume has long been used to measure the work done by the system and its efficiency. This analysis can be applied to heat engines and pumps, including the heart. A considerable amount of information on cardiac performance can be determined from the pressure vs. v...
- **Srdecní_revoluce [from WIKISKRIPTA]** error when loading from The remote server returned an error: (404) Not Found.

Permanent link to this simulator: http://sim.physiovalues.org/WebApp/HumModBrowser.htm#Hemodynamics.Burkhoff_Kofranek.13.10.fmu/43

Simulation editor & browser using Dygraph, Handsontable, Codemirror, JQuery, JQuery UI - by Tomas Kulhanek @ Creative Connections & Charles University

Figure 4.3: Web application to visualize simulation, in this case, pressure volume diagrams of the left atrium and left ventricle of the model of hemodynamics of a cardiovascular system.

4.3.3 Summary

A pilot web domain was established to include the previous results and to start collecting the physiological relevant values at http://www.physiovalues.org. Further development may focus on connecting values of parameters and variables with a physiological or pathological scenario.

Chapter 5

Conclusion

5.1 Discussion

The result presented in section 4.1 is an example how a standard format and protocol DICOM is utilized to integrate current production system in order to exchange medical images (MEDIMED [105]) and a grid-based solution (Globus MEDICUS[109]). Remote Desktop Protocol (RDP) is a key standard for protocol in order to integrate the application of voice analysis[126] into a remote environment, which is accessible via the Internet. This is presented in section 4.2. In the case of parameter estimation, a key factor is the standard Functional Mockup Interface (FMU)[154], which allows the control and simulation of a physiological model in a customized tool that is not related to modeling tool. This is presented in section 4.3.

The selection of a joint element increases the chances of reusability of such a system in future development, when requirements usually change and the reconstruction of a system or architecture is needed. For example, the presented solution, which is based on Globus MEDICUS, is, in general, a data warehouse, that stores one or more copies of DICOM images. However, federated files and metadata that are stored within home institutions, which only share network infrastructure to interchange the DICOM studies, seems to be a preferred and more acceptable solution by hospitals. Thus, in their further development, the authors of Globus MEDICUS followed a way of federation of medical images that are stored within home institutions, as published by Chervenak et al. [159]. The grid computing infrastructure seems to be suitable for research and educational purposes, but not generally acceptable for clinical use.

In the case of remote voice analysis, the remote access to an legacy application via network protocol keeps the majority of user experience, as presented in section 4.2. Such service can be deployed on any web server and the occasional need to educate or perform a higher number of analysis concurrently can be satisfied with cloud computing deployment. The application process for sound signal which is currently analyzed by Fast Fourier Transformation algorithm quite effectively. Another challenge is to analyze a sound signal connected with high-speed video or videokymography methods. This need however to transfer, process and store large amounts of data (GB). With current common network speed available in departments of laryngology (up to one GBit) this may introduce impractical latency needed to transfer data to remote application and is not suitable for real-time analysis compared to analysis of sound signal. But for post-processing and storing such records, this system can bring benefits for second study, gradual improvement of diagnostic methods, etc.

In the case of the application for parameter estimation presented in section 4.3, the computation is sensitive on communication overhead. For simple models, local high performance computing (HPC) resources are most beneficial. For medium and highly complex models, the deployment of worker nodes into a cloud computing environment is worth considering. Another challenge is an optimal size of population for genetic algorithm so the algorithm will converge to some acceptable solution in a reasonable time, as Gotshall et al. proposed a method for determining the optimum population size for a given problem [160].

The parameter sweep problem is considered as embarrasingly parallel and highly suitable for high throughput computing (HTC), which is the main focus of current grid computing infrastructures. Tøndel et al. introduced methods to statistically map variation of large number of parameters and to reduce drastically the number of simulations required for parameter study [161], however only non-complex models focusing on specific phenomenon were considered.

When porting an application to a grid environment, one of the important decision to consider is the platform of the used system. The architecture, which involves computational nodes that are deployed in a cloud computing infrastructure is influenced by the fact that the model implementation is exported from a third party tool to the standard FMU library for the MS Windows platform, as mentioned in section 3.3.3. This determines the platform of the worker node and the virtualization - or, in the case of parameter estimation, cloud computing is utilized on a prepared platform with a MS Windows license. In the case of parameter sweep, a desktop grid computing BOINC worker and application for a MS Windows platform is only prepared for volunteers with the compatible system. To utilize the service grid infrastructure, an export of the model into a FMU library and implementation of the wrapper service must be done in the grid computing platform, which is usually a Linux based system. Another option is to use WINE[1] – a compatibility layer that is capable of running Windows applications on several POSIX-compliant operating systems, such as Linux, Mac OSX and BSD.

For smaller types of application and scientific community with their own tools, the question is, whether or not to invest on porting their tools to grid specific platform and parallel programming model. In the case of integrating with a service grid middleware or with desktop grid framework, expert knowledge is needed to configure and customize the

[1] https://www.winehq.org/ WINE. Accessed March 2015

system. This is the case for the sharing of medical images (section 4.1) and for parameter estimation and parameter sweep, which was tried with the desktop grid approach - BOINC framework (section 4.3.1). Virtualization facilitates the integration effort, as presented in the case of remote analysis of the human voice (section 4.2) and in thecase of deployment of worker nodes in a cloud computing environment for parameter estimation (section 4.3).

Based on previous results and ideas, the answer to the questions from the section 1.1 can be formulated:

- *Is it beneficial to utilize grid computing and cloud computing technology for the processing of medical information and how?*

  Grid computing and cloud computing may significantly speedup parameter study of medium and complex models in computational physiology. Such a speedup might influence its applicability in clinical use. For the case of sharing and processing medical images or analysis of voice signals, grid computing or cloud computing introduces technology that facilitates cooperation among a community of users from different geographically dispersed areas and facilitates the sharing of large data sets.

- *What are the limitations of processing medical information in grid or cloud?*

  Limitation are given by the effort needed to integrate or port an application carry out computation or share data. The cost of porting an application to cloud computing is reduced by virtualization technology, rather than to a grid computing environment, which needs additional work in order to adapt the application for a grid computing platform and API.

  From a programming model point of view, limitation are given by the theoretical features of algorithms and the problems to be solved. Grid computing and cloud computing are not general solutions for hard problems (NP-complete problems), as discussed in section 2.1. However, connected with non-exact methods, a concurrent processing of many tasks may bring an acceptable non-exact solution.

- *How can the grid computing and cloud computing influence the direction of biomedical research?*

  One of the concrete results presented in previous sections is the virtual infrastructure established within local institution. The virtualization allows to consolidate and utilize current available resources and may be the first step towards more powerful infrastructures for grid and cloud computing.

  The research infrastructures, e.g. Integrated Structural Biology Infrastructure for Europe (INSTRUCT)[2], European Life Science Infrastructure for Biological Information (ELIXIR)[3], European Biomedical Imaging Infrastructure (Euro-BioImaging)[4] and

[2] https://www.structuralbiology.eu/ accessed March 2015

[3] http://www.elixir-europe.org/ accessed March 2015

[4] http://www.eurobioimaging.eu/ accessed March 2015

others rely on grid-computing and cloud-computing infrastructures for science. The purpose of these initiatives is to understand high-level phenotypes from genomic, metabolomic, proteomic, imaging and other types of data. They also require multi-scale mathematical models and simulations, as noted e.g. by Hunter et al. [162] in his strategy for Virtual Physiological Human (VPH)[5]. The integration with multidimensional models of geometrical, mechanical properties and the time-dependence of the compartment's data, which is taken from medical and biological repositories, can highly improve complex models of human physiology which are based mainly on lumped-parameter approach. E.g. Itu et al. achieved parameter identification on simplified windkessel model of hemodynamics in order to study aortic coarctation, which is based on processing of MRI, and requires 6-8 minutes of computation time on a standard personal computer [163]. On of the challenge of systems biology approach, as identified by Kohl et al. [128], is to use multiparameter perturbation to identify the safe areas, e.g., for multitarget drug profile. The results presented in section 4.3 shows that the parameter study can be done on much more complex models in a reasonable time. The computation is able to become practical for clinical and further research towards patient-specific health care, in silico trails and drug discovery.

Additionally, it is a business strategy of several new ventures in order to collect anonymized patient records from clinicians, to gradually improve diagnostic methods and to provide reciprocal services for supporting clinical or therapeutic decision, as presented in section 4.2. For example, Fetview[6] is an startup company, which one aim is to support fetal healthcare and gradually improve diagnostic methods, which is based on already collected records in history.

Based on the previous answers, another research question can be formulated for further research in the technology domain:

How can biomedical research influence the direction of grid-computing and cloud-computing development?

One area of discussion about this theme is how to preserve scientific data in long term in order to prevent loss of them [164, 165]. Within the domain of computational physiology, this can be addressed in future, e.g., by the established repository of physiological values at `http://www.physiovalues.org`, keeping the values related to the model implementation and selected scenario. The challenges towards the grid and cloud could be the question about hosting of such type service and other specific requirements.

Another area of discussion is how to facilitate access to computational resources for large amounts of small scientific group, which have limited resources to port, integrate or customize their current tools and processes – to support the "long-tail" of science. The "long-tail" movement was first noted and described by Anderson [166] in the business

[5] `http://www.vph-institute.org/` accessed March 2015

[6] `http://fetview.com/` accessed March 2015

domain. The long-tail term comes from a feature of statistical distribution, e.g., pareto distribution, where only a few (e.g., 20% – noted as head) elements have a high probability of some events (e.g., product being sold), while the rest (e.g., 80% – noted as tail) have a small probability. Thus, most businesses focus on hits (20% of products, the 80-20 rule). The expansion of the Internet and its related technologies have caused reduced sales, marketing and delivery costs for the products from the niche (80% of products) – long-tail. A strategy that focused on these kinds of products became profitable and successful, e.g., for companies such as Amazon or Apple.[166].

Cloud computing technologies seem to be customizable and may be an enabling technology to focus on long-tail science, as noted e.g. by Weinhardt et al. [167]. How to facilitate and decrease an effort to develop, customize and port domain-specific application to some distributed computing model? This problem motivated, e.g., Anjum et al. to establish "platform as a service" (category of cloud computing service model) integrating several grid computing and cloud computing standards glueing via service oriented architecture approach [85]. Complementary approach is to support consultation, training and exchange in research software development toward the domain scientists, e.g., as presented by Crouch et al. regarding the Software Sustainable Institute within United Kingdom [168].

5.2 Summary

This thesis presents the infrastructure, which, thanks to virtualization technology, joined several domain-specific tools in the field of sharing and processing medical images, performing real-time voice analysis and simulating human physiology. Theoretical aspects of computational complexity and parallelism were discussed within use cases of specific biomedical research problems.

A seamless integration of grid-based PACS system was established with the current distributed system in order to share DICOM medical images. Access to real-time voice analysis application via remote desktop technology brings this type of service to any computer that can connect to the Internet. A system and portal to support the analysis and building of complex models of human physiology in the phase of parameter estimation and parameter sweep was introduced. Furthermore, additional computational nodes can be flexibly joined by starting the prepared virtual machines in cloud computing deployment. The methodology of building complex models of human physiology was contributed with the use of acausal and object-oriented modeling techniques. Methods for conducting a parameter study were shown, as well as the parameter study of complex models that gain substantial speedup by utilizing cloud computing deployment, which makes such kinds of complex studies applicable in physiological and biological research.

List of Abbreviations

API	Application Interface, page 11
BOINC	Berkeley Open Infrastructure for Network Computing, page 18
BPEL	Business Process Execution Language, page 21
CPU	Central Processing Unit, page 9
CRUD	Create Read Update Delete, page 15
CVS	Cardiovascular System, page 30
DFT	Discrete Fourier Transformation, page 27
DICOM	Digital Imaging and Communication Protocol, page 24
DWDM	Dense Wavelength Division Multiplexing, page 16
EGI	European Grid Infrastructure, page 17
FFT	Fast Fourier Transformation, page 27
FLOPS	Floating-Point Operations per Second, page 20
FMI	Functional Mockup Interface, page 33
FMU	Functional Mockup Unit, page 33
FTP	File Transfer Protocol, page 26
HPC	High Performance Computing, page 20
HTC	High Throughput Computing, page 20
HTML	HyperText Markup Language, page 14
HTTP	Hypertext Transfer Protocol, page 14
I/O	Input Output, page 19
IaaS	Infrastructure as a Service, page 20

LHC	Large Hadron Collider, page 18
MPI	Message Passing Interface, page 12
MTC	Many Task Computing, page 20
NGI	National Grid Initiative, page 17
OGSA	Open Grid Service Architecture, page 17
PaaS	Platform as a Service, page 20
PACS	Picture Archiving and Communication System, page 2
RDP	Remote Desktop Protocol, page 27
REST	Representation State Transfer, page 15
SaaS	Software as a Service, page 20
SOA	Service-oriented Architecture, page 14
SSL	Secure Sockets Layer, page 25
TCP/IP	Transmission Control Protocol/Internet Protocol, page 14
UML	Unified Modeling Language, page 5
URI	Uniform Resource Identifier, page 14
URL	Uniform Resource Locator, page 14
VPN	Virtual Private Network, page 25
VRP	Voice Range Profile, page 27
WLCG	Worldwide Large Hadron Collider Computing Grid, page 17
WWW	World Wide Web, page 14
XML	Extensible Markup Language, page 33

Appendix A

Processing of Medical Images in Virtual Distributed Environment

The paper [1] published as

T. Kulhánek and M. Šárek. Processing of medical images in virtual distributed environment. In *Proceedings of the 2009 Euro American Conference on Telematics and Information Systems New Opportunities to increase Digital Citizenship - EATIS '09*, pages 1–3. ACM, 2009. doi:10.1145/1551722.1551732

Available online at http://dx.doi.org/10.1145/1551722.1551732

The author of this thesis customized the existing project, Globus MEDICUS, and deployed it in servers that are networked via the academic network, CESNET, and integrated with the existing regional PACS system. Other co-author coordinated the work with selected hospitals and operators of the PACS system.

Processing of Medical Images in Virtual Distributed Environment.

Tomáš Kulhánek
CESNET, z.s.p.o
Zikova 4, 16000 Praha 6
Czech Republic
tomaton@centrum.cz

Milan Šárek
Academy of Sciences Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8
Czech Republic
sarek@euromise.cz

ABSTRACT

The processing of medical images within a PACS system depends on high capacity of communication channels and high performance of computational resources. We introduce pilot project utilizing grid technology to distribute functionality of PACS system to several machines located in distant places which allows economizing utilization of network channels. We also discuss benefits and disadvantages of virtualization techniques allowing to separate physical machine capabilities from the operating system. We compare this pilot project utilizing high speed CESNET 2 network with similar mature projects based mainly on P2P secure connection, centralized system and proprietary protocols.

Categories and Subject Descriptors

J.3 [**Medical information systems**]

General Terms

Management, Design

Keywords

Virtualization, Grid, PACS

1. INTRODUCTION

The Digital Imaging and Communications in Medicine (DICOM) standard is widely used in medical devices and applications. Picture archiving and Communication Systems (PACS) to archive DICOM are currently used in information systems within hospitals and today's effort is focused on connecting the systems among hospitals. The additional security and authorization mechanism must be kept with respect of data privacy and safety as DICOM itself doesn't provide such features [1]. DICOM series represents also usually large amount of data, which has specific requirements of capacity of communication channels.

Dostal et al. [2] introduced the client-server message brokering system with a centrally located server cluster and client application on user computer, the MeDiMed project. It was primarily used in national education network CESNET2; however other clients may connect via public Internet channels too. Client application retrieve DICOM series from the client's local or institutional PACS and send it via proprietary protocol using SSL encryption to server. Client application identifies the receiver and sets some other metadata regarding the message. The receiver must have the same client application and get the DICOM series from the server later. This solution based on the central point of the system architecture may become a bottleneck or single point of failure. There are other commercial solution using SSL encryption and authentication which are based on establishing VPN connection between peer endpoints.

Erberich et al. [3] utilized grid technology and open standards and protocols to process DICOM images securely in distributed environment to prevent some issues coming from VPN and proprietary protocols. They introduced project named Globus MEDICUS which integrates DICOM interface as a service of a grid infrastructure. Montagnat et al. [4] used similar approach in their Medical Data Manager which integrates grid middleware gLite with a DICOM interface providing strong security and encryption mechanism to preserve patient's privacy.

Different systems and technologies have different requirements on hardware and software environment. Virtualization techniques allow providing separation between software and underlying hardware. However virtualization introduces some overhead when translating isolated application instruction to lower level of a system. Current virtualization techniques allow full operating system isolation. Youseff et al. [5] showed that XEN paravirtualization doesn't impose an onerous performance penalty comparing to non-virtualized OS configuration.

We deployed the selected grid middleware and DICOM grid interface service from the Globus MEDICUS project to the virtual machines within the physical servers geographically spread throughout various institutions. We successfully exchanged the DICOM series between the DICOM grid interface and MeDiMed project without any proprietary modification of the systems used.

2. METHODS

The Globus Medicus project [3] provides a DICOM grid interface service (DGIS) able to communicate in DICOM standard, metacatalog service and storage service provider. Each service may run on separated host machine. The DGIS service is a bridge to grid infrastructure and hides the fact that the data are processed

EATIS'09, June 3–5, 2009, Prague, CZ.

throughout a grid. The metacatalog service and storage service provider are deployed on the Globus Toolkit.

We modified the DGIS to be able to communicate with the client application of MeDiMed project and accept the DICOM images exchanged in this project.

Because of specific requirements of the services of Globus MEDICUS, we chose to utilize virtualization techniques to fulfill the requirements dynamically. We installed the opensource XEN paravirtualization implementation, which adds a modification to the kernel of a guest system to be able to be executed and monitored by the host machine. Modification of the host system is, however, not required on hardware with virtualization support.

We installed the services of Globus Medicus within the virtual grid nodes on the paravirtualized guest systems Centos 5.2 Linux, kernel version 2.6 which are hosted on 64-bit Intel XEON running XEN 3.0.3.

DICOM standard uses separated direct IP connection to the user's location to send the results of the user's request. Because of that the DGIS service must have direct access to the user's application or DICOM device via IP transport level. So we decided to deploy DGIS service together with other services of Globus Medicus within the same guest system. The DGIS connects to the other local or remote services of the grid infrastructure via HTTP and gridFTP protocol. The communication between nodes and services is secured by asymmetric encryption and x.509 certificates.

3. RESULTS

We deployed nodes of the pilot grid infrastructure into the following pilot location: CESNET association, First Faculty of Medicine of Charles University and Central Military Hospital. All three locations are in Prague and are connected via high speed national educational and research network CESNET2 operated by CESNET association. We plan to use the pilot grid infrastructure also for another purpose and the XEN paravirtualization allows us to deploy and test another isolated projects next to this one.

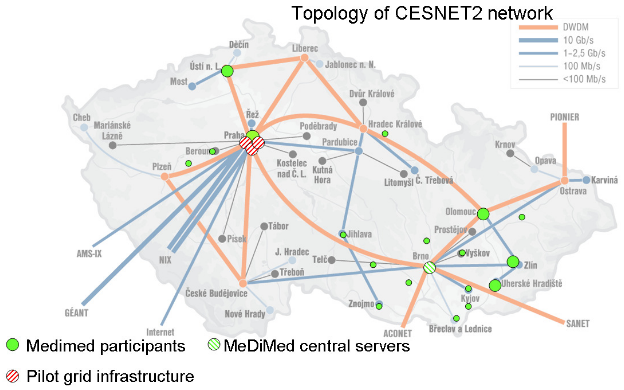


Figure 1. pilot grid infrastructure and MeDiMed project participants

We configured the guest virtual machines to share the same IP connection with the host system with IPv4 address. We configured the transport to virtual machine via network address translation (NAT) and we use Linux ipfiltering "iptables" ruleset to forward incoming connection to the grid services.

Some institutions hosting pilot project servers follow strict security policy, so they require the installation and execution of the grid services in demilitarized zone next to the institutional firewall with restricted access to local resources. With administrators of the institutional firewall we explicitly agreed and configured the firewall exception for the gridFTP protocol as the transport of such protocol uses TCP port number usually restricted by default.

We uploaded initial DICOM studies with about 1300 DICOM images for demonstration purposes. We successfully exchanged and processed the DICOM studies with the desktop application K-PACS.

We demonstrated that connection and DICOM studies exchange is possible between the MeDiMed project and the Globus Medicus. We used the client application of the MeDiMed project to retrieve and send selected DICOM series from the grid Globus MEDICUS to the participant connected in the MeDiMed project successfully and vice versa.

4. CONCLUSION

The grid technology is able to serve medical image processing in secure and reliable way as well as current systems. The only unsecured communication is between DGIS and DICOM compliant client, which is same for other types of solution (MeDiMed or VPN based) and is not usually recognized as security issue if unsecured connections are within trusted local network.

The grid services operate on specific TCP port numbers, the access to them was restricted by default in some institutions and explicit exception had to be implemented on the institutional firewall. Comparing to current production systems to share DICOM images (e.g. the MeDiMed project), they don't need such network configuration or they use VPN. The other problem regarded to network communication is sharing one IP connection among multiple virtual machines on the same host physical server. In such case we set the NAT and IP filtering rules statically on each physical server. Challenge for future development would be dynamic routing to virtual servers.

The DICOM grid service interface behaves as another DICOM compliant device and the whole system with the utilizing grid services may be considered as another PACS system e.g. as a remote backup or an external PACS for exchanging e.g. educational DICOM studies. In contrast the MeDiMed client's application doesn't allow to be controlled via DICOM protocol thus cannot be accessed by institutional application and the proprietary MeDiMed client application must be used to process DICOM studies from MeDiMed project.

The MeDiMed project will have to face problems of scalability and single point of failure. The grid technology and virtualization might be an answer to such problem for future enhancement and development as it can benefit from live network topology and doesn't need to maintain virtual topology established by VPN based solution. The MeDiMed client uses the proprietary protocol to communicate with server in contrast to the pilot grid infrastructure which is based on open standards.

Virtualization techniques allow dynamic allocation and management of physical resources. The pilot physical infrastructure of the servers might be utilized to deploy another virtual application or systems next to the DICOM and PACS services. This benefit is currently considered by the other participated institutions.

5. ACKNOWLEDGMENTS

Our thanks to CESNET z.s.p.o. and grant MSM6383917201.

6. REFERENCES

[1] Sarek, M. New Aspects of Pacs in Dwdm Network. World Congress on Medical Physics and Biomedical Engineering 2006 (2007), 417-420.

[2] Dostal, O., Javornik, M., and Ventruba, P. Collaborative Environment Supporting Research and Education in The Area of Medical Image Information. INTERNATIONAL JOURNAL OF COMPUTER ASSISTED RADIOLOGY AND SURGERY 1 (2006), 98.

[3] Erberich, S., Silverstein, J., Chervenak, A.,Schuler, R., Nelson, M., and Kesselman, C. Globus MEDICUS-Federation of DICOM Medical Imaging Devices into Healthcare Grids. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS 126 (2007), 269.

[4] Montagnat, J., Frohner, A., Jouvenot, D., Pera, C., Kunszt, P., Koblitz, B., Santos, N., Loomis, C., Texier, R., Lingrand, D., Guio, P., Rocha, R. B. D., de Almeida, A. S., and Farkas, Z. A Secure Grid Medical Data Manager Interfaced to The Glite Middleware. J. Gri Comput. 6, 1 (2008), 45-59.

[5] Youseff, L., Wolski, R., Gorda, B., and Krintz, C. Paravirtualization for Hpc Systems. Frontiers of High Performance Computing and Networking a ISPA 2006 Workshops (2006), 474-486.

Appendix B

Remote Analysis of Human Voice – Lossless Sound Recording Redirection

The paper [2] published as

T. Kulhánek, M. Frič, and M. Šárek. Remote Analysis of Human Voice – Lossless Sound Recording Redirection. *Analysis of Biomedical Signals and Images. Proceedings of 20th International EURASIP Conference (BIOSIGNAL)*, pages 394–397, 2010. URL: `http://bs2010.biosignal.cz/papers/1092.pdf`

Available online at bs2010.biosignal.cz/papers/1092.pdf.

The author of this thesis designed and customized the existing network protocol in order to transfer voice signal losslessly. The author also deployed the application on a remote virtual server. Other co-authors implemented the algorithms and application in order to analyze voice signal and coordinated the work.

Remote Analysis of Human Voice – Lossless Sound Recording Redirection

Tomáš Kulhánek [1,š], Marek Frič [2], Milan Šárek [1]
[1] CESNET z.s.p.o., [2] Music and Dance Faculty of the Academy of Performing Arts, [3] First Faculty of Medicine, Charles University
tomaton@centrum.cz

Abstract. This paper introduces a new approach to lossless sound recording redirection over remote desktop protocol. This approach is also demonstrated on an application in the field of Analysis of Human Voice used by phoniatric examination developed in the Musical Acoustics Research Centre of the Music and Dance Faculty of the Academy of Performing Arts. There was developed plug-ins for Remote Desktop Protocol which adds functionality to redirect sound recorded on the client side to the remote application without loss of information. Using these plug-ins the analytic application can be deployed on the remote server and accessed via RDP clients from different locations connected to the Internet.

1 Introduction

The human voice can be analyzed with computation methods during the phoniatric examination and a result of such analysis is produced into a voice range profile, also called as a phonetogram. It shows dynamic range of the human voice in terms of both fundamental frequency and intensity. The analytic application introduced in this paper ParVRP is being developed in the Musical Acoustics Research Centre of the Music and Dance Faculty of the Academy of Performing Arts. It implements computation methods in MATLAB environment and analyses the human voice stored in a sound file or recorded with a local microphone attached to the computer. The ParVRP application allows segmentation of the recording into separated voice events and produces phonetogram. [1]

There is being prepared the complex system which should allow usage of the ParVRP application from different location. This system is planned to support phoniatric examination in more physical locations is being prepared. As a pilot project, the analytic application is deployed into a virtual server which is part of the virtual infrastructure built in the past for another project exchanging medical images using data grid techniques [2]. The access to the analytic application is provided with remote desktop protocol (RDP) following the concept of thin client. Among other features, sound recording redirection from thin client to the remote application was introduced since RDP version 7.0 (server side since Windows Server 2008 R2 and client side since Windows XP SP3). The older version of RDP protocol 5.2 (since Windows Server 2003) needs external modification by third party product. Anyway, when testing the sound recording redirection in both versions, it was founded that the used codecs in both versions degraded the sound characteristics when transferred to remote application and thus this type of sound recording redirection is not acceptable for exact voice analysis [3].

2 Methods

There were discussed an option to use dedicated independent TCP/IP connection between client and server to redirect sound recording over it. This will bring some more configuration effort for the client and remote Internet providers, which may need to configure other communication channel behind the institutional firewall. Thus it was decided to reuse the existing RDP connection and customize virtual channels over RDP to transfer binary data.

The whole schema of this solution is shown on Figure 1. The system consists mainly with RDP plug-ins for client and server side. The main part of these plug-ins were developed in C# using basic .NET libraries. Together with the Mono project, these plug-ins can be deployed on the most used platforms which are Microsoft Windows and Linux-like operating systems [7]. The minor part of the plug-ins were developed with dependency on platform specific libraries for Microsoft Windows as well as for Linux operating systems.

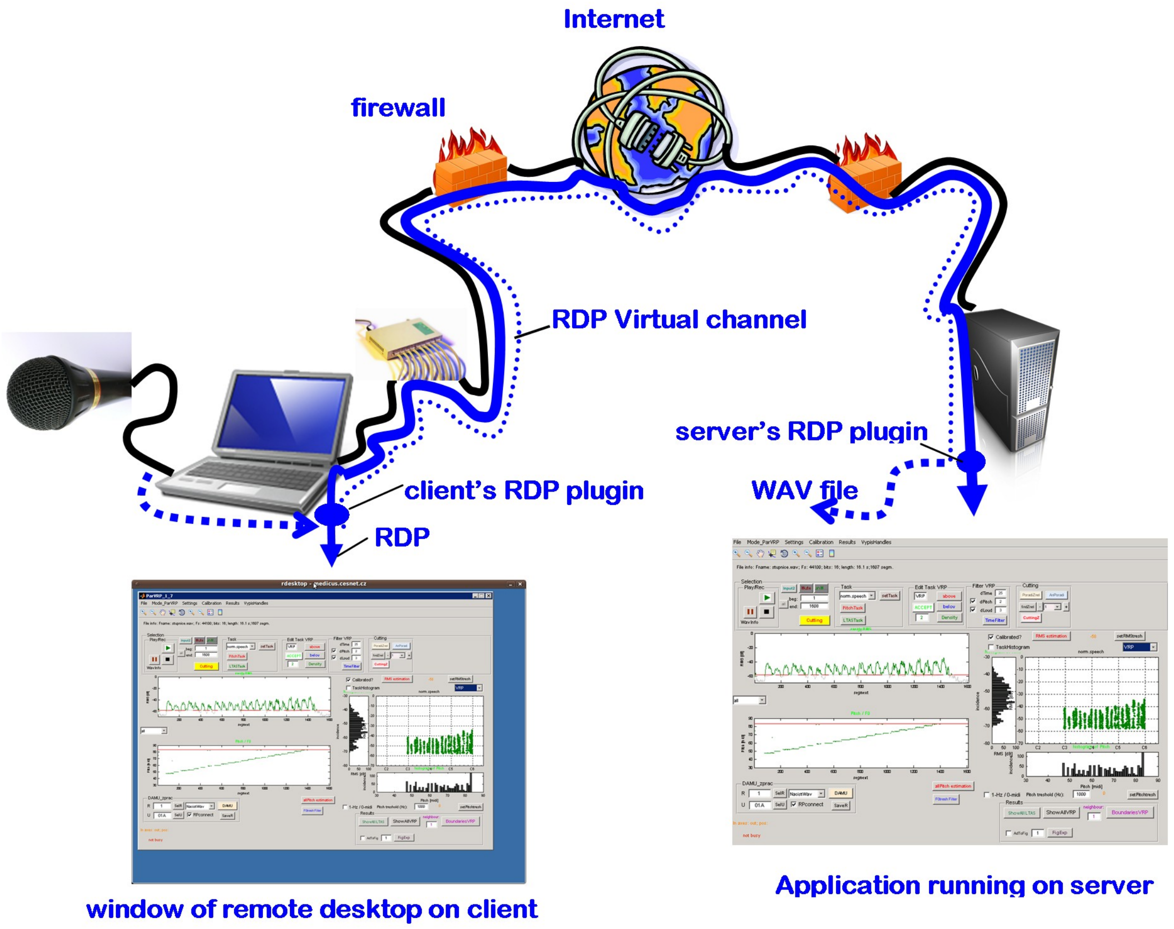


Fig 1. Schema of the sound recording redirection with the client's and server's RDP plug-in

On the remote side, the server's plug-in resides on MS Windows platform and opens a named virtual channel to the RDP connection using Remote Desktop API (Wtsapi32.dll). The plug-in then send messages via this virtual channel to control recording on the client. The received data from the virtual channel is written to a file. A proprietary API is provided for the ParVRP application which controls the recording.

On the local side, the client's plug-in to RDP receives message, starts and stops the sound recording from the system default sound-card input and sends binary data of the recorded sound via virtual channel. In case of Linux (or Unix-like) operating systems, the client's plug-in uses an external patch for the rdesktop [5] which is usually part of a common Linux distributions. It adds an option for rdesktop to redirect named RDP channel communication to standard input and output of a external executable plug-in [4]. The client's RDP plug-in listens standard input and utilize another command-line application arecord [6] also available in a common Linux distributions. Arecord records sound from system default sound card input and returns binary data in WAV format. The plug-in streams the binary data into the standard output which is then redirected by the patched rdesktop to RDP virtual channel.

In case of remote desktop client within Microsoft Windows operating system, the Remote Desktop API (Wtsapi32.dll) is used to register a client's plug-in for receiving events regarding the named virtual channel. However the plug-in in .NET is a so-called managed code which however needs to be called from an unmanaged system API, thus e.g. Selvin proposes automated solution which decompiles managed code from .NET DLL and after modifications it compiles it back into DLL which is then usable from unmanaged API calls[8]. The The WINMM multimedia API (winmm.dll) is used to record sound from the system default sound card input into data stream which is then written to the RDP virtual channel.

3 Results

Using the client's and server's RDP plug-in, there can be transferred audio data via the named RDP virtual channel without any loss of information, which is the main lack of the existing solution of sound recording redirections available for RDP version 5.2 and RDP version 7.0. The plug-ins redirect the binary data in a stream. However the binary stream is transferred in uncompressed WAV format with 44.1 kHz sampling rate and 16 bits per sample thus the bitrate of such recording is 705,6 kbit/s which needs to be transferred via the network. The overhead of base RDP protocol, which uploads events from client's keyboard and mouse to remote application is minimal. This bitrate doesn't cause any problems on the server deployed on the CESNET2 network, where the network bandwidth is above 1 Gbit/s. The bandwidth on the client's side location might be limitation if it is less than 1Mbit/s.

In contrast, the average bitrate of sound recording redirection of RDP 7.0 or Sound over RDP in RDP 5.2 is only 80 kbit/s thus this solution are not demanding on the network bandwith on client's side.

Protocol	**Upload to remote application**	**Type of sound transfer**
RDP v 5.2 + Sound Over RDP	80 kbit/s	Lossy
RDP v 7	80 kbit/s	Lossy
RDP v 5.2 + plugins for WAV via virt.channel	705,6 kbit/s	Lossless

Tab 1. Comparison of upload rate and type of sound recording redirection

The RDP plug-ins are distributed as DLL libraries which can be integrated into server's Windows platform. Proprietary .NET API is provided for ParVRP application developed in MATLAB. On the client side it can be integrated into general RDP clients on both Linux and Windows platforms.

4 Conclusions

In contrast with sound recording redirection of RDP 7.0 or Sound over RDP in RDP 5.2, the introduced redirection of sound recording over RDP doesn't provide a virtual sound card which may be used by a general remote application. This breaks a so-called “transparency” and any application like ParVRP needs modification to use the proprietary API provided by the RDP plug-ins.

Anyway the introduced sound recording redirection is acceptable for planned deployment in production environment to support phoniatric examination and provide high quality on-line analysis with a planned possibility of remote corroboration among more specialists.

The introduced approach of lossless sound recording redirecting over RDP to the remote application is also 10 times demanding for transfer upload rate than the mentioned options using lossy codecs, however still far bellow the bandwidth provided by CESNET2 network, which will generally connect the collaborating workplaces.

Comparing to the mentioned dedicated TCP/IP channel, which can be used to transfer sound recording instead of the introduced virtual channels in RDP, there would be needed to solve authentication and encryption of the communication together with additional effort needed to configure institutional firewalls to allow this channel. RDP protocol provides already authentication and encryption mechanism thus the RDP plug-ins don't need to solve that issues.

Acknowledgement

This work has been supported by the grant projects 1M0531 Musical acoustics research centre and by the research plan Optical National Research Network and its New Application, in association CESNET z.s.p.o.

References

[1] Frič M., Parametrizovaný fonetogram obecných řečových a hlasových projevů –ParVRP, MARC-Technologický list čís. 12, Zvukové studio HAMU, Praha 2007, http://web.hamu.cz/zvuk/vyzkum/dokumenty/TL12x.pdf

[2] Kulhánek, T. and Šárek, M. 2009. Processing of medical images in virtual distributed environment. In Proceedings of the 2009 EATIS'09. ACM, New York, NY, 1-3. DOI= http://doi.acm.org/10.1145/1551722.1551732

[3] Kulhánek T., Frič M., Šárek M.: Remote Access to Virtual Education and Research Application, MEFANET 2009 proceeding, Brno, ISBN: 978-80-7392-118-7, http://www.mefanet.cz/res/file/mefanet2009/prispevky/kulhanek_full.pdf

[4] Yakimenko S.,“Writing plugins for Rdesktop“, The Code Project, Apriorit Inc, 1.12.2009, web, http://www.codeproject.com/KB/cross-platform/writing-plugins-forrd.aspx

[5] rdesktop: A Remote Desktop Protocol Client for accessing Windows Terminal Services, web: http://www.rdesktop.org

[6] Arecord, command line wav recorder, web: http://alsa.opensrc.org/index.php/Arecord

[7] mono, cross platform, open source .NET development framework, web: http://www.mono-project.com/Main_Page

[8] Selvin, How to Write a Terminal Services Add-in in Pure C#, The Code Project, 14.11.2006, web: http://www.codeproject.com/KB/system/TSAddinInCS.aspx

Appendix C

Infrastructure for Data Storage and Computation in Biomedical Research

The paper [3] published as

T. Kulhánek. Infrastructure for Data Storage and Computation in Biomedical Research. *European Journal for Biomedical Informatics (EJBI)*, 6(1):55–58, 2010. URL: http://www.ejbi.org/img/ejbi/ejbi2010-1.pdf

Available online at www.ejbi.org

The author of this thesis proposed the idea of consolidating and sharing physical resources in order to provide a virtual environment for the specific needs of particular use-cases. The pilot infrastructure was tested on examples of selected research projects.

Infrastructure for Data Storage and Computation in Biomedical Research

T. Kulhánek[1]

[1]Institute of Pathophysiology, 1st Faculty of Medicine of Charles University, Laboratory of Biocybernetics and Computer Aided Teaching, Prague, Czech Republic

Supervisor: Ing. Milan Šárek, Csc.

Summary

Infrastructure as a service (infrastructure which is offered to customer in the form of service of the provider) is a deployment model which allows utilize data and computing capacity of a cloud as a set of virtual devices and virtualized machines. Infrastructure as a service can be offered separately to each project. The same capacity of connected physical machines and devices can be shared. Currently, the concept of an Infrastructure as a service is tested on several projects within activity of the CESNET association, First Faculty of Medicine, Charles University, Prague and Musical and Dance Faculty of Academy of Performing Arts in Prague.

The current research in the field of computation physiology is demanding on a high computation capacity. The computation tasks are distributed to computers, which are provided by the infrastructure. The project in the field of the analysis of a human voice is demanding on high throughput of a computer network between an acoustic or video device on the local side and an analytic application on the remote high performance server side. This paper describes features and main challenges for infrastructure dedicated for such a type of an application. Infrastructure as a deployment model of cloud computing might be beneficial for a multi domain team and for collaboration and integration of a high specialized software application.

Keywords: cloud computing, infrastructure as a service, virtualization, computation physiology, identification of physiological systems, validation of physiological models, remote desktop protocol, grid computing, voice range profile

1. Introduction

Several tasks can be found in the field of an application dedicated to support biomedical research of the current distributed computing systems. The main tasks cover exchanging, storing and retrieving data. The other task is to support the analysis of data and allow long lasting parallel computation. The requirement to keep privacy of patients data is the important feature of these systems and thus it must be ensured that only authorized users may access to application and data. The high level of security is a must, or an appropriate anonymization should be implemented.

Systems focused on data exchange among different organizations try to optimize data flow via the computer network, they encrypt the data which are sent via the network, ensure the required level of reliability and integrates several incompatible systems.

The first example of distributed systems are PACS (Picture Archiving and Communication Systems). DICOM (Digital Imaging and Communications in Medicine) is the most often used standard of format and protocol to exchange medical imaging information in the field of radiology. The security of transfered images is kept on other levels of the system. PACS systems are built upon the DICOM and solves the storage and maintenance of medical images. These systems are mainly deployed and closed within a hospital or within a network of hospitals maintained by the same owner. There became a requirement to join these PACS systems from different locations. Although the DICOM protocol is used, PACS systems have proprietary implementation of management and maintenance of DICOM images and there appeared an issue caused by incompatibilities of PACS systems.[1]. There were introduced systems for exchanging DICOM, which followed the classical structure of central storage and distributed user access (e.g. MeDiMed)[2]. There were introduced systems built as a communication centers with ability to send data among the different PACS systems (e.g. ePACS[3] or ReDiMed).

The project R-Bay was the another example of distributed systems for medicine. There were researched the possibilities how to join general systems, including exchange of DICOM images among institutions from different European countries. There was researched also the ability to provide and consume services of radiologists remotely on an international level. [4]

The systems which use computation and/or data grid is the other example of the distributed system in medicine. The project Globus Medicus is built upon the grid middleware Globus Toolkit. It provides services which presents a the DICOM interface to the user. The usage of grid middleware brings some beneficial features like reliability, security and effective transfer of data [5].

The system built within the project FONIATR is an example of the system demanding on the data transfer rate. This system supports a phoniatric examination and provides an application for the analysis of the human voice over the remote desktop protocol (RDP) in the MS Windows platform [6].

The system built within the project IDENTIFIKACE is an example of the computation system. Computational models of human physiology are developed in the MATLAB/Simulink environment and the new models also in the Modelica language [7]. Current models cover the whole complex functional parts of human physiology and reuse published relations and schema [8]. The models are validated against the data measured on patients within the project work, this is so-called identification of physiological systems. Some of the model parameters cannot be measured thus they are estimated by optimization techniques, which are demanding on performance and take a lot of a computational time. The estimation of parameters may take several hours or days on some more complex models. There are developed techniques to parallelize the computation and distribute the computation tasks to desktop computers in the laboratory or computers in the academic grid centers [9].

It's possible to built an own infrastructure for each of the previously mentioned systems. This may be, however, demanding on money, time and human resourcess when purchasing, installing and configuring all the needed servers and devices.

2. Methods

The infrastructure provided in an academic environment in the form of shared computational and data clusters is one of the way how to streamline the process of building up the computational and data resources. This possibility is offered by a national or international computational and data grid. Some of the projects may require a specific environment or a specific version of software library, which is not present in general grid systems. This requirement might be solved by using virtualization and cloud computing.

Computation and data grid

The computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [10]. Data grid may be characterized as an integrating architecture to do an efficient and reliable execution of data queries, which requires careful management of terabyte caches, gigabit per second data transfer over wide area networks, coscheduling of data transfer and supercomputer computation, accurate performance estimations to guide the selection of data replicas and other other advanced techniques that collectively maximize use of scarce storage, networking and computing resources [11].

It needs an additional effort to administer and maintain the grid infrastructure. This task is typically provided by a national grid initiative and the grid infrastructure is shared among different independent users. The national grid initiative was established in the Czech Republic and is maintained by the METACENTRUM activity which is one activity of the NREN (national research and educational network) provider, the CESNET association [12] and coordinates also the work with NGI from neighboring countries in the European Grid Initiative (EGI).

The system needs to be customized and use the API provided by the grid middleware Globus Toolkit, gLite and others.

Desktop grid

Anybody (from the academic and research community) can join and use the grid built by the NGI. Usually only the provider of NGI maintains and enlarges this infrastructure and grid middleware.

There exists different concepts how to built the grid. Everybody can enlarge the grid infrastructure by joining their computer and decides which application can use their computer for computation. The most well known example of such a grid system is SETI@home [13]. The concept of SETI@home follows the idea that anyone connected to Internet can join a grid by downloading a small client program and execute it in the background which periodically asks for computational jobs and computes in the background or as a screen-saver. The grid nodes are typically PCs owned by individuals. Such systems are usually referred as Volunteer grid systems or Desktop Grids and a general desktop grid system BOINC is used to build such systems, customized server site and client application to form a custom desktop grid application[14], [15].

Cloud computing

Virtualization is a technology which provides separation between a software layer and an underlying hardware layer. It allows execution of one of more so-called virtual machines sharing one physical hardware. The virtual machine is fully or partly (paravirtualization) separated from the physical layer of the hardware and thus different platforms (Windows, Linux) may work together on one physical machine concurrently. Virtualization techniques introduce some overhead when translating an isolated application instruction to the lower level of a system. However, the open source paravirtualization system XEN does not impose an onerous performance penalty comparing to non-virtualized operating system configuration [17].

The virtualization is sometimes characterized as a key technology which enables cloud computing and execution of different isolated systems on shared hardware.

Virtual infrastructure

Virtual organization is a group of users, who share the same resources. The virtual infrastructure belonging to a virtual organization is built from virtual machines connected via the network, which may be virtual too and accessible only to users from the virtual organization. Figure 1 shows an example of several virtual organizations and their infrastructures. On the right part there is a schematic view on physical connections among different organizations (hospitals, research institutions) via the academic network or the Internet. The physical resources are shown as vertexes and network connections are shown as edges. Each cloud shows one virtual infrastructure. On the left part there is a physical server executing more virtual machines, each machine belongs to a different virtual infrastructure.

3. Results

The pilot infrastructure dedicated for the medical application was established within the CESNET's activity "Application support" in several locations in Prague. It utilizes the open-source virtualization system XEN and tools of the operating system Linux for configuring virtual machines and virtual infrastructure as a service.

The instance of the grid system Globus MEDICUS was the primary system deployed to a set of virtual machines. It was shown that the grid system based on open standards can be easily integrated with current medical systems using the DICOM format [18].

The application to support a phoniatric examination was deployed next to the previous system to exchange medical images. It was needed to develop an enhancement of the RDP protocol, which adds the transfer of an audio signal from local computer's microphone to the remote application [6]. This system is currently deployed on one virtual machine. See Figure 2.

The pilot infrastructure contains also the system for identification of physiological systems, which offers a web service distributing the computational task to desktop computers connected via the desktop grid system BOINC and SZTAKI Desktop Grid API [9]. The schema in Figure 3 shows the architecture of the system. The server is in operation as an independent virtual machine and contains the web service. Some of the BOINC workers are in operation as independent virtual machines deployed on less used physical servers of the pilot infrastructure. Some of the desktop computers of laboratory and classroom of the First Faculty of Medicine are connected to this desktop grid system. Other computers may be easily joined. Current research is focused on the possibility to enhance the computational capacity of the infrastructure by the resources provided by the NGI MetaCentrum. There is also researched an utilization of GPU computing.

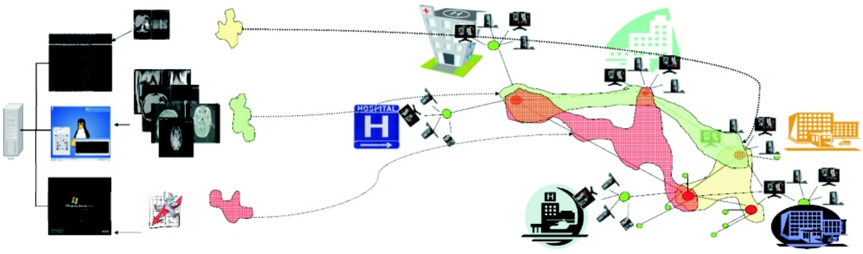


Fig. 1. General schema of virtual infrastructure.

All the physical servers of the pilot infrastructure have a virtual environment built upon the XEN system and they share the IP network addresses. Each virtual machine has its own disk partition within the Logical Volume Management (LVM) on the physical server. The virtual machines are administered by the tool virt-manager and the network environment is configured with the tool iptables.

4. Discussion

The pilot infrastructure can be characterized as a private cloud, which is accessible only to the limited community of users from the field of biomedical research. Virtual machines share the physical network connection via IPv4. Because of the lack of numbers of unique IPv4 addresses, the configuration of network services (webserver, RDP) is realized using network address translation and port mapping. If the network devices passed to the version 6 of IP protocol, there would be opened again the possibility to provide unique IP addresses to virtual machines and there would not be needed extra configurations of network address translation and port mapping.

There are not used special tools to administer cloud within the pilot infrastructure, because the number of projects is relatively small currently. Anyway, there exist free or commercial products (Eucalyptus, OpenNebula, VMWare vSphere), which provide a set of tools to automatize the maintenance of private cloud, including virtual network configuration, live migration of virtual

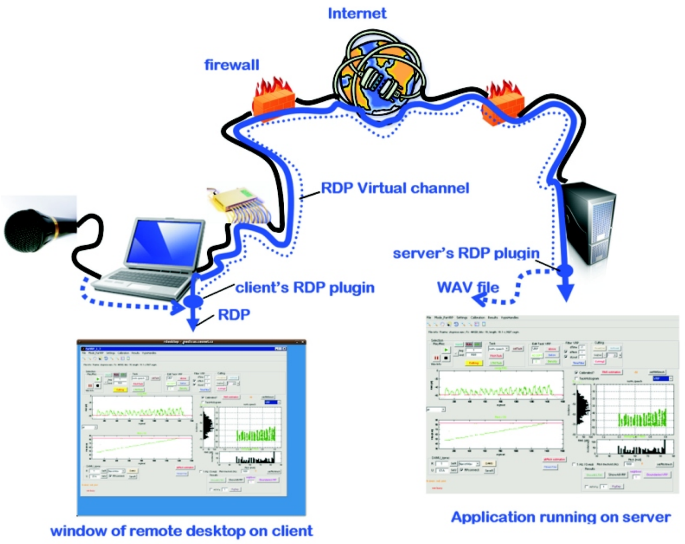


Fig. 2. Schema of system for human voice analysis and remote recording via RDP protocol.

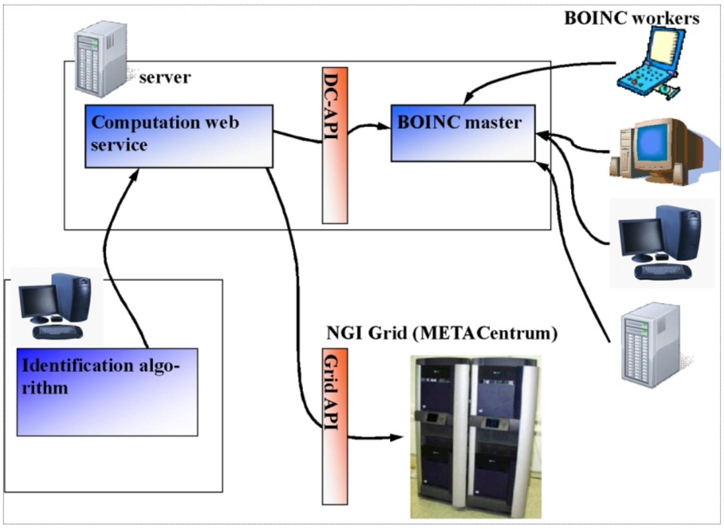


Fig. 3. Schema of computational infrastructure for identification of physiological systems.

machine, etc. Deploying these tools will be necessary in future after expected enhancement of the physical capacity, which is planned to be built within the academic environment of the Czech Republic.

The important question is: which type of the application is suitable for clouds operating on physical resources spread in different geographical locations compared to clouds operating in supercomputing centers. Clouds in supercomputing centers are suitable for highly parallel tasks which need fast communication between parallel computational tasks. Cloud operating on physical servers in different geographical locations can offer a free capacity in the time period, when the owner does not utilize its physical resources and offers them to other users of cloud.

5. Conclusion

It is possible to operate a private cloud on the physical infrastructure and to provide the virtual infrastructure to the users, who can utilize it to execute their own applications and systems. The infrastructure as a service can open an access to distributed systems to a higher amount of users, who have been so far prevented from using them by a complicated administration, too long process of purchasing and installing computing resources.

The cloud operating on physical servers in different geographical locations can be a suitable complement to the clouds in supercomputing centers.

Acknowledgment

The work was supported by the grant SVV-2010-265 513.

References

[1] KulhánekT., Šárek M.: Virtualizace a integrace v gridovém PACS systému, MEFANET 2008, Brno, ISBN 978-80-7392-065-4

[2] Dostal, O., Javornik, M., and Ventruba, P.. “Collaborative environment supporting research and education in the area of medical image information”. Internation Journal of Computer Assisted Radiology and Surgery. Sringer Science+ Business Media. 1. 98. 2006.

[3] E-PACS, DICOM komunikace mezi zdravotnickými zařízeními, 2010, czech language, web: http://www.epacs.cz/

[4] Pedersen C. D., Rasmussen J., Beck C.: R-Bay; Creating an eMarketplace for the transfer of imaging related eHealth services in Europe. Final project report, version 2, web: http://www.r-bay.org

[5] Erberich S.G., Silverstein J.C., Chervenak, A., Schuler, R., Nelson, M.D., and Kesselman, C.: “Globus MEDICUS-Federation of DICOM Medical Imaging Devices into Healthcare Grids”. Studies in Health Technology and Informatics. IOS Press; 1999. 126. 269. 2007.

[6] Kulhánek T., Frič M., Šárek M.: Remote Analysis of Human Voice Lossless Sound Recording Redirection,Proceedings of Biosignal 2010: Analysis of Biomedical Signals and Images, ISSN 1211-412X, ISBN 978-80-214-4106-4

[7] Stodulka, P., Privitzer, P., Kofránek, J., Tribula, M., and Vacek, O.. Development of Web Accesible Medical Educational Simulators”. Proceedings of EUROSIM 2007, Ljubljana. 2. 16.

[8] Abram S.R., Hodnett B. L., Summers R.L., Coleman T.G., Hester R.L.: Quantitative circulatory physiology: an integrative mathematical model of human physiology for medical education. Adv. Physiol. Educ., 31, 202-210, (2007).

[9] Kulhánek T., Šilar J., Mateják M., Privitzer P., Kofránek J., Tribula M.: Distributed computation and parameter estimation in identification of physiological systems, VPH conference 2010, Brussels

[10] Foster I., Kesselman C. “Computational grids”, Vector and Parallel Processing, VECPAR 2000, 2001 Springer

[11] Chervenak A., Foster I., Kesselman C., Salisbury C., Tuecke S.: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientic Datasets. Journal of Network and Computer Applications, Elsevier, 2000

[12] Šustr Z., Sitera J, Mulač M, Ruda M., Antoš D., Hejtmánek L., Holub P., Salvet Z., Matyska L.: MetaCentrum, the Czech Virtualized NGI, 2009, web: http://egee.cesnet.cz

[13] Anderson D. P., Cobb J., Korpela E., Lebofsky M., and Werthimer, D. 2002. SETI@home: an experiment in public-resource computing. Commun. ACM 45, 11 (Nov. 2002), 56-61.

[14] Anderson D. P. 2004. BOINC: A System for Public-Resource Computing and Storage. In Proceedings of the 5th IEEE/ACM international Workshop on Grid Computing (November 08 - 08, 2004). International Conference on Grid Computing. IEEE Computer Society, Washington, DC, 4-10.

[15] Balaton Z., Gombás G., Kacsuk P., Kornafeld A., Kovács J., Marosi C.A., Vida G., Podhorszki N., Kiss T.: SZTAKI Desktop Grid: a Modular and Scalable Way of Building Large Computing Grids. Proceedings of the 21st International Parallel and Distributed Processing Symposium (IPDPS 2007), 26-30 March 2007, Long Beach, California, USA

[16] Velte T., Velte A., and Elsenpeter R.: Cloud Computing: A Practical Approach". McGraw-Hill Osborne Media. 2009.

[17] Youseff L., Wolski R., Gorda B., Krintz, C. Paravirtualization for HPC Systems. Frontiers of High Performance Computing and Networking â€" ISPA 2006 Workshops. 474486. 2006.

[18] Kulhánek T., Šárek M.: Processing of medical images in virtual distributed environment. ACM. 13. 2009.

Contact
Mgr. Tomáš Kulhánek
Institute of Pathophysiology,
First Faculty of Medicine of Charles University
Laboratory of Biocybernetics and Computer Aided Teaching
U Nemocnice 5
128 53 Prague 2
Czech Republic
e-mail: tomaton@centrum.cz

Appendix D

Parameter Estimation of Complex Mathematical Models of Human Physiology Using Remote Simulation Distributed in Scientific Cloud

The paper [4] published as

T. Kulhánek, M. Mateják, J. Šilar, and J. Kofránek. Parameter Estimation of Complex Mathematical Models of Human Physiology Using Remote Simulation Distributed in Scientific Cloud. In *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, pages 712–715, 2014. doi:10.1109/BHI.2014.6864463

Available online at ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6864463

The author of this thesis designed architecture for a distributed parameter estimation, integrated models and implemented a pilot deployment, which utilized a scientific cloud computing infrastructure. Other co-authors implemented complex models of the human physiology in Modelica language and tested several algorithms for parameter estimation.

Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud

Tomáš Kulhánek*,1, Marek Mateják1, Jan Šilar1
and Jiří Kofránek1

Abstract— **A generic system for estimation of model parameters —calibrate models— is introduced. The proposed system architecture is built of several loosely coupled modules behaving as RESTful web services and allowing to integrate other parts of the system via HTTP protocol and data exchanged in JSON format. The system was designed in such a way that the most demanding computational part is computed in parallel and computation may be distributed to remote computational resources. A test deployment was done in scientific cloud provided by czech NGI CESNET. Parameter identification of complex models got significant speedup on cloud computing resources.**

I. INTRODUCTION

There are several methodologies and technologies how to model a complex reality in biological domain. The large-scale mathematical description of physiological systems was introduced by Guyton et al. in 1972[1] and continues today by Hester et al. who introduced HumMod - a derivative of the Guyton's model and in-house modeling language and simulation tool[2][3].

Kofranek et al. implemented Guyton's 1972 model in MATLAB®Simulink[4]. However, the complexity of the model increased from Guyton's model to HumMod and it becomes too complicated keeping the model up-to-date using block oriented tools like MATLAB®Simulink. Therefore an implementation of the HumMod model was introduced in object oriented Modelica language[5][6]. Modelica is an acausal object oriented language introduced by Fritzson et al.[7] and it is curently mantained by the international Modelica association and implemented by several vendors[8].

The parameter identification is a task to estimate the unknown model parameters in such a way that the model simulation fits the experimental data[9]. The objective of this task could be for example to minimize the following function (least squares):

$$f(\vec{p}) = \sum_{i=1}^{n}(M(t_i, \vec{p}) - d(t_i))^2 \rightarrow min \qquad (1)$$

where $\vec{p}$ is vector of values of parameters, $M(t_i, \vec{p})$ is model simulated at time t_i with the given parameter values $\vec{p}$ and $d(t_i)$ is the measured experimental value at time t_i.

The models of human physiology are in general set of linear and non-linear algebraic and differential equations, some of them may change it's behavior based on discrete conditions, thus output of such model can be non-differentiable and non-continuous. Thus global optimization methods that work without derivatives has to be used in general to find minimum of the objective function. There were tested several global optimization methods to identify parameters of models in Modelica language[10][11]. The simulation must be performed many times using this methods and it may take extremely long for complex models or larger space of parameters. The computation time can be however reduced using distributed computing techniques. Maffioletti et al. introduced GC3Pie framework and shown workflow to identify parameters using grid computing[12]. Humphrey et al. calibrated hydrology models utilizing cloud computing [13]. We proposed a system which should support the process of parameter identification mentioned in above scientific publication in more generic way, so the researcher may focus on experimental data, selecting the parameters from a model to identify and interpreting the estimates during computation and hide the technical details of configuring the computational modules in distributed systems.

The proposed system integrates visualization, identification algorithm and simulation into loosely coupled modules opened to any modeling and numerical technology. We implemented the system and tested it with the models of human physiology in Modelica language, we selected genetic algorithm as a global optimization method for parameter identification and we distributed simulation into scientific cloud.

II. SYSTEM DESCRIPTION

The proposed system as seen on Figure 1 consists of several loosely coupled distinct pieces of software modules which can be replaced by another technology or implementation with no or minimal intervention into other related modules. The communication among modules is done via HTTP protocol and endpoints follows REST architectural style[14].

The simulation module consist of Modelica model exported as a Functional Mockup Unit (FMU) conforming the standardized Functional Mockup Interface[16]. The FMU is in fact DLL library for MS Windows platform. We wrapped this FMU by the ServiceStack [17] framework to provide web interface and control the simulation via HTTP protocol and JSON format. The simulation module can be deployed in multiple instances and each one can be executed in parallel.

*corresponding author tomas.kulhanek@lf1.cuni.cz
1Institute of Pathological Physiology, First Faculty of Medicine, Charles University in Prague, Czech Republic

978-1-4799-2131-7/14/$31.00 ©2014 IEEE

TABLE I

TIME AND SPEEDUP OF ESTIMATING THE PARAMETERS OF MODELS.

CPUs involved	1	2	3	4	10	20	30	40	100
model HumMod[6]									
time	71d21h	38d1h	25d11h	20d8h	7d16h	3d21h	2d10h	1d20h	18h
speedup	–	1.8x	2.8x	3.5x	9.3x	18.5x	29.7x	39.2x	95.8x
model Rossi-Bernardi[15]									
time	50 min	27 min	20 min	19 min					
speedup	–	1.8x	2.5x	2.6x					

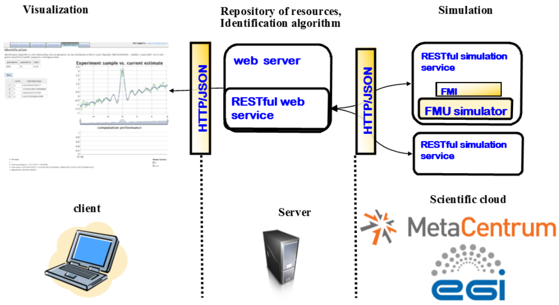


Fig. 1. Architecture of the system for parameter estimation

The simulation module instance registers it's unique endpoint URL to the repository module.

The repository module is a RESTful web service which provides storage of several data entities and provides access to the data via HTTP protocol and JSON format. The parameters are presented in web application as a table of names and values, the experiment and simulation data are visualized in a graph.

The identification algorithm is executed within this module and is used for model parameter estimation. We selected genetic algorithm from MATLAB-Optimization toolbox exported into a DLL library able to be executed with Matlab Common Runtime (MCR) environment. Each step of the algorithm produces a vector of parameter values which needs to be simulated and can be computed parallely by the instances of simulation modules.

The visualization module is implemented in HTML (version 5) and Javascript. This module gives user a list of available models. The table for experimental data allows to insert manualy or to copy&paste data from a desktop application. The table for parameters defines names, initial value and estimated maximal and minimal value, which will be taken into account by identification algorithm. After starting the identification process, the current best estimation is visualized periodically in a graph together with experimental data. From this perspective the identification algorithm behaves as a curve fitting process.

The whole system can be deployed in single computer, however it was designed to be deployed in several different computing elements. We deployed simulation modules in a local cluster and in a virtual infrastructure within scientific cloud provided by the Czech grid infrastructure provider CESNET[1], member of the European Grid Infrastructure foundation (EGI[2]). The repository module and identification algorithm controls the connected simulation modules.

III. RESULTS

For testing purposes we selected one known parameter from the HumMod model[5] and identify it again. We also tested to identify 4 parameters of model of hemoglobin saturation curve in variable condition of acidity and concentration of carbon dioxid based on the model of Rossi-Bernardi1967[15] implemented in Modelica language. We set the genetic algorithm to finish after 200 thousands single simulations for both models giving the best result found during the computation. The simulation was distributed into local cluster (up to 4 parallel simulation processes, CPU Intel XEON 2.7GHz)) and in scientific cloud (up to 100 parallel simulation processes, CPU Intel E5-2620 2GHz).

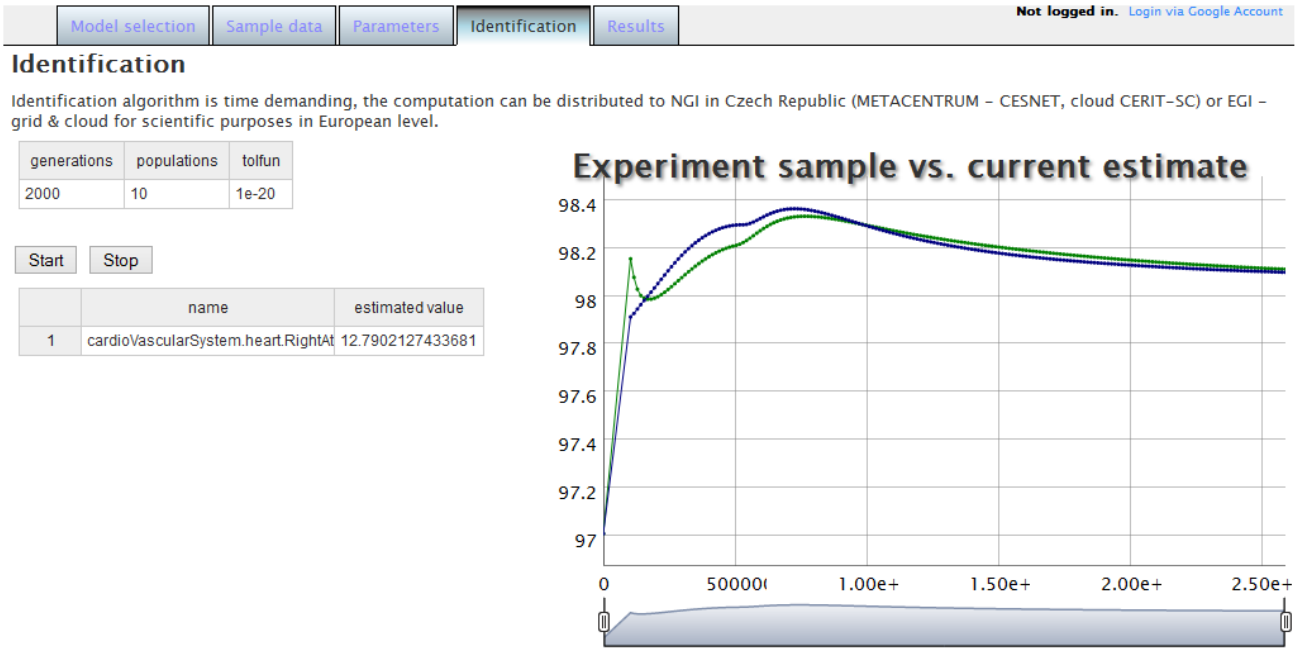
Experimental data for HumMod model were generated from single simulation of the model with specified parameter. In the case of the Rossi-Bernardi1967 model, we took the experimental data from the publication[15]. The values of parameters identified during the computation were comparable with the known values. However, we focus on the time of computation and possible speedup when the simulation was distributed into more parallel processes. The measured computation time and speedup is in the table I.

The single simulation of HumMod model takes about 30s of computation time. And we estimate the whole process of computation to 71 days. We didn't wait more than 2 months for this results, rather we estimate this after couple of hours from the number of the simulation done. When distributed into the local cluster up to the 4 CPU, we got the speedup about 3.5 times. When the computation was distributed into virtual infrastructure of 10 computers each contributing by 10 CPU with computation (totally 100CPU) we got the speedup about 96x and the estimation of 1 parameter was done in 18 hours.

The single simulation of the Rossi-Bernardini1967 model takes about 15miliseconds. When computation was distributed into 3 parallel nodes (3 CPU) the utilization of the service module was high and adding another computation node (totally 4 CPU) we did not get any other significant speedup. Distributing the computation to the scientific cloud we got even worse results influenced mainly by the network latency and increased communication overhead.

[1] http://www.cesnet.cz

[2] http://www.egi.eu



Identification

Identification algorithm is time demanding, the computation can be distributed to NGI in Czech Republic (METACENTRUM - CESNET, cloud CERIT-SC) or EGI - grid & cloud for scientific purposes in European level.

generations	populations	tolfun
2000	10	1e-20

	name	estimated value
1	cardioVascularSystem.heart.RightAt	12.7902127433681

Fig. 2. Screenshot showing running estimation. Current best estimation of parameters values on the left table and curve of estimated model data (blue) vs. experimental data (green).

IV. DISCUSSION

The parameter identification of the complex model spends majority of it's time in parallel simulation and minority in communication and synchronization procedures. This type of tasks can achieve significant speedup if the computation is distributed into remote capacity e.g. within a cloud or grid. On the other hand, the parameter identification of simpler models converges to a highly parallel computation system where time spent in parallel simulation is moreless same as in communication and synchronization procedures.

We estimate that if the single simulation takes more than a second, then the identification task is worth to deploy into cloud computing environment and gain a speedup from it. However if the single simulation takes less for simpler models, than the identification task should stay on local cluster or should be computed in some supercomputer. Distributing them into cloud or grid using this system we do not get any significant speedup. However more exact distinction should be done in further studies.

The parameter estimation within this paper was provided by genetic algorithm, however, there are other identification algorithm (e.g. other evolutionary algorithms) which can gain significant speedup utilizing distributed computing environment.

The system was tested with models implemented in Modelica language, however, significant contribution to the knowledge of human physiology were done by other projects, e.g. VPH[18] or IUPS Physiome[19], which has a so called Physiome model repository[20] and the majority of models are in CellML modeling language or JSIM modeling language. There is an effort to develop translation tool among the modeling technologies to give researchers freedom of choosing the modeling technology e.g.[21]. The further development of the system for parameter estimation can be enhanced to support and simulate the models in the above mentioned modeling technologies.

The web application provides minimal set of functionality for system analysis. Further development of the web application and introduced system needs to do an usability survey and incorporate most useful functionality. For other methods and tasks related to system analysis in physiology use other specialized tools e.g. Design.Calibrate library available in Dymola[22], Optimization toolbox from MATLAB®etc.

V. CONCLUSION

We presented a system to support identification of physiological system in the phase of parameter identification. The loosely coupled part of the system might be deployed into remote distributed computational capacity and significant speedup was shown in the case of the large complex physiological model computed in cloud computing infrastructure.

The described system is accessible via a web application and allows user to focus on input experimental data, names of parameters, visual control of calibration process and hide unnecessary complexity of configuration of the remote computation.

The continued work is oriented to enhance the complex model of human physiology - HumMod and to integrate other modeling technologies.

ACKNOWLEDGMENT

This work was supported by project MPO FR-TI3/869 and by project FR CESNET number 431. The computation was

done on virtual resources of scientific cloud provided by the czech national grid infrastructure, the association CESNET.

References

[1] A. C. Guyton, T. G. Coleman, and H. J. Granger, "Circulation: overall regulation." *Annual review of physiology*, vol. 34, pp. 13–46, Jan. 1972. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/4334846

[2] R. L. Hester, A. J. Brown, L. Husband, R. Iliescu, D. Pruett, R. Summers, and T. G. Coleman, "Hummod: a modeling environment for the simulation of integrative human physiology," *Frontiers in physiology*, vol. 2, 2011.

[3] R. L. Hester, R. Iliescu, R. Summers, and T. G. Coleman, "Systems biology and integrative physiological modelling," *The Journal of physiology*, vol. 589, no. 5, pp. 1053–1060, 2011.

[4] J. Kofránek and J. Rusz, "Restoration of Guyton's diagram for regulation of the circulation as a basis for quantitative physiological model development." *Physiological research / Academia Scientiarum Bohemoslovaca*, vol. 59, no. 6, pp. 897–908, Jan. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20533860

[5] J. Kofranek, M. Matejak, and P. Privitzer, "Hummod-large scale physiological models in modelica," in *Proceedings 8th Modelica Conference, Dresden, Germany*, 2011, pp. 713–724.

[6] J. Kofránek, M. Mateják, P. Privitzer, M. Tribula, T. Kulhánek, J. Šilar, and R. Pecinovský, "Hummod-golem edition: large scale model of integrative physiology for virtual patient simulators," in *Proceedings of World Congress in Computer Science 2013 (WORLDCOMP'13), International Conference on Modeling, Simulation and Visualisation Methods (MSV'13)*, 2013, pp. 182–188.

[7] P. Fritzson and P. Bunus, "Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation," in *Simulation Symposium, 2002. Proceedings. 35th Annual*. IEEE, 2002, pp. 365–380.

[8] "Modelica and the Modelica Association," https://modelica.org, accessed: 2013-12-01.

[9] M. C. Khoo, *Physiological control systems*. IEEE press, 2000, ch. 7: Identification of Physiological Control Systems.

[10] S. Proßand B. Bachmann, "An Advanced Environment for Hybrid Modeling and Parameter Identification of Biological Systems," *Proceedings 8th Modelica Conference, Dresden, Germany*, pp. 557–571, Jun. 2011.

[11] T. Kulhanek, J. Silar, M. Matejak, P. Privitzer, J. Kofranek, and M. Tribula, "Distributed computation and parameter estimation in identification of physiological systems," *Proceedings of VPH conference*, 2010.

[12] S. Maffioletti, R. Murri, B. Jonen, and S. Scheuring, "Computational workflows with gc3pie," in *Proceedings of the EGI Community Forum*, 2012.

[13] M. Humphrey and N. Beekwilder, "Calibration of watershed models using cloud computing," *Proceedings of IEEE 8th International Conference on E-Science*, 2012.

[14] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, 2000.

[15] L. Rossi-Bernardi and F. Roughton, "The specific influence of carbon dioxide and carbamate compounds on the buffer power and Bohr effects in human haemoglobin solutions," *The Journal of Physiology*, pp. 1–29, 1967.

[16] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J. Peetz, S. Wolf, G. I. T. I. Gmbh, and D. L. R. Oberpfaffenhofen, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models The Functional Mock-Up Interface," *Modelica*, pp. 105–114, 2011.

[17] "Servicestack," https://github.com/ServiceStack/ServiceStack, accessed: 2014-01-06.

[18] P. Kohl and D. Noble, "Systems biology and the virtual physiological human," *Molecular Systems Biology*, vol. 5, no. 1, 2009.

[19] P. Hunter, P. Robbins, and D. Noble, "The IUPS human physiome project," *European journal of physiology*, vol. 445, no. 1, pp. 1–9, 2002.

[20] T. Yu, C. M. Lloyd, D. P. Nickerson, M. T. Cooling, A. K. Miller, A. Garny, J. R. Terkildsen, J. Lawson, R. D. Britten, P. J. Hunter, and P. M. F. Nielsen, "The Physiome Model Repository 2." *Bioinformatics (Oxford, England)*, vol. 27, no. 5, pp. 743–4, Mar. 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21216774

[21] L. P. Smith, E. Butterworth, J. B. Bassingthwaighte, and H. M. Sauro, "SBML and CellML translation in Antimony and JSim." *Bioinformatics (Oxford, England)*, Nov. 2013.

[22] H. Elmqvist, H. Olsson, and S. Mattsson, "Optimization for design and parameter estimation," *Proceedings of the 4th International Modelica Conference, Hamburg*, 2005. [Online]. Available: http://elib.dlr.de/12296/

Appendix E

Modeling of Short-term Mechanism of Arterial Pressure Control in the Cardiovascular System: Object-oriented and Acausal Approach

The paper [5] published as

T. Kulhánek, J. Kofránek, and M. Mateják. Modeling of Short-term Mechanism of Arterial Pressure Control in the Cardiovascular System: Object-oriented and Acausal Approach. *Computers in Biology and Medicine*, 54:137–144, September 2014. doi:10.1016/j.compbiomed.2014.08.025

Available online at http://dx.doi.org/10.1016/j.compbiomed.2014.08.025

The author of this thesis contributed to the idea of building complex mathematical models from the basic components and keeping them in an understandable and maintainable form. Additionally, the author implemented several basic blocks and models of a pulsatile cardiovascular system in Modelica language. The other co-authors implemented the library in order to model physiology, using an integrative approach. They also implemented the complex models, which integrated different domains together.

Computers in Biology and Medicine 54 (2014) 137–144

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

Letter to the Editor

Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach

ARTICLE INFO

Keywords:
Acausal modeling
MODELICA programming language
OPENMODELICA modeling environment
DYMOLA modeling environment
Cardiovascular system

ABSTRACT

This letter introduces an alternative approach to modeling the cardiovascular system with a short-term control mechanism published in Computers in Biology and Medicine, Vol. 47 (2014), pp. 104–112. We recommend using abstract components on a distinct physical level, separating the model into hydraulic components, subsystems of the cardiovascular system and individual subsystems of the control mechanism and scenario. We recommend utilizing an acausal modeling feature of Modelica language, which allows model variables to be expressed declaratively. Furthermore, the Modelica tool identifies which are the dependent and independent variables upon compilation. An example of our approach is introduced on several elementary components representing the hydraulic resistance to fluid flow and the elastic response of the vessel, among others.

The introduced model implementation can be more reusable and understandable for the general scientific community.

1. Introduction

Fernandez de Canete et al. [1] described a closed loop cardiovascular model and short-term and long-term mechanisms of arterial pressure control in the Modelica language and DYMOLA tool. The model is decomposed into several distinct components, which express computation of output volume, pressure and flow, based on elastance input, input flow and pressure. The basic blocks from standard Modelica libraries are composed together to model the whole hemodynamics with the control mechanism [1]. The process of computing of similar flowratevolume, pressure, elastance and resistance is repeated in multiple blocks of pulmonary circulation, systemic circulation and heart circulation [1].

Redundant definition of similar relations and the mixing of more phenomenon in a single component were identified as antipatterns (bad practices) by Tiller [2], who recommends expressing single phenomenon as a general component model. This should be conducted by composing a subsystem model with instances of component models and utilizing the object-oriented features of the Modelica language.

Additionally the model [1] defines the flow of the computation of output values from input values. It is identified as causal or signal-oriented modeling. Modelica allows the expression of models not only in the causal (signal) manner, but also in an acausal manner. The acausal modeling technique is based on the fact that model variables are expressed declaratively, and the Modelica tool identifies which are the dependent and independent variables upon compilation. We have shown that an acausal description captures the essence of the modeled reality much better, and the simulation models are much more legible and, thus, also less prone to mistakes [3]. With the greatest respect to the authors of the above-mentioned publication, we would like to introduce an alternative implementation of the original model with recommendations to reduce redundancy and utilize acausal and object-oriented modeling techniques. An example of our approach is shown with respect to an elastic vessel component with unstressed volume and external pressure.

As a result of the acausal approach we believe that the alternative implementation method introduced can be more reusable by the scientific community for educational as well as for research purposes.

Additionally, Modelica tools allow rich sets of numerical solving methods for further simulation and analysis, as well as export to third party tools used by computational physiologists.

Supplementary materials contain a full source code of the alternative model implementation derived from the original publication in the Modelica language together with dependent libraries.

2. Methods

Model behavior (equations) can be expressed in Modelica in text form and in graphical form – diagram. Later in this paper we present text form as a source code listing and graphical form as detailed by the figures.

To introduce an alternative implementation of the original model, the following recommendations should be followed:

(a) Introduce an acausal connector for a hydraulic domain with "flow" variable q–flowrate and "non-flow" variable p–pressure. Introduce a single component for an elastic vessel, a hydraulic resistor and a cardiac valve with one or two acausal connectors and describe equations of volumetric flow, volume and pressure based on the parameters of elastance and resistance.

http://dx.doi.org/10.1016/j.compbiomed.2014.08.025
0010-4825/© 2014 Elsevier Ltd. All rights reserved.

(b) Separate a general model from a control model and a specific experiment. Utilize object-oriented features of the Modelica language to reuse the architecture model and replace the concrete implementation of the subsystem model with a derived model experiment.

(c) Prefer the text form of Modelica notation to define equations on a component level model. Prefer the diagram form of Modelica notation to express relations between components on a higher subsystem and system level model.

We have recently published a Modelica library, referred to as Physiolibrary [4], to support modeling in the physiological domain. The library contains several hydraulic components that can be directly used to model the cardiovascular system following the recommendation (a). Table 1 contains icon, description and equations characterizing the components. The equations of these components are defined in Modelica using text form following the recommendation (c) and can be seen in supplementary materials. The models of the pulmonary circulation and the systemic circulation are defined by diagrams in Figs. 1 and 2 utilizing the components from Physiolibrary (from Table 1). The models are almost equivalent to the pulmonary and systemic blocks from the original work [1], apart from the pulmonary and aortic valves, that we moved to the model of the heart subsystem described later.

Additionally the systemic circulation contains a block to measure blood properties in aorta, it extends the existing block from Physiolibrary that measures flow. Additionally it computes systolic, diastolic and mean pressure during a single cardiac cycle. The mean arterial pressure P_{mean} during the cardiac cycle is counted as the average of pressure going into the component ($q_{in_{pressure}}$) from the beginning of the cardiac cycle (T_0) during the heart period (HP) by formula:

$$P_{mean} = \frac{\int_{T_0}^{T_0+HP} q_{in_{pressure}}\,\mathrm{d}t}{HP} \tag{1}$$

The component is declared by icon () and by the statements and equations in the following Modelica listing (shortened):

```
model AortaFlowMeasurement "measures flow, diastolic, systolic and mean pressure"
...
  discrete Boolean b(start=false) "beat signal";
  Time T0(start=0) "start of cardiac cycle ";
  discrete Time HP(start=1) "length of cardiac cycle";
initial algorithm
  Ps := q_in.pressure;
  Pd := q_in.pressure;
equation
  Pmax = max(Pmax, q_in.pressure);
  Pmin = min(Pmin, q_in.pressure);
  b = der(q_in.pressure) > 0;
  when {b and not pre(b)} then
    T0 = time "initial time of current cardiac cycle";
    HP = if (pre(T0) > 0) then time - pre(T0) else 1;
    Pmean = SumPressure / pre(HP) "mean pressure";
    Ps = Pmax "systolic pressure = maximum pressure during cardiac cycle";
    Pd = Pmin "diastolic pressure=minimal pressure during cardiac cycle";
    reinit(SumPressure, 0) "reinitialisation of sum pressure";
    reinit(Pmax, q_in.pressure) "reinitialisation of maximal pressure";
    reinit(Pmin, q_in.pressure) "reinitialisation minimal pressure";
  end when;
  der(SumPressure) = q_in.pressure;
end AortaFlowMeasurement;
```

We separate out the variable elastance(compliance) generator from the heart subsystem in Fig. 3.

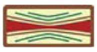
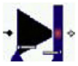
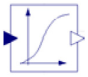
The block "pulsos" (identified by icon ()) generates the relative heart phase within the heart period during a simulation time based on the heart rate signal [1]. However, in contrast to the original implementation we define it in text form per recommendation (c), and with the changed behavior: the heart period HP is changed per the input signal heart rate only at the moment when the new cardiac cycle begins. The output signal "heartphase" modeled in the original work can be presented as the following equation:

$$heartphase = \frac{time - T_0}{HP} \tag{2}$$

Thus, the model behavior is defined by the following listing:

```
model pulsos  "relative position in heart period"
  discrete Physiolibrary.Types.Time HP(start = 0)
    "heart period - duration of cardiac cycle";
  Boolean b(start = false);
  discrete Physiolibrary.Types.RealIO.TimeOutput T0
    "start time of cardiac cycle";
  Physiolibrary.Types.RealIO.FrequencyInput HR;
  Modelica.Blocks.Interfaces.RealOutput heartphase;
equation
  b = time - pre(T0) >= pre(HP); //new cycle begins?
  when {initial(),b} then
    T0 = time;  //update start time of cardiac cycle
    HP = 1 / HR; //update heart period
  end when;
  heartphase = (time - pre(T0))/pre(HP);
end pulsos;
```

Table 1
Icon and description of hydraulic components used from Physiolibrary [4].

Icon	Description
◆◇	Hydraulic connectors – the MODELICA tool generates the following equations to keep “Kirchhoff law” analogy for all connected component's non-flow variables $p_1 \dots p_n$ - pressure and “flow” variable $q1 \dots q_n$ - flowrate: $p_1 = p_2 = \cdots = p_n$ (3) $\sum_{i=1}^{n} q_i = 0$ (4)
	Hydraulic Resistor – characterized by G–conductance parameter (reciprocal value of resistance $G = 1/R$) and defined by relation among quantities from both hydraulic connectors, q–flowrate and $(p_{out} - p_{in})$ –pressure gradient: $q = G * (p_{out} - p_{in})$ (5)
	Elastic compartment – characterized by C–compliance parameter(reciprocal value of elastance $C = 1/E$), V_0–unstressed volume parameter, p_0–external pressure parameter and by equation among p–pressure, V–volume and q–flowrate: $p - p_0 = \begin{cases} 0 & \text{if } V < V_0 \\ \frac{V - V_0}{C} & \text{otherwise} \end{cases}$ (6) $\frac{\mathrm{d}V}{\mathrm{d}t} = q$ (7)
	Valve is characterized by the direction where the flow is allowed, by inflow and backflow conductance
	2D natural cubic interpolation spline defined by x, y and slope points. Used to define curve determined by empirical data

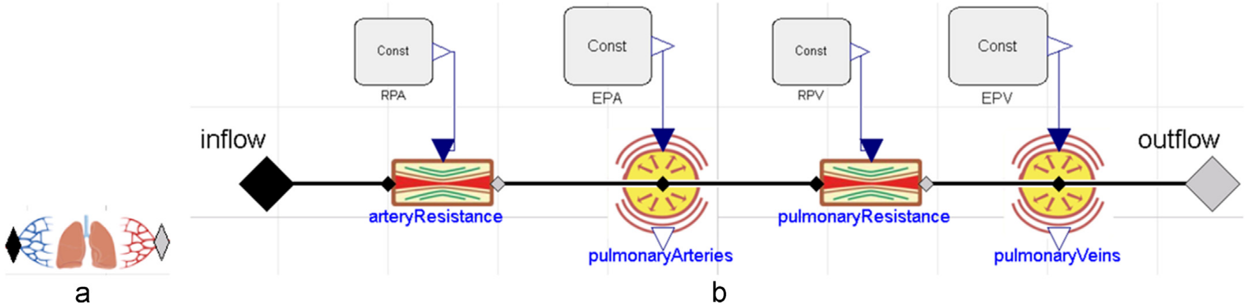


Fig. 1. Pulmonary circulation model icon (a) and its constitutive diagram (b). The elastic compartments and the resistors are connected via hydraulic connectors. The model parameters are presented as block Const with an appropriate type and value.

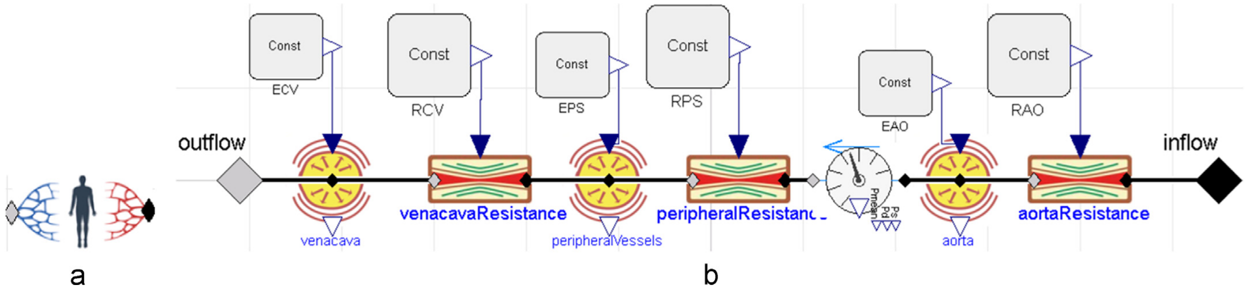


Fig. 2. Systemic circulation icon (a) and its constitutive diagram (b). The elastic compartments and the resistors are connected via hydraulic connectors. The model parameters are presented as block Const with an appropriate type and value.

The heart model in Fig. 4 consists of the left and right parts; they are driven separately by the appropriate elastance generator, but with the same heart rate signal.

Finally, the generic model of the hemodynamics without any control mechanism is shown in Fig. 5. Note that the diagram looks very similar to the usual conceptual decomposition of the cardiovascular system.

Following recommendation (b), the model defines heart, systemicCirculation and pulmonaryCirculation components as “replaceable” to allow replacement with some derived subtype

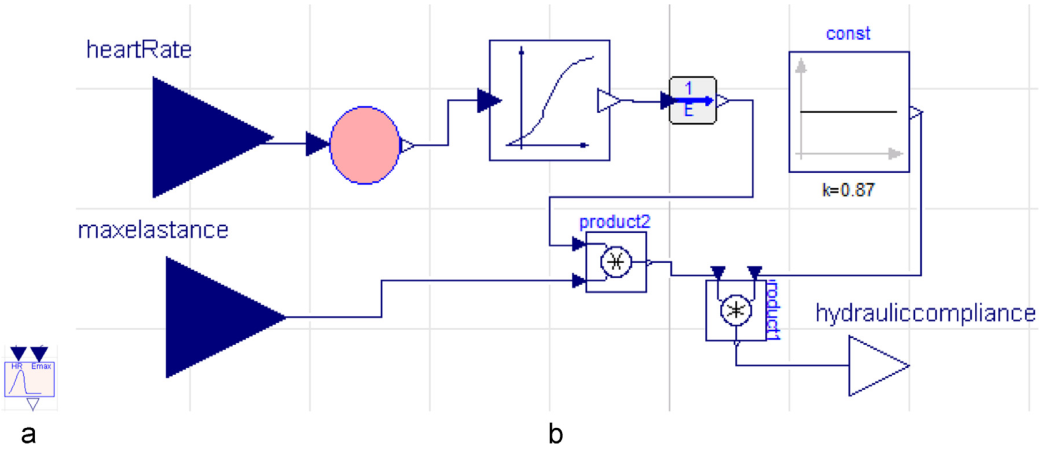


Fig. 3. Compliance generator icon (a) and its constitutive diagram (b). The relative elastance is generated from the relative time position in the heart period via empirical curves generated from points. As Physiolibrary uses a connector for compliance – reciprocal value of elastance – we convert elastance to compliance with a block $1/E$.

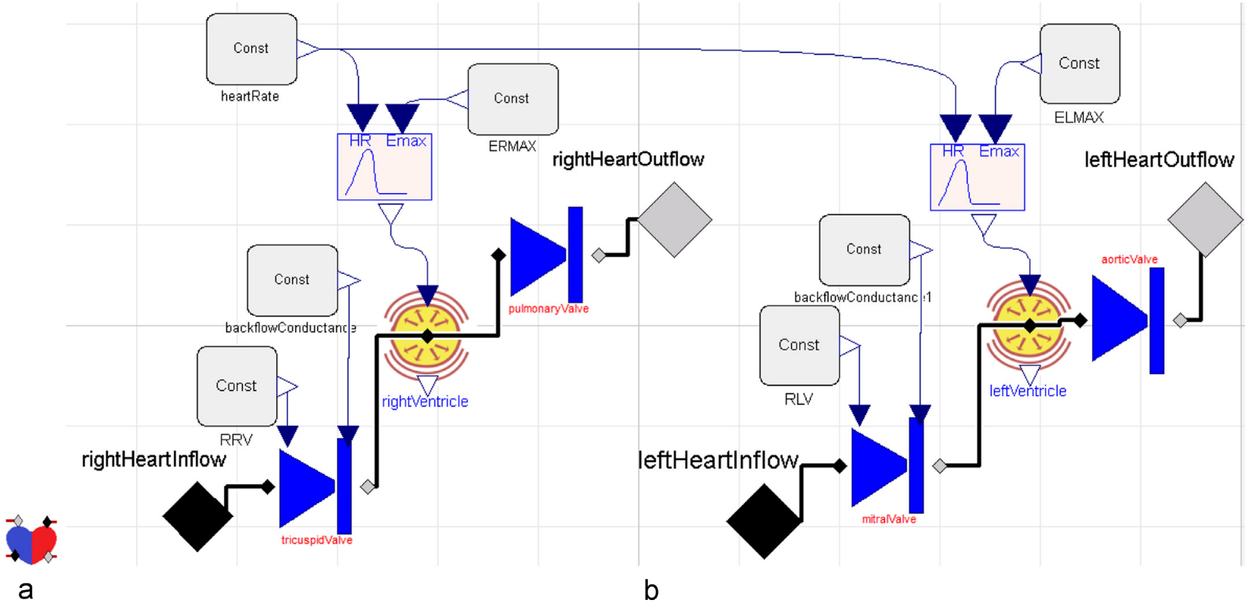


Fig. 4. Heart model icon (a) and its constitutive diagram (b). The valves are connected with the elastic compartment modeling the ventricle driven by the variable elastance component. Heart rate is shared but other parameters are different for each part.

of the model as seen from the text form of the model in the following listing:

```
model Hemodynamics_pure
  replaceable Parts.Heart heart;
  replaceable Parts.SystemicCirculation systemicCirculation;
  replaceable Parts.PulmonaryCirculation pulmonaryCirculation;
...
```

To simulate a particular scenario or control mechanism, selected parameters of the model need to be manipulated externally during simulation. We introduced derived model *SystemicCirculation_baro* and redeclared the constant block with a compatible control block as seen in the following listing which allows manipulation with the model parameter by an external component.

```
model SystemicCirculation_baro
  extends FernandezModel.Parts.SystemicCirculation(
    redeclare HydraulicConductanceControl RPS,
    redeclare HydraulicComplianceControl ECV);
...
```

The subsystem *SystemicCirculation_baro* contains additional input connectors connected with the previously redeclared control blocks as shown in Fig. 6.

A similar technique is used for the heart component in Fig. 7. The model of controllable hemodynamics is an extension of the generic model of hemodynamics. The model uses new implementation of the controllable heart and systemic circulation and introduces input and output connectors specific for the baroreceptor control system, as seen in the following listing and in Fig. 8.

```
model Hemodynamics_controllable
  extends Models.Hemodynamics_pure(
    redeclare Heart_baro heart,
    redeclare SystemicCirculation_baro systemicCirculation);
...
```

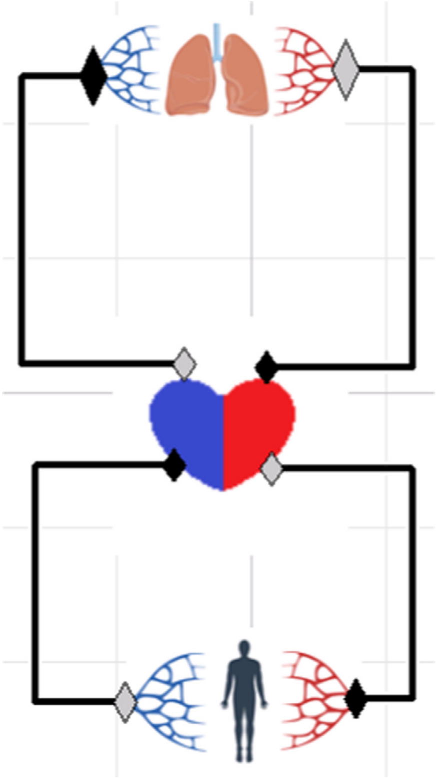


Fig. 5. Complete model of the cardiovascular system without any regulation.

The controllable model of hemodynamics is connected with baroreceptor module and with the module driving the simulation of stenosis, as shown in Fig. 9

3. Results

The implementation was changed in the case of block *pulsos* and block *AortaFlowMeasurement*. Other parts are equivalent to the original work and use the same initial values of state variables and parameters [1]. The model can be executed in a free OPENMODELICA tool developed by Open Source Modelica Consortium [5]. The dynamics of the aortic pressure during a sudden change of elastance of vena cava shown in Fig. 10 gives similar results as the original work. The evolution of heart rate and ventricle elastance is smoother because the mean arterial pressure and the heart period signals are changed only at the beginning of the next cardiac cycle.

4. Discussion and summary

We believe that the recommendation (a) applied to the model allows it to better capture the essence of the modeled reality, in contrast to the original "signal"-like approach where it might be hard to deduce the concept of the model for a user who is not familiar with this modeling technique. Additionally further modification or enhancements of the basic component, e.g., elastic vessel by adding non-linear compliance or active tone, will be propagated to the existing model using this component with no or minimal need for further modification of the model. Such modification will appear in the model within the modified components as modified behavior or new parameters which can be set.

For example, the original model expresses the pressure p, volume V and compliance C (reciprocal value of elastance) as equation $p = V/C$. In contrast to the original model, the alternative implementation modifies the basic element of elastic compartment with unstressed volume V_0 and external pressure p_0 as Eq. (6). V_0 and p_0 are set to 0 by default and can be changed in further model experiments.

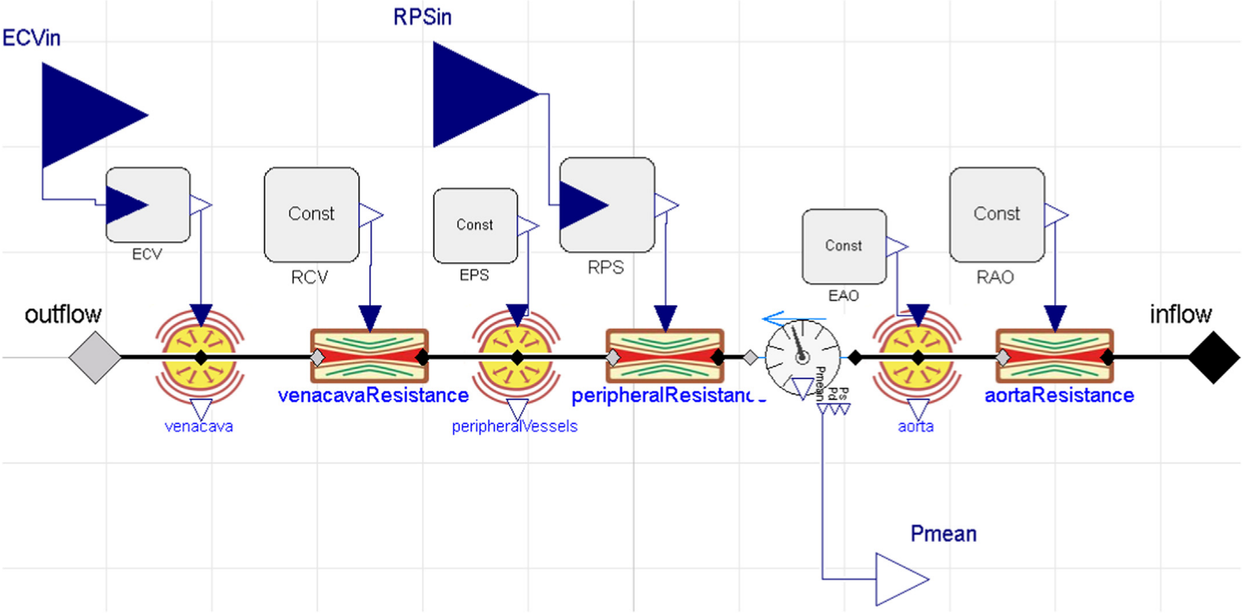


Fig. 6. SystemicCirculation prepared to be driven by elastance of vena cava and peripheral resistance coming from outside.

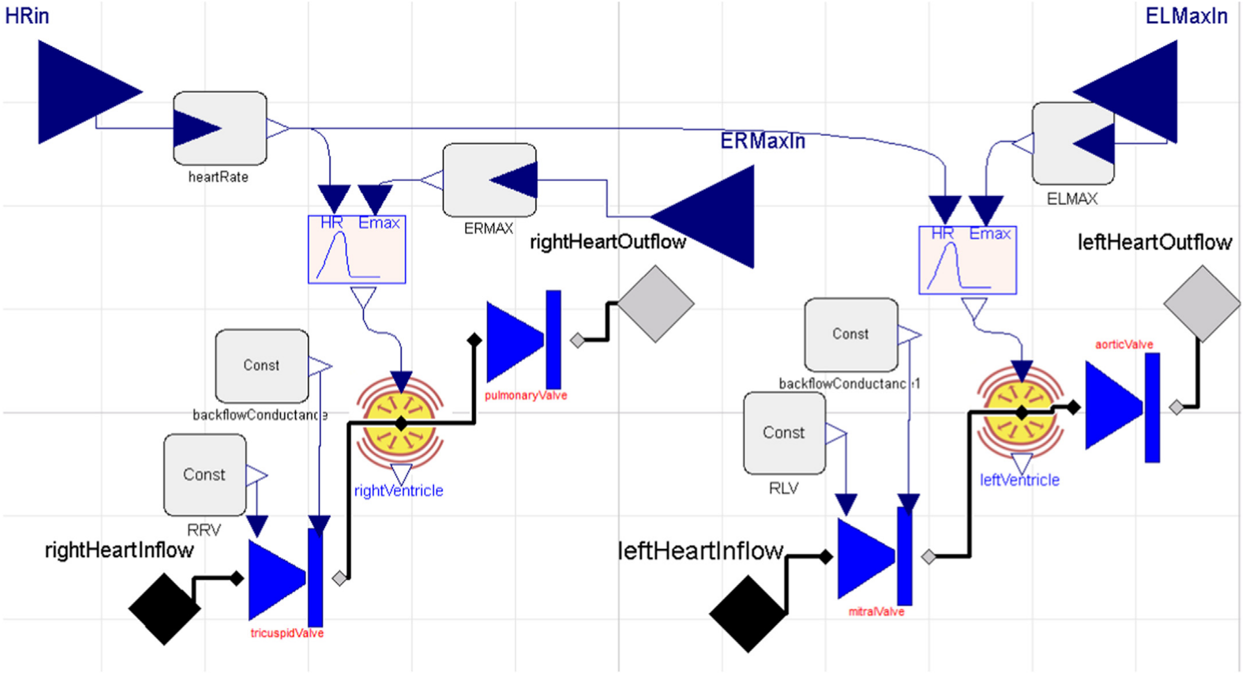


Fig. 7. The Heart model prepared to be driven by an elastance, and a change of the heart rate from outside control mechanism.

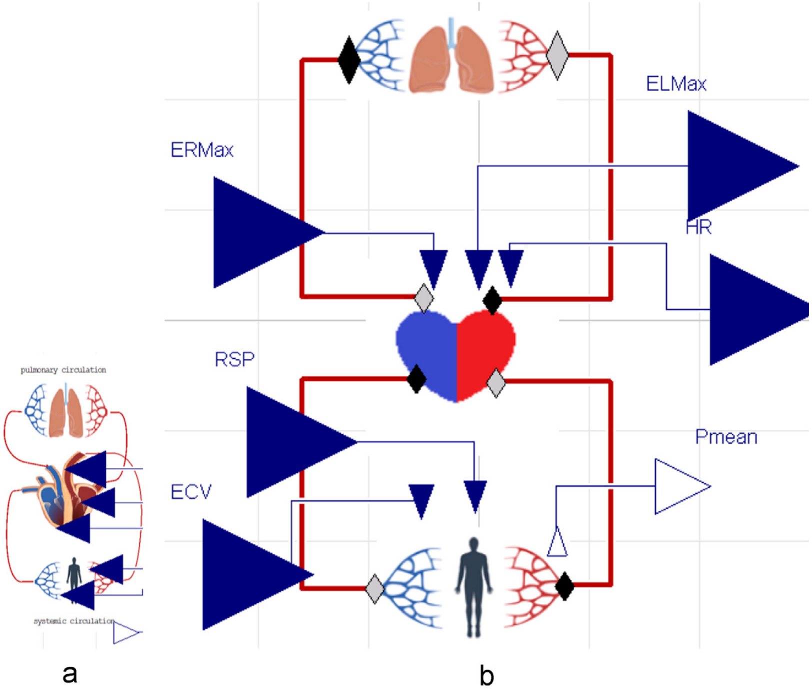


Fig. 8. Icon of the controllable model of hemodynamics (a) extended by input and output connectors and its constitutive diagram (b) connecting new connectors with the redeclared implementation of controllable heart and systemic circulation.

Such enhancement is propagated by inheritance to all elastic compartments used within the model. When the original model [1] needs to be enhanced with such features of unstressed volume or external pressure effect, then more than one component (compartment with valve, compartment without valve and systemic circulation) needs to be modified.

The recommendation (b) applied to the generic model brings the advantage of a major feature of object-oriented programming –

polymorphism. A model (e.g. Hemodynamics_pure) comprises a subsystem model of a certain type (e.g. Heart) will work correctly with another subsystem that is compatible with the certain type (e.g. Heart_baro). The subsystem models might be reimplemented and replaced in an existing model without touching the generic model. Therefore, it may be appropriate for specific in silico experiments.

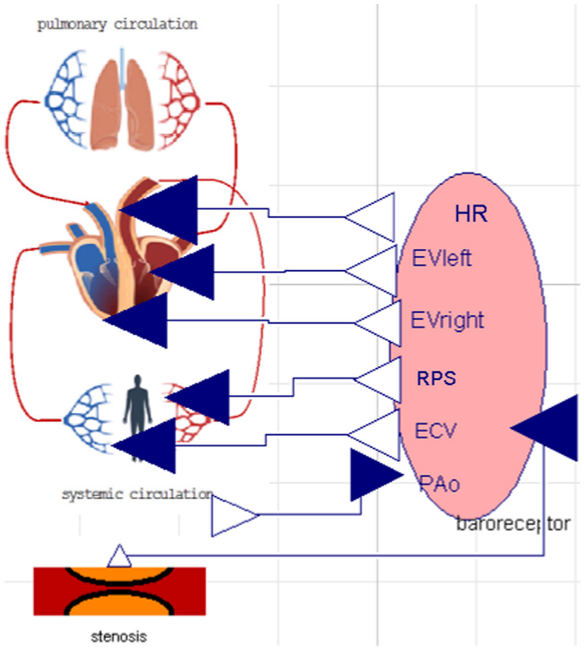


Fig. 9. The model of hemodynamics with baroreceptor control and a module manipulating the elastance during simulation – stenosis of vena cava simulation.

The Modelica standard library allows equations to be expressed using blocks in diagrams; however, it should not be overused. For example, the recommendation (c) applied to the component “pulsos” shows the Eq. (2) cleanly in the text form of Modelica notation (see the listing above). During the reimplementation of this block, we reformulated the method of changing the heart rate signal where heart period and pulse generation are updated at the beginning of the next cardiac cycle. This facilitates the computation. In the original work, the heart period and pulse generation are changed immediately. This difference is seen in simulating the baroreflex control, where the control mechanism is smoother, as seen in Fig. 10.

On the higher subsystem level, the recommendation (c) is optional. If there are more relations among components then a diagram form may be more understandable than a textual form. For example, the block VariableElasticityGenerator in Fig. 3 can be expressed in an equivalent text form as seen in the following model listing:

```
model VariableElasticityGenerator_text
...
equation
  pulsos.HR = heartRate;
  curve.u = pulsos.heartPhase;
  hydrauliccompliance = 0.87 * (maxelastance/curve.val);
end VariableElasticityGenerator_text
```

The original work contains a long-term pressure control mechanism, which was not implemented within this alternative implementation. However, it can be done following the above recommendations similar to the presented short-term “baroreflex” pressure control.

We believe that the introduced alternative approach to modeling the cardiovascular system will enhance the understandability

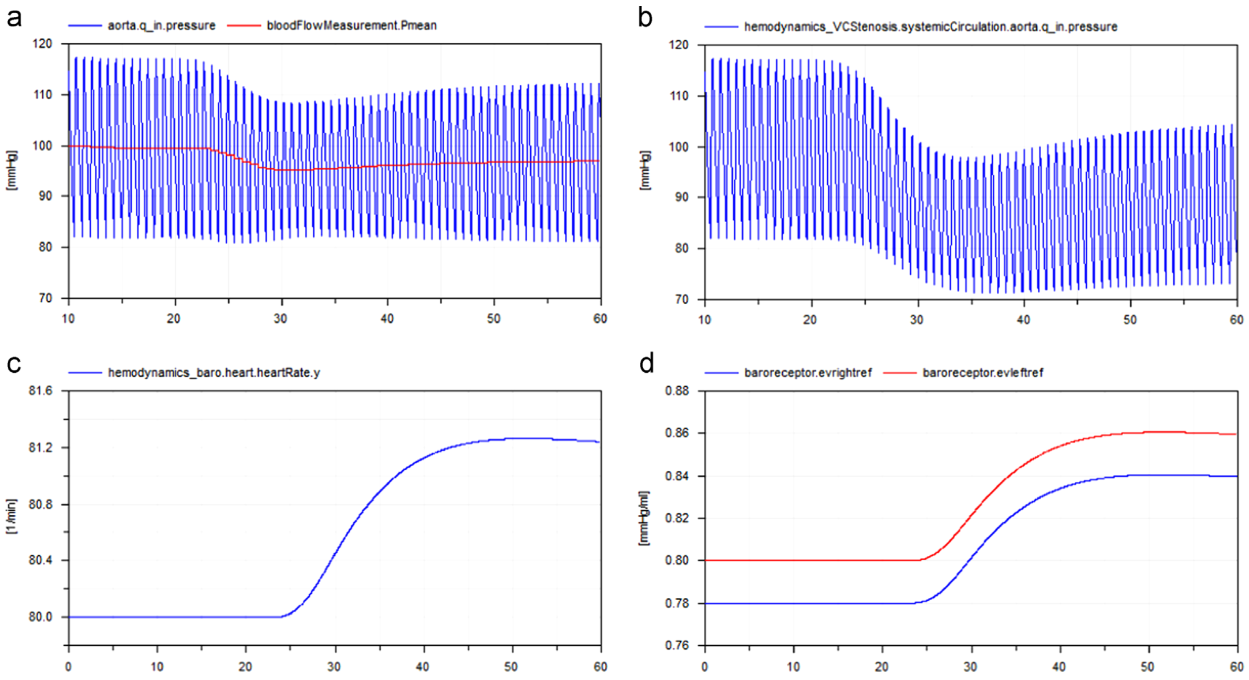


Fig. 10. Simulation of the aortic pressure response on vena cava elastance reduction at the simulation time 20 s for model with (a) and without (b) baroreceptor control showing the evolution of heart rate (c) and ventricle elastance (d).

and reusability of the excellent work done by the authors of the original model.

Conflict of interest statement

None declared.

Acknowledgments

This paper describes the outcome of research that has been accomplished as part of a research program funded by the Ministry of Industry and Trade of the Czech Republic via the grant FR-TI3/869; and by The Ministry of Education, Youth and Sports via the grant SVV-2014-260033.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.compbiomed.2014.08.025.

References

[1] J. Fernandez de Canete, J. Luque, J. Barbancho, V. Munoz, Modelling of long-term and short-term mechanisms of arterial pressure control in the cardiovascular system: an object-oriented approach, Comput. Biol. Med. 47 (2014) 104–112. http://dx.doi.org/10.1016/j.compbiomed.2014.01.006.

[2] M. Tiller, Patterns and anti-patterns in Modelica, in: Proceedings Modelica 2008 Conference, 2008, pp. 647–655.

[3] J. Kofránek, M. Mateják, P. Privitzer, M. Tribula, Causal or acausal modeling: labour for humans or labour for machines, Technical Computing Prague, 2008, pp. 1–16.

[4] M. Mateják, T. Kulhánek, J. Šilar, P. Privitzer, F. Ježek, J. Kofránek, Physiolibrary—modelica library for physiology, in: Proceedings of the Tenth Modelica Conference, March 10–12, 2014, Lund, Sweden, 2014, pp. 499–505. URL ⟨http://dx.doi.org/10.3384/ecp14096499⟩.

[5] P. Fritzson, P. Aronsson, H. Lundvall, K. Nyström, A. Pop, L. Saldamli, D. Broman, The OpenModelica modeling, simulation, and software development environment, Simul. News Eur. 44 (45) (2005).

Tomáš Kulhánek, Jiří Kofránek*, Marek Mateják
Institute of Pathological Physiology, Charles University in Prague, U Nemocnice 5, 12853 Praha 2, Czech Republic
E-mail address: kofranek@gmail.com (J. Kofránek)

Received 15 May 2014; accepted 22 August 2014

* Corresponding author.

Appendix F

Simple Models of the Cardiovascular System for Educational and Research Purposes

The paper [6] published as

T. Kulhánek, M. Tribula, J. Kofránek, and M. Mateják. Simple Models of the Cardiovascular System for Educational and Research Purposes. *MEFANET Journal*, 2(2):56–63, 2014

Available online at `http://mj.mefanet.cz/mj-04140914`

Author of this thesis proposed and implemented several models related to cardiovascular system in Modelica language. Other co-authors updated general library for modeling physiology and proposed utilization of the models in educational simulators.

SIMPLE MODELS OF THE CARDIOVASCULAR SYSTEM FOR EDUCATIONAL AND RESEARCH PURPOSES

Tomáš Kulhánek*, Martin Tribula, Jiří Kofránek, Marek Mateják
Institute of Pathological Physiology, 1st Faculty of Medicine, Charles University in Prague, Prague, Czech Republic
** Corresponding author: tomaton@centrum.cz*

ARTICLE HISTORY
Received 14 September 2014
Revised 30 October 2014
Accepted 4 November 2014
Available online 24 November 2014

KEYWORDS
Modelica
modeling physiology
cardiovascular system
Physiolibrary

ABSTRACT — *Modeling the cardiovascular system as an analogy of an electrical circuit composed of resistors, capacitors and inductors is introduced in many research papers. This contribution uses an object oriented and acausal approach, which was recently introduced by several other authors, for educational and research purpose. Examples of several hydraulic systems and whole system modeling hemodynamics of a pulsatile cardiovascular system are presented in Modelica language using Physiolibrary.*

INTRODUCTION

The mathematical formalization of the cardiovascular system (CVS), i.e. the models, can be divided into two main approaches. The first approach builds 3D models with geometrical, mechanical properties and the time-dependence of the compartments of the CVS. The complexity of such models implicates high demand on computational power to simulate them.

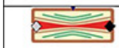
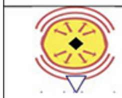
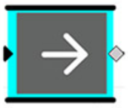
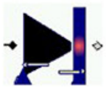
The second approach, lumped parameter approach, uses a high degree of simplification and different complex regions are generalized as single compartments characterized by lumped parameter. Using this approach, the cardiovascular system is modeled as an analogy of an electrical circuit with a set of resistors, capacitors (elastic vessels), diodes (valves) and inductance (inertial elements) in a closed loop. The relationship between pressure and volume quantities is studied throughout the modeled system. This type of model is used for the improvement of reliability in patient-specific diagnosis, for example [1,2]; or for educational purposes and training [3–9]. Technology used for modeling is either proprietary, e.g. MATLAB–Simulink or MATLAB-Simscape in the case of the CVS model introduced by Fernandez et al.[7], home grown open technology introduced by Hester et al.[3], standard grown technology from the scientific community, such as SBML [10], JSIM [11] or CellML [1,12] or industrial standard technology implemented by several vendors, such as the Modelica language [8,9].

This article introduces a modeling method which follows the second approach for modeling CVS as pressure-volume relations and introduces example models implemented in the Modelica language. We believe that one of the properties of the Modelica language — acausal modeling — seems to capture the modeled reality much better and allows connecting simpler models into quite complex systems of differential equations in an understandable form [4,13,14]. Additionally, we use a library for modeling physiology in Modelica — Physiolibrary — as the model diagrams based on the Physiolibrary components are self-descriptive in most cases [15,16].

METHODS

Modelica is an object-oriented language, thus, in further text when we refer to components, models and systems they are classes, and the one's with concrete values of parameters are instances as is usual in standard object-oriented programming. Models (classes) can inherit behavior or can be composed from other

TABLE 1 Selected components from Physiolibrary with description

Icon	Description
◆◇	**Hydraulic connectors** – declared as acausal, without prejudicing any kind of computational order. The MODELICA tool generates the following equations to keep the analogy of "Kirchhoff laws" for all connected components "non-flow" variables *p1..pn* determining the pressure and "flow" variable *q1..qn* determining flowrate: $p_1 = p_2 = \ldots = p_n$ (1) $\sum_{i=1}^{n} q_i = 0$ (2)
	**Hydraulic conductor** or (hydraulic resistor) is characterized by the parameters *G*-conductance (reciprocal value of resistance $G=\frac{1}{R}$) and models the relation between pressures p_{in}, p_{out} and flowrates q_{in}, q_{out} from the two connectors by the following formulas: $q_{out} = -q_{in}$ (3) $q_{in} = G*(p_{out} - p_{in})$ (4)
	**Elastic vessel** is characterized by these parameters: *C*-compliance (reciprocal value of elastance $C=\frac{1}{E}$), *V0* – unstressed volume, *p0* – external pressure and by equation among the variable *V*-volume and variables of the hydraulic connector *q*-flowrate and *p*-pressure: $p - p_0 = \begin{cases} 0 & \text{if } V < V_0 \\ (V - V_0)/C & \text{otherwise} \end{cases}$ (5) $\frac{\mathrm{d}V}{\mathrm{d}t} = q$ (6)
	UnlimitedPump is defined as the generator of the flowrate q_{out} based on the value "solutionFlow". $q_{out} = -solutionFlow$ (7)
	**Inertia element** is characterized by the *I* – inertance parameter and the relationship between pressure and solution flow from the two connectors by the following formula: $q_{out} = -q_{in}$ (8) $\frac{\mathrm{d}q_{in}}{\mathrm{d}t} = \frac{p_{in} - p_{out}}{I}$ (9)
	**Hydraulic valve** is characterized by g_{on} outflow and g_{off} backflow conductances, described by parametric equations: $dp = \begin{cases} pass/g_{on} + P_{knee} & \text{for } pass > 0 \\ pass + P_{knee} & \text{otherwise} \end{cases}$ (10) $q = \begin{cases} pass + P_{knee} * g_{off} & \text{for } pass > 0 \\ pass * g_{off} + P_{knee} * g_{off} & \text{otherwise} \end{cases}$ (11)

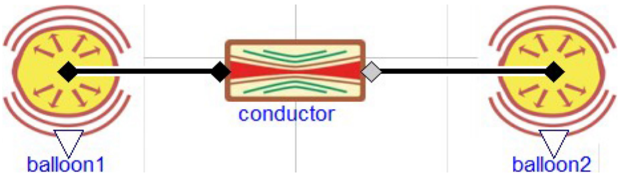


FIGURE 1 Two balloon example model diagram. It connects two balloons characterized with compliance (elastance) via a conductor (resistor).

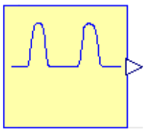


FIGURE 2 Icon of component "pulses"

models (classes). For the composition a special purpose class — connector — can be used to define variables of the model shared with other models.

The following models in the Modelica language are usually declared by an icon and defined either in diagram view or textual notation. The icon declares an interface of how the particular model instance can appear in other higher level model diagrams. The diagram and textual view declares the internal structure, which are in fact algebraic and differential equations among model parameters and variables.

As was mentioned earlier, Modelica models can be acausal, the equations can be expressed declaratively and the Modelica tool will decide which variables are dependent and independent based on the context upon compilation. This is extremely useful when several existing components are composed into a larger system model and the complexity is kept in a maintainable set of diagrams.

In further examples we use the following components from Physiolibrary (Table 1).

Two connected balloons

The model diagram (Figure 1) expresses the situation of two elastic balloons filled with liquid connected via an inelastic tube with characteristic resistance. Such a model diagram connects the system of equations of the two elastic vessels and one resistor and Modelica will figure out which variables will be independent and which will be dependent. In this example the pressure will be computed first from the initial volume of the two balloons by equation (5). This determines the pressure gradient, which directs the flow from the first balloon to the second one in equation (4). The flowrate affects the volume change in each elastic vessel in equation (6) and again causes the change of pressure from equation (5).

Windkessel models

The Windkessel models simplify the view on CVS as a series of compartments with capacity (compliance or elasticity of an elastic compartment), resistance (conductance or resistance of a conductor/resistor), impedance and inertance properties. They study the Windkessel effect, which maintains a relatively stable flowrate from the system, although the pulsatile flowrate comes from the pump. Several derivatives and improvements of Windkessel models were introduced [17].

Figures 3, 4 and 5 show diagrams of the Windkessel model of 2 elements, 3 elements and 4 elements in Modelica using Physiolibrary components. Additionally, the model diagrams contain component 'pulses' declared by the icon in Figure 2 and defined by following Modelica listing:

```
model pulses
  import Physiolibrary.Types.*;
  Physiolibrary.Types.RealIO.VolumeFlowRateOutput
volumeflowrate;
  discrete Time T0 "beginning of cardiac cycle";
  Boolean b(start=false);
  discrete Time HP "duration of cardiac cycle";
  parameter Frequency HR = 1.2;
  Time tc "relative time in cardiac cycle";
  parameter Time TD1=0.07 "relative time of start of
systole";
  discrete Time TD2 "relative time of end of systole";
  parameter VolumeFlowRate QP = 0.000424 "peak volume
flowrate";
equation
  b = time - pre(T0) >=pre(HP) "true if new cardiac cycle
begins";
  when {initial(),b} then
    T0 = time "set beginning of cardiac cycle";
    HP=1/HR "update length of cardiac cycle";
    TD2 = TD1+(2/5)*HP "compute end time of systole";
  end when;
  tc = time-T0 "relative time in cardiac cycle";
  volumeflowrate =
    if tc<TD1 then 0 else
    if tc<TD2 then sin((tc-TD1)/(TD2-TD1)*Modelica.
Constants.pi)*QP else
    0 "zero before and after systole, otherwise sin up to
peak flow";
end pulses;
```

This component generates regular pulses of flowrate during the systole period, which are approximated by the sinus function increasing from zero up to the peak flowrate and back to zero during 2/5 of a cardiac cycle, while at other times it generates a zero signal. The keyword 'discrete' notes the Modelica tool, wherein such variables will be changed only in discrete events, in our case at the beginning of the cardiac cycle.

A simple model of the cardiovascular system

We divide the CVS system into systemic circulation and pulmonary circulation; each circulation is characterized by a modified 2-element Windkessel model with an additional compliance component expressing the systemic and pulmonary veins (Figure 6).

The 'rightHeart' and 'leftHeart' components are instances of the model 'HeartPump'. The flowrate of the model "HeartPump" is determined by the filling pressure and by the slope of the Starling curve. The equations are defined in the following Modelica text notation:

```
model HeartPump
  Physiolibrary.Hydraulic.Interfaces.HydraulicPort_a
inflow "inflow";
  Physiolibrary.Hydraulic.Interfaces.HydraulicPort_b
outflow "outflow";
  parameter Physiolibrary.Types.HydraulicConductance
StarlingSlope;
equation
  inflow.q + outflow.q =0;
  inflow.q = StarlingSlope * inflow.pressure;
end HeartPump;
```

The non-pulsatile model shows the mean values of pressure and flow throughout CVS. To enhance the above mentioned model, we first define the pulsatile model PulsatileHeartPump by diagram (Figure 7).

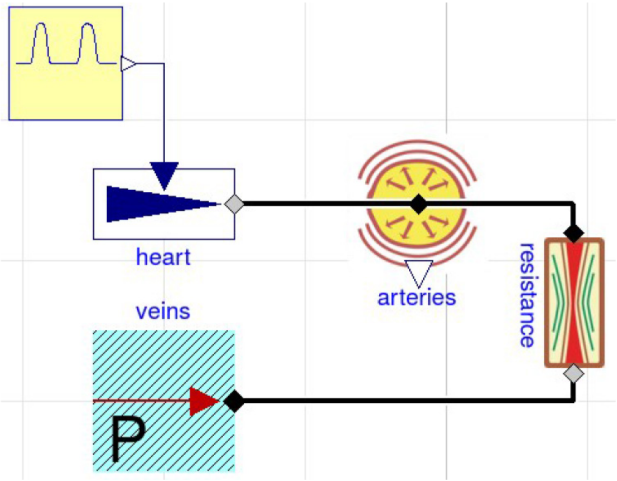


FIGURE 3 2-element Windkessel model characterized by compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s)

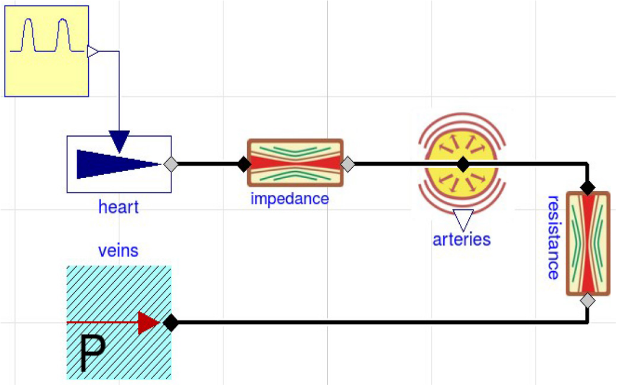


FIGURE 4 3-element Windkessel model characterized by impedance (here approximated by conductance = 200ml/(mmHg.s)), compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s)

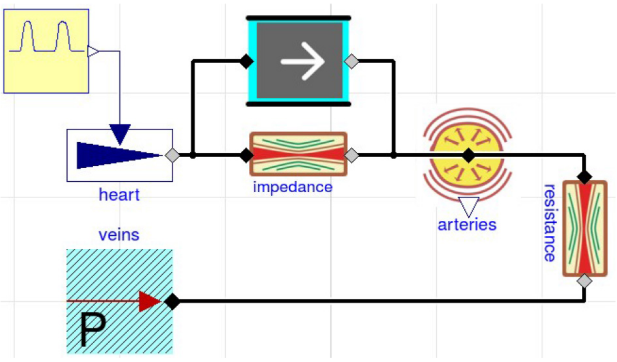


FIGURE 5 4-element Windkessel model characterized by impedance (here approximate by conductance = 200ml/(mmHg.s)), inertia I = 0.005 mmHg.s^2/ml, compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s)

We utilize the object-oriented features in Modelica to extend the non-pulsatile circulation by only replacing the heartPump instances by pulsatile heart pump instances as seen in the following listing. Note the redefinition of values (in SI units) of parameter QP, k, volume_start:

```
model PulsatileCirculation
  extends NonPulsatileCirculation(
    redeclare Parts.PulsatileHeartPump
rightHeart(pulses(QP = 0.000338)),
    redeclare Parts.PulsatileHeartPump
leftHeart(pulses(QP=0.000338)),
    CAS(k=7.2755972857029e-09),
    SystemicArteries(volume_start=0.000603),
    SystemicVeins(volume_start=0.003991));
end PulsatileCirculation;
```

REFERENCE MODEL WITH VENTRICLES

For further comparison purposes, we have chosen model of CVS published by Fernandez de Canete et al. implemented originally in MATLAB-Simscape [7] . This model is presented in Modelica using Physiolibrary components in one diagram (Figure 8). Systemic and pulmonary circulation are presented as modified Windkessel model with inertia element connected via heart subsystem. Additionally, each side of heart is composed by two valves and a ventricle. The ventricle is modeled as an elastic compartment driven by variable elasticity generator defined by the following Modelica listing:

```
model TimeVaryingElastance
  parameter Physiolibrary.Types.HydraulicElastance Ed
"e. of diastole";
  parameter Physiolibrary.Types.HydraulicElastance Es
"e.of systole";
  parameter Physiolibrary.Types.Pressure Pi0 "peak
isovolumic pressure";
  Physiolibrary.Types.Time tm
    "relative time from the beginning of cardiac cycle";
  discrete Physiolibrary.Types.Time HP "heart period";
  discrete Physiolibrary.Types.Time t0 "start time of the
cardiac cycle";
  discrete Physiolibrary.Types.Time ts "duration of
systole";
  Real a;
  Physiolibrary.Types.RealIO.HydraulicComplianceOutput C;
  Physiolibrary.Types.HydraulicElastance E;
  Physiolibrary.Types.RealIO.PressureOutput Pi;
  Physiolibrary.Types.RealIO.FrequencyInput HR "heart
rate";
equation
  tm = time - pre(t0);
  if (tm<pre(ts)) then
    a= (1-cos(2*Modelica.Constants.pi*tm/pre(ts)))/2;
  else
    a=0;
  end if;
  E=Ed+Es*a;
  C=1/E;
  Pi = Pi0*a;
  when {initial(), tm >= pre(HP)} then
    HP = 1/HR;
    t0= time;
    ts = 0.16+0.3*HP;
  end when;
end TimeVaryingElastance;
```

RESULTS

The two balloons model (Figure 1) is used in simulators to demonstrate that a liquid (e.g. blood) flows from the part with higher pressure to the part with lower pressure. If there is no other active force, the system converges to an equilibrium, where the pressures will be equal and no other flow occurs between the balloons (Figure 9).

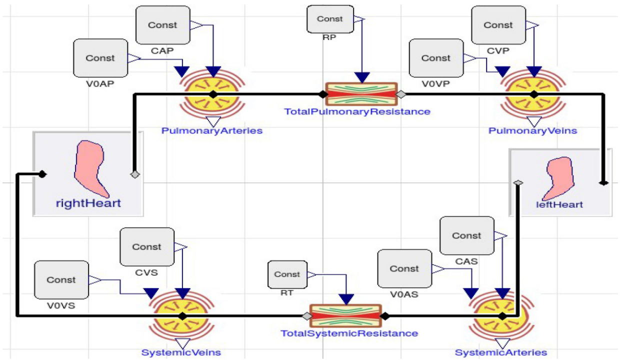


FIGURE 6 Non-pulsatile circulation model diagram. Pulmonary circulation on top of the diagram is connected via side of the heart with systemic circulation at the bottom of the diagram

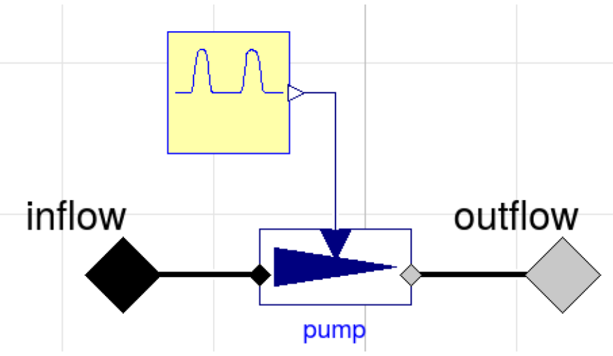


FIGURE 7 Pulsatile heart pump model

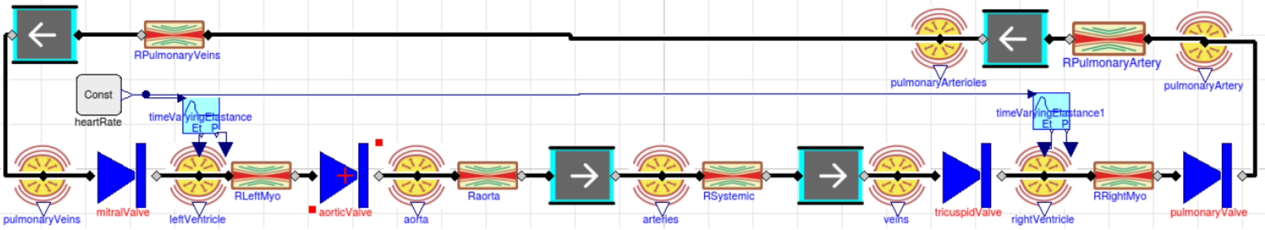


FIGURE 8 Reference model by Fernandez de Canete et al. [7] in Modelica using Physiolibrary components

This model and simulation is used to demonstrate that the blood has a tendency to flow from the arteries with lower compliance and higher pressure to the veins. Without any external force, the majority of blood accumulates in veins rather than in arteries.

Simulation of the Windkessel models imitate the heart pulses by the generator controlled with the 'pulses' component (Figure 10).

The simulation of Windkessel model shows the Windkessel effect, which reduces a high variation of flowrate coming from the heart to a relatively stable flowrate in the systemic peripheral vessels due to the compliance of the elastic compartment.

The simulation of newly introduced extended model with pulsatile circulation (Figure 11) shows approximate dynamic pressure in aorta and pulmonary artery compared to the same variables of non-pulsatile model showing rather a mean values. The values of parameters and initial values of state variables are in Table 2 and 3 using normal physiological units as well as SI units.

The simulation of reference model by Fernandez de Canete et al. (Figure 8) was performed using original values of parameters (Table 4). The pressure dynamics in aorta and pulmonary arteries (Figure 12) shows quite realistic dicrotic notch after closure of aortic valve. Detailed pressure dynamics of aorta and left ventricle during one cardiac cycle is in Figure 13.

DISCUSSION

Our non-pulsatile and pulsatile models generate a raw approximation of outgoing flowrate. However, other models, such as the model developed by Fernandez de Cante et al. [7,8], or that developed by Meurs et al. [5,6], introduce a heart with elastic vessels driven by variable compliance and valves rather than a pump with variable outgoing flowrate. The shapes of the pressure curves in these models are closer to the real experimental measurements.

In order to study more specific phenomena, additional components can be added to the model. For instance, our implementation does not use the inertia element and valves. This seems to be useful for examining the effect of a dicrotic notch in the arterial pressure when the aortic valve closes as seen, for instance, in the model and simulation developed by Fernandez de Canete et al. [7]. Figure 14 shows comparison with simple pulsatile model.

Our implementation uses one resistance and two elastic vessel components for the pulmonary and systemic circulation with total of 16 parameters and initial values of state variables (Table 2). The pulsatile model adds new 4 parameters and redefine values of three existing (Table 4). This seems satisfactory to explain basic physiological and pathophysiological phenomena, e.g. influence of congestive left or right heart failure to volumes of blood in systemic or pulmonary

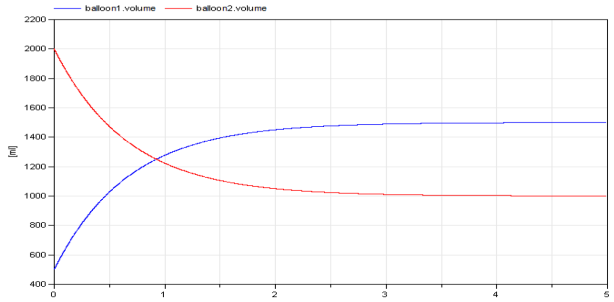


FIGURE 9 Simulation of volume dynamics from 500 ml to 1500 ml for the first balloon and from 2000 ml to 1000 ml for the second balloon. Unstressed volumes of 500 ml for both balloons and compliances of 2 ml/mmHg and 1 ml/mmHg and conductance 1 ml/(mmHg.s). After five seconds the system is almost at equilibrium

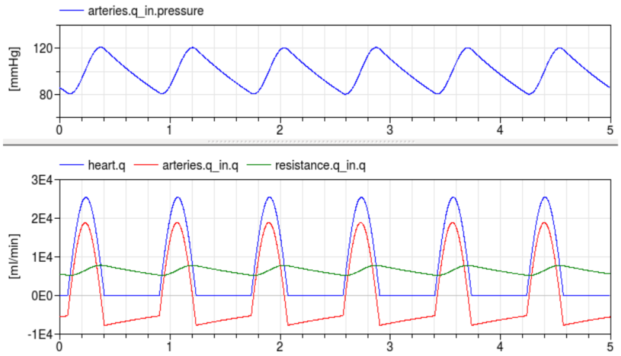


FIGURE 10 Simulation of 4-element Windkessel model characterized by impedance (approximate by conductance = 200 ml/(mmHg.s)), inertia I = 0.005 mmHg.s^2/ml, compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s) and initial volume of arteries at 0.97 l and unstressed volume of arteries at 0.85 l. The pressure during six beats is kept between 120/80 and the flowrate going from the heart ranging from 0 to 25 l per minute (blue) is compensated by the compliance compartment flowrate going from −5 to +18 l per minute (red) to the resulting average flowrate going from peripheral arteries 5 to 7 l per minute (green)

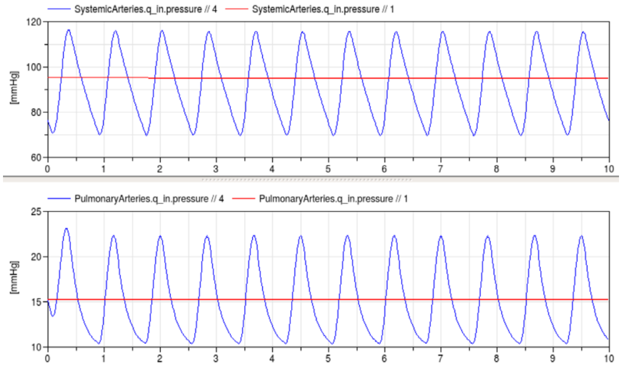


FIGURE 11 Pressure dynamics in systemic and pulmonary arteries simulated by both non-pulsatile (red) and pulsatile CVS models (red)

TABLE 2 Initial values of state variables and parameters of the non-pulsatile model of CVS in Modelica. Asterisk indicates initial volume of a component computed from steady state simulation

Parameter	Value	Value in SI	Ref.
V0CAS.k (unstressed volume of systemic arteries)	529 ml	5.29×10^{-4} m^3	
CAS.k (compliance of systemic arteries)	1.5 ml/mmHg	1.13×10^{-8} m^3/Pa	
systemicArteries.volume_start	672 ml	6.72×10^{-4} m^3	*
V0VS.k (unstressed volume of systemic veins)	2845 ml	2.845×10^{-3} m^3	
CVS.k (compliance of systemic veins)	200 ml/mmHg	1.50×10^{-6} m^3/Pa	
systemicVeins.volume_start	3922 ml	3.922×10^{-3} m^3	*
V0AP.k (unstressed volume of pulmonary arteries)	327 ml	3.27×10^{-4} m^3	
CAP.k (compliance of pulmonary arteries)	3.01 ml/mmHg	2.26×10^{-8} m^3/Pa	
pulmonaryArteries.volume_start	373 ml	3.73×10^{-4} m^3	*
V0VP.k (unstressed volume of pulmonary veins)	435 ml	4.35×10^{-4} m^3	
CVP.k (compliance of pulmonary veins)	30 ml/mmHg	2.25×10^{-7} m^3/Pa	
pulmonaryVeins.volume_start	704 ml	7.04×10^{-4} m^3	*
RT.k (total systemic resistance)	1 (mmHg.s)/ml	1.33×10^{8} (Pa.s)/m^3	
RP.k (total pulmonary resistance)	0.07 (mmHg.s)/ml	9.33×10^{6} (Pa.s)/m^3	
rightHeart.StarlingSlope	16.67 ml/(mmHg.s)	1.25×10^{-7} m^3/(Pa.s)	
leftHeart.StarlingSlope	10 ml/(mmHg.s)	7.50×10^{-8} m^3/(Pa.s)	

TABLE 3 Additional initial values of state variables and parameters of the pulsatile model of CVS in Modelica. Decreased value of CAS and changed initial volumes of systemic arteries and veins to compare with reference model. Other values inherited from non-pulsatile model (Table 2)

Parameter	Value	Value in SI	Ref.
CAS.k (compliance of systemic arteries)	0.97 ml/mmHg	7.28×10^{-9} m^3/Pa	
systemicArteries.volume_start	603 ml	6.03×10^{-4} m^3	*
systemicVeins.volume_start	3991 ml	3.991×10^{-3} m^3	*
rightHeart.pulses.QP (peak flow of right heart)	20.28 l/min	3.38×10^{-4} m^3/s	
leftHeart.pulses.QP (peak flow of left heart)	20.28 l/min	3.38×10^{-4} m^3/s	
pulses.HR (default heart rate)	72 beats per min	1.2 Hz	
Pulses.TD1 (relative time of systole start)	0.07 s	0.07 s	

TABLE 4 Initial values of state variables and parameters of the reference model by Fernandez de Canete et al. [7] in Modelica. Values with reference were taken from the original publication, asterisk indicates initial values computed from steady state simulation, other values were estimated. "ZeroPressureVolume" is unstressed volume of a component, "volume_start" is initial volume of a component. "_Gon" is conductance of a valve in opened state. Compliance and conductance values were counted as reciprocal (1/×) of original elastance and resistance values

Parameter	Value	Value in SI	Ref.
aorta.ZeroPressureVolume	30 ml	3.00×10^{-5} m^3	
aorta.Compliance	0.22 ml/mmHg	1.65×10^{-9} m^3/Pa	[7]
aorta.volume_start	46 ml	4.60×10^{-5} m^3	*
arteries.ZeroPressureVolume	700 ml	7.00×10^{-4} m^3	
arteries.Compliance	1.46 ml/mmHg	1.10×10^{-8} m^3/Pa	[7]
arteries.volume_start	805 ml	8.05×10^{-4} m^3	*
veins.ZeroPressureVolume	2370 ml	2.37×10^{-3} m^3	
veins.Compliance	20 ml/mmHg	1.50×10^{-7} m^3/Pa	[7]
veins.volume_start	2443 ml	2.44×10^{-3} m^3	*
pulmonaryArtery.ZeroPressureVolume	20 ml	2.00×10^{-5} m^3	
pulmonaryArtery.Compliance	0.09 ml/mmHg	6.75×10^{-10} m^3/Pa	[7]
pulmonaryArtery.volume_start	21 ml	2.10×10^{-5} m^3	*
pulmonaryArterioles.ZeroPressureVolume	600 ml	6.00×10^{-4} m^3	
pulmonaryArterioles.Compliance	2.67 ml/mmHg	2.00×10^{-8} m^3/Pa	[7]

Parameter	Value	Value in SI	Ref.
pulmonaryArterioles.volume_start	637 ml	6.37×10^{-4} m^3	*
pulmonaryVeins.ZeroPressureVolume	100 ml	1.00×10^{-4} m^3	
pulmonaryVeins.Compliance	46.7 ml/mmHg	3.50×10^{-7} m^3/Pa	[7]
pulmonaryVeins.volume_start	659.7 ml	6.597×10^{-4} m^3	*
heartRate.k	72 1/min	1.2 Hz	
leftVentricle.ZeroPressureVolume	90 ml	9.00×10^{-5} m^3	
leftVentricle.volume_start	209.7 ml	2.097×10^{-4} m^3	*
timeVaryingElastanceLeft.Ed	0.1 mmHg/ml	1.33×10^{7} Pa/m^3	[7]
timeVaryingElastanceLeft.Es	1.375 mmHg/ml	1.83×10^{8} Pa/m^3	[7]
timeVaryingElastanceLeft.Pi0	50 mmHg	6.66×10^{3} Pa	[7]
rightVentricle.ZeroPressureVolume	70 ml	7.00×10^{-5} m^3	
rightVentricle.volume_start	180 ml	1.80×10^{-4} m^3	
timeVaryingElastanceRight.Ed	0.03 mmHg/ml	4.00×10^{6} Pa/m^3	[7]
timeVaryingElastanceRight.Es	0.328 mmHg/ml	4.37×10^{7} Pa/m^3	[7]
timeVaryingElastanceRight.Pi0	24 mmHg	3.20×10^{3} Pa	[7]
mitralValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m^3/(Pa.s)	[7]
RleftMyo.Conductance	12.5 ml/(mmHg.s)	9.37×10^{-8} m^3/(Pa.s)	[7]
aorticValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m^3/(Pa.s)	[7]
Raorta.Conductance	14.81 ml/(mmHg.s)	1.11×10^{-7} m^3/(Pa.s)	[7]
Rsystemic.Conductance	1 ml/(mmHg.s)	7.50×10^{-9} m^3/(Pa.s)	[7]
tricuspidValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m^3/(Pa.s)	[7]
RRightMyo.Conductance	57.14 ml/(mmHg.s)	4.29×10^{-7} m^3/(Pa.s)	[7]
pulmonaryValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m^3/(Pa.s)	[7]
RPulmonaryArtery.Conductance	29.62 ml/(mmHg.s)	2.22×10^{-7} m^3/(Pa.s)	[7]
RPulmonaryVeins.Conductance	9.9 ml/(mmHg.s)	7.43×10^{-8} m^3/(Pa.s)	[7]
aorticInertia.I	8.25×10^{-4} mmHg.s^2/ml	1.10×10^{5} Pa.s^2/m^3	[7]
aorticInertia.volumeFlow_start	623.1 ml/min	1.04×10^{-5} m^3/s	*
systemicInertia.I	3.6×10^{-3} mmHg.s^2/ml	4.80×10^{5} Pa.s^2/m^3	[7]
systemicInertia.volumeFlow_start	4761 ml/min	7.94×10^{-5} m^3/s	*
pulmonaryArterialInertia.I	7.5×10^{-4} mmHg.s^2/ml	1.00×10^{5} Pa.s^2/m^3	[7]
pulmonaryArterialInertia.volumeFlow_start	43.94 ml/min	7.32×10^{-7} m^3/s	*
pulmonaryVeinsInertia.I	3.08×10^{-3} mmHg.s^2/ml	4.11×10^{5} Pa.s^2/m^{3}3	[7]
pulmonaryVeinsInertia.volumeFlow_start	1335 ml/min	2.23×10^{-5} m^3/s	*

veins. The more realistic models with additional components introduce also additional parameters and initial values for state variables. The CVS model by Fernandez de Canete et al. [7] has 48 parameters (Table 4) and this may cause some confusion when teaching only the effects described above. But these more complex models are, however, more suitable to study e.g. the phases within cardiac cycle as seen in Figure 12.

The non-pulsatile model (Figure 6) is a base of an existing educational application simple circulation available in the Atlas of Physiology and Pathophysiology (www.physiome.cz/atlas) [18] and is used in teaching physiology and pathophysiology of cardiovascular system of students of medicine and biomedical engineering[19,20]. The new pulsatile model is used in further development of the simulators to describe additionally the Windkessel effect. The simulators are introduced in lectures of pathophysiology for students of medicine and several basic scenarios are demonstrated. Students, according to surveys, use the simulators again at home when they prepare for tests and exams. This has been acknowledged as an improvement of understanding complex relation and mechanism in CVS compared to static text or non-interactive e-learning materials.

The models introduced above are used in teaching modeling and simulation of students of biomedical engineering. Because the physical system can be decomposed into basic components and modeled in an understandable form, the students focus much more on system analysis rather than on implementation issues [21].

There are some general limitations to the models based on the Windkessel phenomenon and modeling pressure volume relationship, e.g. wave transmission and wave travel cannot be studied.

CONCLUSION

Modeling technique and example models were introduced to demonstrate some effects that are important in understanding the physiology of the cardiovascular system, e.g. flow is determined by pressure gradient and Windkessel effect stabilizes the flowrate changes going from the heart to the peripheral vessels.

The implementation in Modelica language, utilizing the open source Physiolibrary, allows the expression of a complex system of differential and algebraic equations in self-descriptive model diagrams. The mathematics is hidden in the low level component model definition.

The newly introduced pulsatile model of CVS is used in further development of educational simulators.

A full source code of presented models is attached as supplementary materials to this paper and can be tried in the commercial tool Dymola or in the open source tool OpenModelica (www.openmodelica.org) using Physiolibrary (www.physiolibrary.org).

Mgr. Tomáš Kulhánek

ACKNOWLEDGMENTS

This paper describes the outcome of research that has been accomplished as part of a research program funded by the Ministry of Industry and Trade of the Czech Republic via grant FR-TI3/869; and by the Ministry of Education, Youth and Sports via the grant SVV-2014-260033.

REFERENCES

[1] Smith BW, Chase JG, Nokes RI, Shaw GM, Wake G. Minimal Haemodynamic System Model Including Ventricular Interaction and Valve Dynamics. Med Eng Phys 2004; 26(2): 131–139.

[2] Revie JA, Stevenson DJ, Chase JG, Hann CE, Lambermont BC, Ghuysen A, Kolh P, Shaw GM, Heldmann S, Desaive T. Validation of Subject — Specific Cardiovascular System Models from Porcine Measurements. Comput Methods Programs Biomed 2013; 109(2): 197–210.

[3] Hester RL, Brown AJ, Husband L, Iliescu R, Pruett D, Summers R, Coleman TG. HumMod: A Modeling Environment for the Simulation of Integrative Human Physiology. Front Physiol 2011; 2: 12.

[4] Kofránek J, Mateják M, Privitzer P. Complex Model of Integrated Physiological System — A Theoretical Basis For Medical Training Simulators. Mefanet Report 04 2011: 32–59.

[5] Van Meurs W. Modeling and Simulation in Biomedical Engineering: Applications in Cardiorespiratory Physiology. McGraw-Hill Professional 2011.

[6] Sá Couto CD, van Meurs WL, Goodwin JA, Andriessen P. A Model for Educational Simulation of Neonatal Cardiovascular Pathophysiology. Simul Healthc 2006; 1(1): 4–12.

[7] Fernandez de Canete J, del Saz-Orozco P, Moreno-Boza D, Duran-Venegas E. Object-Oriented Modeling and Simulation of the Closed Loop Cardiovascular System by Using SIMSCAPE. Comput Biol Med 2013; 43(4): 323–33.

[8] Fernandez de Canete J, Luque J, Barbancho J, Munoz V. Modelling of Long-Term and Short-Term Mechanisms of Arterial Pressure Control in the Cardiovascular System: An Object-Oriented Approach. Comput Biol Med 2014; 47: 104–112.

[9] Kofránek J. Modelica. In: Sborník Příspěvků MEDSOFT. 2013; 64–114.

[10] Hucka M, Finney A, Bornstein BJ, Keating SM, Shapiro BE, Matthews J, Kovitz BL, Schilstra MJ, Funahashi A, Doyle JC. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. Syst Biol (Stevenage) 2004; 1(1): 41–53.

[11] Butterworth E, Jardine BE, Raymond GM, Neal ML, Bassingthwaighte JB. JSim, an Open-Source Modeling System for Data Analysis. F1000Research 2014; 2: 288.

[12] Cooling M, Hunter P, Crampin E. Modelling Biological Modularity with CellML. IET Syst Biol 2008; 2(2): 73–79.

[13] Kofránek J, Mateják M, Privitzer P, Tribula M. Causal or Acausal Modeling: Labour for Humans or Labour for Machines. Tech Comput Prague 2008: 1–16.

[14] Kulhánek T, Kofránek J, Mateják M. Modeling of Short-Term Mechanism of Arterial Pressure Control in the Cardiovascular System: Object Oriented and Acausal Approach. Comput Biol Med 2014; 54(1): 137–144.

[15] Mateják M. Physiology in Modelica. Mefanet J 2014; 2(1): 10–14.

[16] Mateják M, Kulhánek T, Šilar J, Privitzer P, Ježek F, Kofránek J. Physiolibrary — Modelica Library for Physiology. In: Proc. 10th Int. Model. Conf. March 10–12, 2014, Lund, Sweden. Linköping University Electronic Press 2014: 499–505.

[17] Westerhof N, Lankhaar J-W, Westerhof BE. The Arterial Windkessel. Med Biol Eng Comput 2009; 47(2): 131–141.

[18] Kofranek J, Matousek S, Rusz J, Stodulka P, Privitzer P, Matejak M, Tribula M. The Atlas of Physiology and Pathophysiology: Web-Based Multimedia Enabled Interactive Simulations. Comput Methods Programs Biomed 2011; 104(2): 143–53.

[19] Kofránek J, Mateják M, Ježek F, Privitzer P, Šilar J. Výukový webový simulator krevního oběhu. In: Sborník Příspěvků MEDSOFT 2011: 106–121.

[20] Tribula M, Ježek F, Privitzer P, Kofránek J, Kolman J. Webový výukový simulátor krevního oběhu. In: Sborník Příspěvků MEDSOFT 2013: 197–204.

[21] Ježek F, Kroček T, Mateják M, Kofránek J. Zkušenosti Z Inovace Výuky Modelování a Simulace Na FEL ČVUT. In: Sborník Příspěvků MEDSOFT 2012: 139–146.

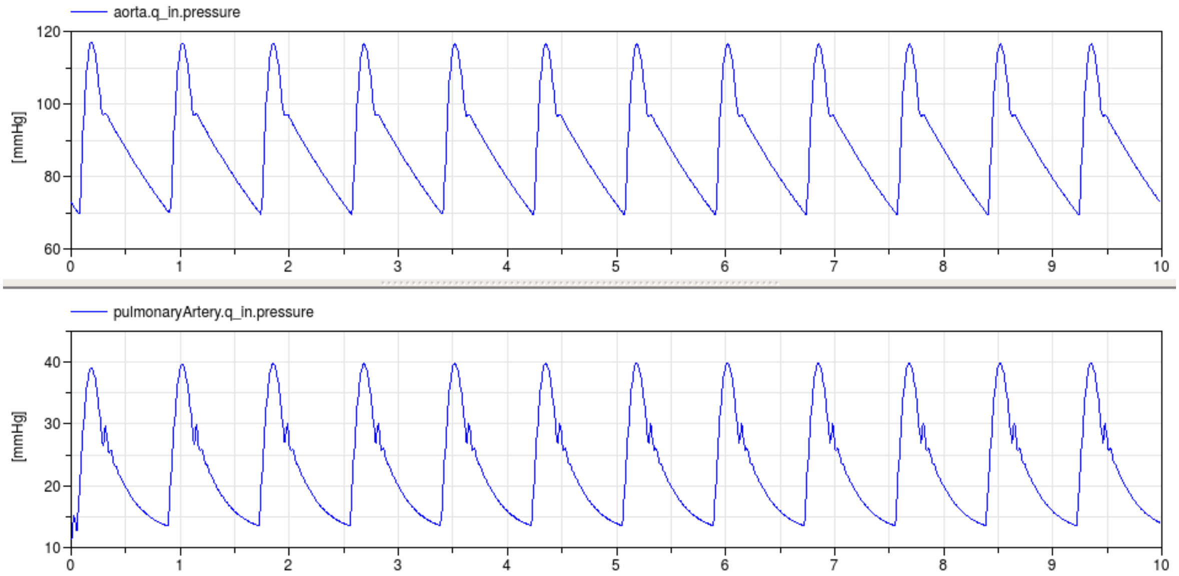


Figure 12: Pressure dynamics in systemic and pulmonary artery simulated in reference model by Fernandez de Canete et al [7].

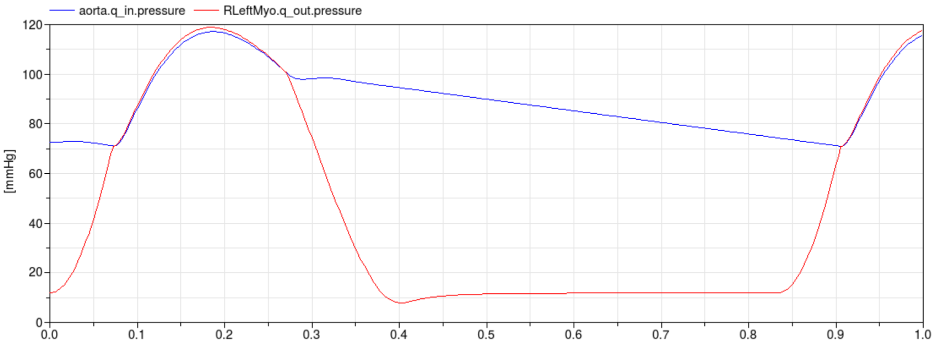


Figure 13: Pressure dynamics in aorta (blue) and left ventricle (red) simulated in reference model by Fernandez de Canete et al. [7]. At time 0.07s the aortic valve opens and the pressure in left ventricle is only slightly higher than in aorta allowing the blood to flow from left ventricle to aorta.

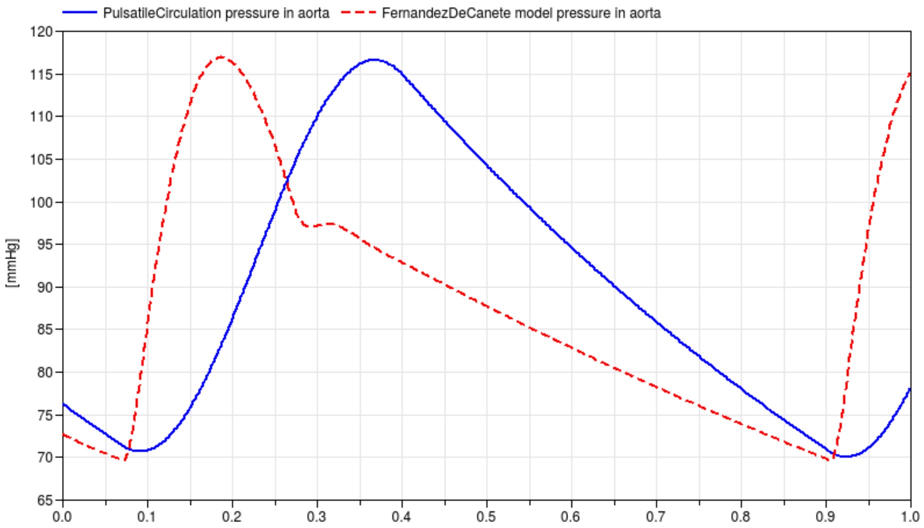


Figure 14: Comparison of pressure dynamics in systemic arteries simulated by simple pulsatile CVS models (blue) and by reference Fernandez de Canete model[7] implemented in Modelica (red dashed).

Appendix G

Adair-based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity

The paper [7] published as

M. Mateják, T. Kulhánek, and S. Matoušek. Adair-based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity. *Scandinavian Journal of Clinical \& Laboratory Investigation*, 2014. doi:10.3109/00365513.2014.984320

Available online at http://dx.doi.org/10.3109/00365513.2014.984320

The author of this thesis implemented this model in Modelica language and identified its parameters. Other co-authors analyzed and proposed a new mathematical model, based on the basic physical and chemical laws and in relation to previously published studies.

Scandinavian Journal of Clinical & Laboratory Investigation, 2015; Early Online: 1–8

informa healthcare

ORIGINAL ARTICLE

Adair-based hemoglobin equilibrium with oxygen, carbon dioxide and hydrogen ion activity

MAREK MATEJÁK, TOMÁŠ KULHÁNEK & STANISLAV MATOUŠEK

The Institute of Pathological Physiology, First Faculty of Medicine, Charles University in Prague, Czech Republic

Abstract

As has been known for over a century, oxygen binding onto hemoglobin is influenced by the activity of hydrogen ions (H^+), as well as the concentration of carbon dioxide (CO_2). As is also known, the binding of both CO_2 and H^+ on terminal valine-1 residues is competitive. One-parametric situations of these hemoglobin equilibria at specific levels of H^+, O_2 or CO_2 are also well described. However, we think interpolating or extrapolating this knowledge into an 'empirical' function of three independent variables has not yet been completely satisfactory. We present a model that integrates three orthogonal views of hemoglobin oxygenation, titration, and carbamination at different temperatures. The model is based only on chemical principles, Adair's oxygenation steps and Van't Hoff equation of temperature dependences. Our model fits the measurements of the Haldane coefficient and CO_2 hemoglobin saturation. It also fits the oxygen dissociation curve influenced by simultaneous changes in H^+, CO_2 and O_2, which makes it a strong candidate for integration into more complex models of blood acid-base with gas transport, where any combination of mentioned substances can appear.

Key Words: *Acid-base equilibrium, blood gas analysis, carboxyhemoglobin, hemoglobin A, oxyhemoglobins*

Abbreviations: ODC, hemoglobin oxygen dissociation curve; 2,3-DPG, 2,3-diphosphoglycerate; [X], molar concentration of X in $mol.m^{-3}$; aH^+, activity of hydrogen ions, where $pH = -\log_{10}(aH^+)$; αO_2, O_2 solubility in $mol.m^{-3}.Pa^{-1}$; αCO_2, CO_2 solubility in $mol.m^{-3}.Pa^{-1}$; pO_2, partial pressure of O_2 in Pa; pCO_2, partial pressure of CO_2 in Pa; Hb_u, hemoglobin alpha or beta subunit; Hb_{uD}, deoxygenated Hb_u; Hb_{uO}, oxygenated Hb_u; $Hb_uNH_3{}^+$, Hb_u with protonated Nterminus; $Hb_{uD}NH_3{}^+$, Hb_{uD} with protonated Nterminus; $Hb_{uO}NH_3{}^+$, Hb_{uO} with protonated Nterminus; Hb_uNH_2, Hb_u with $-NH_2$ form of Nterminus; $Hb_{uD}NH_2$, Hb_{uD} with $-NH_2$ form of Nterminus; $Hb_{uO}NH_2$, Hb_{uO} with $-NH_2$ form of Nterminus; Hb_uCOO^-, Hb_u with carboxylated Nterminus; $Hb_{uD}COO^-$, Hb_{uD} with carboxylated Nterminus; $Hb_{uO}COO^-$, Hb_{uO} with carboxylated Nterminus; Hb_uAH, Hb_u with protonated side-chains; $Hb_{uD}AH$, Hb_{uD} with protonated side-chains; $Hb_{uO}AH$, Hb_{uO} with protonated side-chains; Hb_uA^-, Hb_u with deprotonated side-chains; $Hb_{uD}A^-$, Hb_{uD} with deprotonated side-chains; $Hb_{uO}A^-$, Hb_{uO} with deprotonated side-chains; $Hb_uA^-NH_2$, selected normalized form of Hb_u with deprotonated side-chains and NH_2 form of Nterminus; $Hb_{uD}A^-NH_2$, deoxygenated form of $Hb_uA^-NH_2$; $Hb_{uO}A^-NH_2$, oxygenated form of $Hb_uA^-NH_2$; f_{nD}, fraction of $Hb_{uD}A^-NH_2$ from Hb_{uD}; f_{nO}, fraction of $Hb_{uO}A^-NH_2$ from Hb_{uO}; f_{zcD}, fraction of $Hb_{uD}NH_2$ form Hb_{uD}; f_{zcO}, fraction of $Hb_{uO}NH_2$ form Hb_{uO}; f_{hD}, fraction of $Hb_{uD}A^-$ form Hb_{uD}; f_{hO}, fraction of $Hb_{uO}A^-$ form Hb_{uO}; $\Delta H+_h$, change of valence (charge) on side-chains during deoxygenation per one Hb_u; $\Delta H+_{\mathbf{z}}$, protonation of $-NH_2$ form of Nterminus during deoxygenation per one Hb_u; $\Delta H+_c$, decarboxylation of carboxylated Nterminus during deoxygenation per one Hb_u; ΔH^+, Haldane coefficient per hemoglobin subunit; Hb_t, hemoglobin tetramer without bound O_2 molecules; $(O_2)_iHb_t$, hemoglobin tetramer with the number of i bound O_2 molecules; $(O_2)_iHb_{tn}$, hemoglobin tetramer composed only of $Hb_uA^-NH_2$ subunit forms with the number of i bound O_2 molecules; sO_2, O_2 saturation of hemoglobin; sCO_{2D}, CO_2 saturation of deoxyhemoglobin; sCO_{2O}, CO_2 saturation of oxyhemoglobin; sCO_2, CO_2 saturation of hemoglobin; tCO_2, total concentration of CO_2 = free dissolved $CO_{2\,+}$$+HCO_3{}^-$ $+CO_3{}^{2-}$ $+Hb_uCOO^-$; dTH, shift of titration curve with the change of both pO_2 and pCO_2 to zero, which equals to how many moles of strong acid must be added to one mole of Hb_u to reach the pH of the chosen standard titration curve (O_2 and CO_2 free); pH, acidity/basicity of hemoglobin solution (e.g. inside erythrocytes); pH_p, pH in plasma (e.g. acidity/basicity of blood).

Correspondence: Marek Mateják, U Nemocnice 5, Prague 2, 128 53, Czech Republic. Tel: + 420 776 301 395. Fax: + 420 224 965 916. E-mail: marek.matejak@lf1.cuni.cz

(Received 20 June 2014; accepted 2 November 2014)

ISSN 0036-5513 print/ISSN 1502-7686 online © 2015 Informa Healthcare
DOI: 10.3109/00365513.2014.984320

Introduction

Human hemoglobin A is one of the most extensively studied protein macromolecules. The composition and 3D conformation of both α and β chains is known [1,2] and the binding of O_2, H^+ and 2,3-DPG has been described [2,3]. Since the 1930s it has generally been believed that CO_2 binding to Hb occurs by carbamination of the amino-terminus, i.e. forming a carboxylate compound [4–6]. This hemoglobin carbamination was verified by Morrow et al. [7], who used nuclear magnetic resonance of $^{13}CO_2$ to find its exact binding sites at valine-1. The shift of titration between oxygenated and deoxygenated forms is also well known. Called the Bohr or Haldane effect [8–10], it is caused by the same valine-1 side and by more than 10 other acid-base residues [2,11].

The most common descriptions of the hemoglobin oxygen dissociation curve (ODC) are the allosteric models [12,13], the model based on Hill equation [14,15], and Adair's four-step model [16]. Some of the ODC models [15,17,18] also include the effects of varying pH and CO_2 concentration. However, these models operate only with interpolation or extrapolation of ODC from normal pH or normal pCO_2 and do not take into account the chemical dependences between titration [8] and carbamination [19]. They fail when both pH and CO_2 are not at normal values, especially in the alkaline pH range.

Based on these findings, we propose a hemoglobin binding equilibrium model that starts with a description of hemoglobin-oxygen dissociation using a slightly modified version of Adair's approach [16]. We continue by describing the relationship between CO_2 and H^+ as competitive inhibition in the amino group of terminal valine-1 residue on each chain (suffixes *z* and *c*), which is in accordance with the known facts [4,6,20]. Finally, we lump all other acid-base side chains residues in a hemoglobin subunit into one Bohr proton-binding site of the side chain residues. These are denoted by the suffix *h* in the article.

Methods

The model is built in Mathematica 9.0 (Wolfram, Champaign, IL, USA) and also in Dymola FD01 2014 (Dassault Systemes, Paris, France) as an example of chemical package in open-source Modelica library Physiolibrary 2.2.0 (http://www.physiolibrary.org) [21,22] according to the model structure defined in the following section. The model contains only physiological parameters such as gas solubilities and dissociation coefficients of defined reactions. The unknown parameters are fitted to the data of Siggaard-Andersen [23], Bauer and Schröder [24], Severinghaus [18], Matthew et al. [1] and Reeves [25] using Mathematica function FindFit and also using parameter estimation method suggested by Kulhanek et al. [26].

Free dissolved concentrations of $[O_2]$ and $[CO_2]$ in red blood cells are calculated from gas partial pressures using Henry's law ($[O_2] = \alpha O_2 * pO_2$ and $[CO_2] = \alpha CO_2 * pCO_2$) with solubility coefficients $\alpha O_2 = 1.005*10^5$ $mol.m^{-3}.Pa^{-1}$ measured at 38°C by Sendroy et al. [27], and $\alpha CO_2 = 2.3*10^4$ $mol.m^{-3}.Pa^{-1}$ at 37°C, as proposed by Maas et al. [28]. These solubilities are slightly different from solubilities in pure water or in plasma because of the effects of salts and proteins inside erythrocyte [27].

Model structure

The model is built upon several simplifying assumptions. Firstly, each of four hemoglobin tetramer subunits is treated as identical, even though (slight) differences are known to exist between α and β subunits. Secondly, CO_2 or H^+ binding is supposed not to affect the CO_2 or H^+ affinities in the other three subunits. Thus, the interaction between the subunits is modeled purely through the varying affinities of oxygen in each oxygenation step. The third simplifying assumption has already been mentioned; it is lumping all Bohr proton binding sites of subunit into two (first for the valine-1 amino-terminus and second for the side chains residues), a simplification first suggested by Antonini [29].

Model structure of hemoglobin subunit

The three reactions that participate in the Bohr and Haldane effect of each subunit are as follows:

Reaction z: $Hb_uNH_3{}^+ \leftrightarrow Hb_uNH_2 + H^+$
Reaction c: $Hb_uNH_2 + CO_2 \leftrightarrow Hb_uCOO^- + H^+$
Reaction h: $Hb_uAH \leftrightarrow Hb_uA^- + H^+$

where the reactions z and c are competitive on the valine-1 amino-terminus, and the reaction h is independent of z or c.

The chemical equilibrium equations of these three reactions are Equations 1–3, where K_x is the equilibrium dissociation coefficients of the reaction *x* (i.e. *z*, *c* or *h*).

$$\text{Reaction z: } K_z = \frac{[Hb_uNH_2]*aH^+}{[Hb_uNH_3^+]} \quad (1)$$

$$\text{Reaction c: } K_c = \frac{[Hb_uCOO^-]*aH^+}{[CO_2][Hb_uNH_2]} \quad (2)$$

$$\text{Reaction h: } K_h = \frac{[Hb_uA^-]*aH^+}{[Hb_uAH]} \quad (3)$$

These dissociation coefficients are different between oxy and deoxy subunits, which are distinguished by the subscripts O and D in the following text. They can be also written in their logarithmic form, where $\mathbf{pK_x}$ means $-\log_{10}(K_x)$. Thus, for instance, pK_{zO} denotes the equilibrium coefficient of the reaction z for the oxy form of the hemoglobin subunit. Similar notation is used for describing the activity of hydrogen ions (acidity), where $pH = -\log_{10}(aH^+)$.

Using Equations 1–3 it is possible to express fractions of chosen species for deoxy and oxy subunits. We label these fractions as follows: Hb_uNH_2 fractions are called $\mathbf{f_{zcD}}$ ($\mathbf{f_{zcO}}$), Hb_uA^- fractions $\mathbf{f_{hD}}$ ($\mathbf{f_{hO}}$), $Hb_uA^-NH_2$ fractions $\mathbf{fn_D}$ ($\mathbf{fn_O}$) and Hb_uCOO^- fractions $\mathbf{sCO_{2D}}$ ($\mathbf{sCO_{2O}}$) as in Equation 4–7. The selection of form $Hb_uA^-NH_2$ from the orthogonal division into Hb_uNH_2 and Hb_uA^- is also illustrated in Figure 1A, B.

$$f_{\mathrm{zcD}} = \frac{[Hb_{uD}NH_2]}{[Hb_{uD}]} = \frac{1}{1+10^{\mathrm{pKzD}-pH}+[CO_2]10^{pH-pKcD}} \tag{4}$$

$$f_{\mathrm{hD}} = \frac{[Hb_{uD}A^-]}{[Hb_{uD}]} = \frac{1}{10^{\mathrm{pKhD}-pH}+1} \tag{5}$$

$$f_{n\mathrm{D}} = \frac{[Hb_{uD}A^-NH_2]}{[Hb_{uD}]} = f_{\mathrm{zcD}} \times f_{\mathrm{hD}} \tag{6}$$

$$\mathrm{sCO_{2D}} = \frac{[Hb_{uD}COO^-]}{[Hb_{uD}]} = 10^{pH-pKcD} f_{\mathrm{zcD}}[CO_2] \tag{7}$$

We define a titration shift as the amount of acid that must be added to achieve the same pH after full deoxygenation of the hemoglobin subunit. This change of subunit charge during deoxygenation (by the Bohr protons) is also called Haldane coefficient ΔH^+. The coefficient can be divided into contributions of the previously mentioned reactions $\Delta H^+{}_h$, $\Delta H^+{}_z$ and $\Delta H^+{}_c$, as is algebraically expressed by Equations 8–11.

$$\Delta H_h^+ = -\frac{[Hb_{uD}A^-]-[Hb_{uO}A^-]}{[Hb_u]} = -(f_{\mathrm{hD}} - f_{\mathrm{ho}}) \tag{8}$$

$$\begin{aligned}\Delta H_z^+ &= \frac{[Hb_{uD}NH_3^+]-[Hb_{uo}NH_3^+]}{[Hb_u]} \\ &= \left(\frac{10^{\mathrm{pKzD}}}{10^{pH}} f_{\mathrm{zcD}} - \frac{10^{\mathrm{pKzO}}}{10^{pH}} f_{\mathrm{zcO}}\right)\end{aligned} \tag{9}$$

$$\begin{aligned}\Delta H_c^+ &= -\frac{[Hb_{uD}COO^-]-[Hb_{uO}COO^-]}{[Hb_u]} \\ &= -(\mathrm{sCO_{2D}} - \mathrm{sCO_{2O}})\end{aligned} \tag{10}$$

$$\Delta H^+ = \Delta H_z^+ + \Delta H_c^+ + \Delta H_h^+ \tag{11}$$

Model structure of hemoglobin tetramer

Chemical speciation of the hemoglobin tetramer molecule can be considered at various levels of detail; the one chosen as appropriate in our approach is indicated in Figure 1C. The possible forms include different combinations of oxygenated and deoxygenated

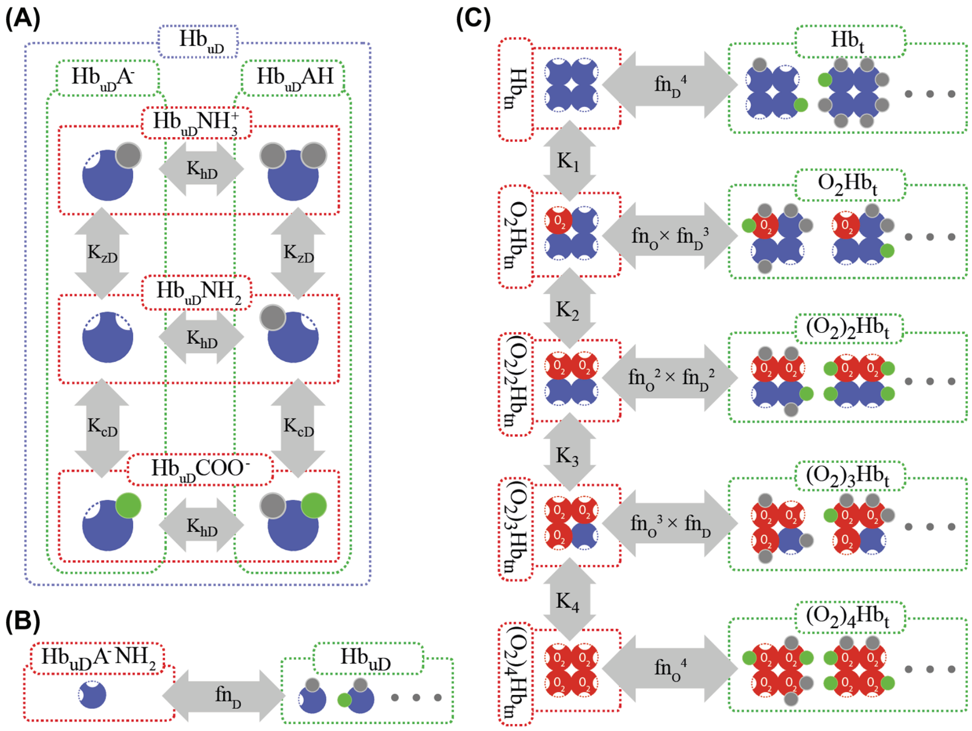


Figure 1. Schema of hemoglobin calculation. (A) Possible forms of deoxyhemoglobin subunit (the blue circles). The gray circles represent hydrogen ions, and the green circles represent carbon dioxide. The arrows with dissociation coefficients represent the reactions z, c and h. (B) Schema of chemical speciation of deoxyhemoglobin subunit. (C) Schema of chemical speciation and Adair's oxygenation steps of hemoglobin tetramer. This Figure is reproduced in color in the online version of *The Scandinavian Journal of Clinical & Laboratory Investigation.*

hemes, protonated and deprotonated oxygen-linked acid groups, and free or carboxylated amino endings of each chain. Yet because of the law of detailed balance in equilibrium [30], it is not necessary to calculate all reactions between these forms. As a result, only four oxygenation reactions need be selected for the calculation. This is done in accordance with Figure 1C, by selecting a tetramer form $(O_2)_iHb_{tn}$ composed only of four subunits in the form $Hb_uA^-NH_2$ (H^+ and CO_2 free), and modeling oxygen binding to these forms with Adair-type coefficients of the following reactions:

$$O_2Hb_{tn} \leftrightarrow Hb_{tn} + O_2$$
$$(O_2)_2Hb_{tn} \leftrightarrow O_2Hb_{tn} + O_2$$
$$(O_2)_3Hb_{tn} \leftrightarrow (O_2)_2Hb_{tn} + O_2$$
$$(O_2)_4Hb_{tn} \leftrightarrow (O_2)_3Hb_{tn} + O_2$$

where dissociation coefficients are defined by Equation 12, which are also represented by vertical arrows at Figure 1C.

$$K_i = \frac{[(O_2)_{i-1}Hb_{tn}]\cdot[O_2]}{[(O_2)_iHb_{tn}]} \qquad (12)$$

For the next calculation, all forms can be expressed as a fraction of the deoxy-tetramer form $(O_2)_0Hb_{tn}$, as shown in Equation 13.

$$\left[(O_2)_iHb_{tn}\right] = \frac{[(O_2)_0Hb_{tn}]\cdot[O_2]^i}{\prod_{j=1}^{i}K_i} \qquad (13)$$

Let us move the attention from specific forms of Hb_{tn} to the description of the equilibrium within the whole group of Hb_t. Looking at Figure 1C, one can see that for each oxygenation step we can calculate the equilibrium in each horizontal line using Equation 14.

$$[(O_2)_iHb_t]\cdot fn_D^{\,4-i}\cdot fn_O^{\,i} = [(O_2)_iHb_{tn}] \qquad (14)$$

The CO_2- and pH-dependent oxygen saturation equation in the Adair style (Equation 15) is algebraically derived from Equation 13–14, where $a_i = \frac{1}{\left(\prod_{j=1}^{i}K_j\right)}$ $x = (fn_D(pH,[CO_2])\,/\,fn_O(pH,[CO_2]))*[O_2]$.

$$sO_2 = \frac{a_1x + 2\,a_2x^2 + 3\,a_3x^3 + 4\,a_4x^4}{4 + 4\,a_1x + 4\,a_2x^2 + 4\,a_3x^3 + 4\,a_4x^4} \qquad (15)$$

Hemoglobin saturation with CO_2 (sCO_2) is calculated separately in oxygenated and deoxygenated subunits forms (sCO_{2O} and sCO_{2D}) using Equation 16.

$$sCO_2 = sO_2\cdot sCO_{2O} + (1 - sO_2)\cdot sCO_{2D} \qquad (16)$$

Finally, the shift of titration after deoxygenation and decarbamination of the hemoglobin subunit can be expressed by Equation 17.

$$dTH = sO_2\cdot\Delta H^+ + sCO_{2D} + \frac{sCO_{2D}}{1+10^{pH-pKzD}} \qquad (17)$$

Temperature dependences

The temperature dependences are integrated using the Van’t Hoff equation (Equation 18), where $R = 8.314\ J.K^{-1}.mol^{-1}$ is gas constant, ΔH^θ is the standard enthalpy change, i.e. an amount of heat consumed by a reaction changing one mole of substrates to products, and K_2 and K_1 are Henry’s coefficients of solution or the dissociation coefficient of the chemical reaction at temperature T_2 and T_1 (expressed in Kelvin).

$$\ln\left(\frac{K_2}{K_1}\right) = \frac{-\Delta H^\theta}{R}\left(\frac{1}{T_2} - \frac{1}{T_1}\right) \qquad (18)$$

Results

The Adair’s coefficients are fitted to the collection of ODC measurements by Severinghaus [18] at $pCO_2 = 0$ Pa and $pH_p = 7.4$, see Figure 2A and Table I. The dissociation coefficients of carboxylation are determined in close agreement with Bauer and Schröder using their data [24]; the resulting fit can be seen in Figure 2B, the coefficients are in Table II. The lumped acid dissociation coefficients for the side-chains pK_{hD} and pK_{hO} are estimated by optimization of Siggaard-Andersen’s data [23] at DPG/Hbt = 0.84, pH = 6.5–8.0 and 37°C; the resulting fit can be seen in Figure 2C, and the coefficients complete Table II.

We conclude that the model also describes the ODC shifts by comparing with Naerea et al.’s [31] oxygen saturation measurements at different plasma pH and CO_2 levels, see Figure 2D. The recalculation of data from plasma pH_p to intracellular erythrocyte pH uses the equation $pH = 7.2464 + 0.796(pH_p - 7.4)$, as presented by Siggaard-Andersen and Salling [32].

The gas solubility as Henry’s coefficient at different temperatures can be recalculated using the enthalpy of gas solution (− 14 kJ.mol^{-1} for O_2, and − 20 kJ.mol^{-1} for CO_2 [33]). If we assume that the examined hemoglobin of Matthew et al. [1] is at compatible conditions, but at a temperature of 30°C with $pK_{zD} = 7.53$, $pK_{zO} = 7.28$, $pK_{cD} = 4.77$ and $pK_{cO} = 5.20$ as plotted in Figure 2E, then the enthalpies of these reactions are − 51, 8, 59 and − 41 kJ.mol^{-1}.

Atha and Ackers measured the heat of hemoglobin oxygenation under conditions independent of carbon dioxide and Bohr protons and not including the oxygen heat of solution, coming to the value of 59 kJ.mol^{-1} [34]. Using this value for each Adair oxygenation step, one can optimize other enthalpies to fit the Reeves data [25] measured at 19–43°C (Figure 2F). Resulting enthalpies are 59 kJ.mol^{-1} for the deoxy and 127 kJ.mol^{-1} for the oxy version of reaction h.

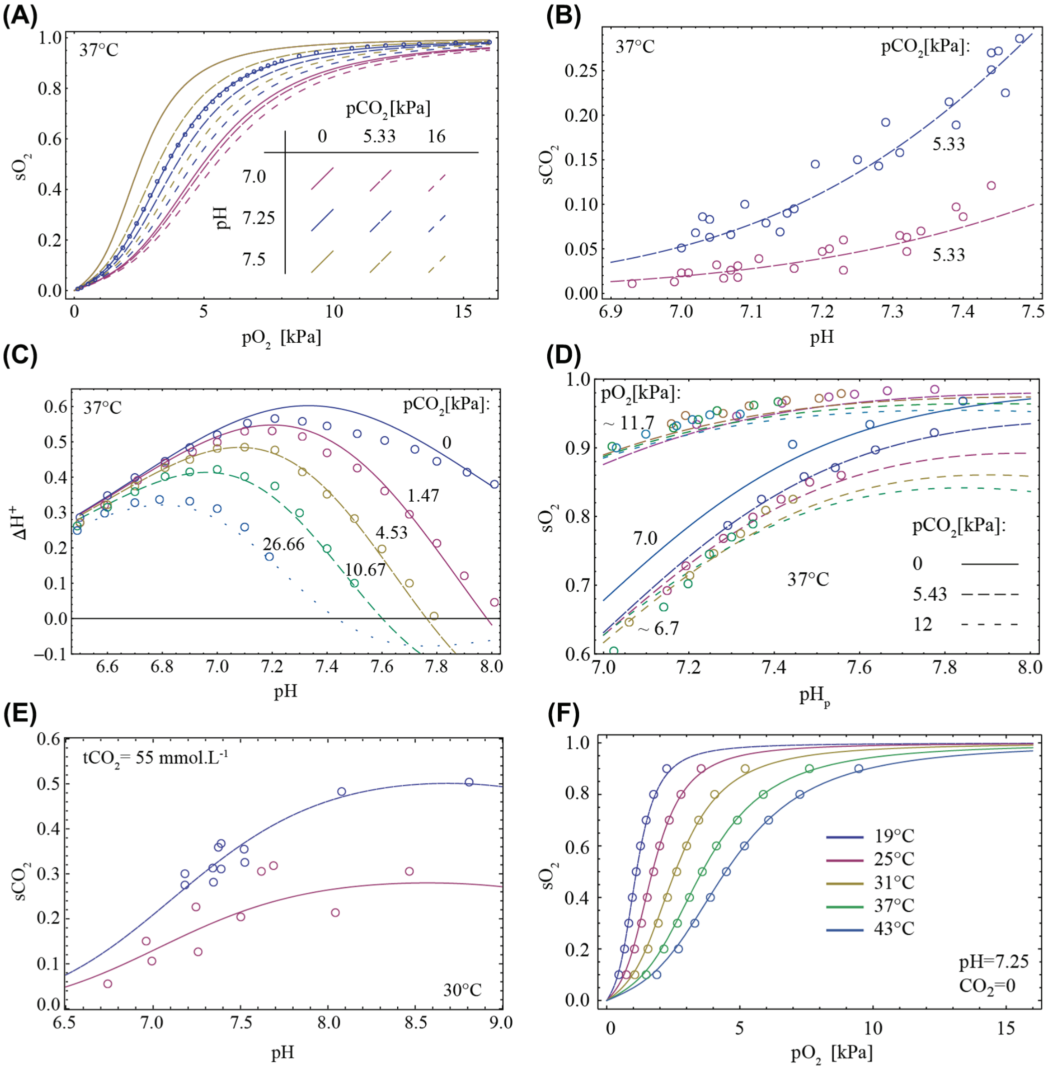


Figure 2. Model curves and measured data points. (A) Points are from Severinghaus's oxygen dissociation curve collection [18] at $pCO_2 = 0$ Pa, erythrocyte intracellular pH = 7.25 and temperature 37°C. Lines are sO_2 defined by Equation 15. (B) Carboxylation of hemoglobin measured and estimated by Bauer and Schröder [24]. The lines (Equation 7) are sCO_{2D} and sCO_{2O} at $pCO_2 = 5.33$ kPa. (C) Bohr protons released during oxygenation of one hemoglobin subunit. Dots are data measured by Siggaard-Andersen [23] in erythrolysate at 37°C, $DPG/Hb_t = 0.84$. Lines are calculated from Equation 11 using coefficients from Table II. From top to bottom data are plotted for different $pCO_2 = 0$ kPa, 1.47 kPa, 4.53 kPa, 10.67 kPa and 26.66 kPa. (D) Naeraa et al.'s [31] oxygen saturation data comparison. The two groups of lines (Equation 15) are determined mainly by pO_2 around 6.7 kPa and 11.7 kPa. The small nuances in each group are caused by different pCO_2 changed from 0–12 kPa. (E) Matthew et al.'s [1] carbamination data measured at 30°C with constant content of all carbonates 55 mmol.L^{-1}. Fitted dissociation constants, which determine the lines, are used to estimate enthalpies of their reactions to be consistent with Bauer and Schröder [24]. (F) Reeves [25] oxygen saturation data at different temperatures. Enthalpies of reaction h and specific Adair oxygenation step enthalpy is estimated (lines) to fit the data (circles). This Figure is reproduced in color in the online version of *The Scandinavian Journal of Clinical & Laboratory Investigation.*

Discussion

Having a precise quantitative description of hemoglobin behavior is a crucial aspect in describing the behavior of whole blood, which has numerous uses in medicine today. It is important in areas such as blood gas analysis [15,34], the building of complex models [35], simulation for teaching purposes [36], or rebreathing-based methods for cardiac output estimation, which represent a more specific use [37]. For instance, the precision of the latter methods is crucially dependent on the exact calculation of the total amount of carbon dioxide in blood for various

Table I. Estimated form-specific Adair's coefficients [mol.m^{-3}] at 37°C.

K_1	K_2	K_3	K_4
0.0121	0.0117	0.0871	0.000386

Table II. Estimated acid dissociation coefficients at 37°C.

Reaction z	Reaction c	Reaction h
$pK_{zD} = 7.73$	$pK_{cD} = 7.54$	$pK_{hD} = 7.52$
$pK_{zO} = 7.25$	$pK_{cO} = 8.35$	$pK_{hO} = 6.89$

(abnormal) patient conditions, as has been recently pointed out [38].

This model offers a precise description of various phenomena that take place with hemoglobin, based on relatively simple starting points, such as competitive binding of H+ and CO_2 at valine-1 amino terminus or the law of detailed balance [30]. Even with a simple structure, it offers a remarkably good fit to the data of hemoglobin oxygenation, titration, and carbamination at different temperatures, as shown in Figure 2A–E. These can be calculated with any combination of oxygen, carbon dioxide and hydrogen ions defined as open system (lungs), where the partial pressures are equilibrated, and also in a closed system (tissues), where mass conservation laws and the total amount of substances take place, as was also modeled by Rees and Andreassen [39], among others.

The results of our model (Figure 2A–E) show strong nonlinear dependences between variables, which is in agreement with the known data [1,23–25,40]. Looking at Figure A, one can compare sets of ODCs, where each color represents a different pH. Various curves of each color represent ODCs for various levels of pCO_2 for a given pH. As can be seen, the effect of pCO_2 on the ODC is stronger at high (alkaline) pH, which is in agreement with the data [23]. Similarly, one can compare the curves within the sets of solid or dashed lines, where each set represents dissociation curves for various values of pH at a given level of pCO_2. The variation between the curves of each line type represents the Bohr effect for the given level of CO_2, this effect can also be appreciated from data of Figure 2C, which shows the average amount of released H^+ upon oxygenation of one hemoglobin subunit.

The model uses enthalpies to calculate temperature dependences of hemoglobin behavior (Figure 2F, E), which allows examination of the heat transfers during single chemical processes. For instance, binding of aqueous oxygen onto hemoglobin produces 30–40 kJ/mol of heat [41–44], and the same amount of heat is consumed by deoxygenation process in metabolically active tissues, thus helping to cool them down. When the hemoglobin model is used in the standard conditions of a large-scale HumMod model [35], the resulting heat transfer due to the exothermy of the hemoglobin oxygen reaction is 4–7% of the total heat produced by muscle, which is in agreement with experimental results [41,43,44]. It is interesting to note that this heat transfer occurs without any increase in blood temperature.

The integration of chemical processes in a macromolecule requires a precise view into their underlying principles. Some physiologists use the elementary chemical equations [39], but do not implement the principle of detailed balance [30]. Other physiologists make empirically-based equations with a raw linear gradient approximation of possible combinations of model values [15]. We feel that it is almost impossible to see the problem as this type of blackbox function with more than two inputs. Instead, it is better to have an integrated model of oxygenation, titration and carbamination.

Today, allosteric hemoglobin oxygenation models do exist that seem more in agreement with the structural knowledge of hemoglobin [12,45–47]. These models take into account two or more structurally different forms: relaxed and tensed. However, these models have so far been limited to hemoglobin oxygenation only. Our Adair-based model can explain not only oxygen and carbon dioxide saturation, but also their cooperation with acid-base buffering properties of hemoglobin. All three of these connected phenomena fit to measured data in physiological ranges.

As any work, the presented model has its limitations. First of all, it does not include the effects of changes in Hb tetramer conformations. Also the model could be extended with dependences on electrolytes such as chloride, 2,3-bisphosphoglycerate or other organic phosphates and their binding reactions, as many research studies show these interactions [48–51]. The next extension of this model could be performed by the integration of an intracellular red cell environment to calculate with phosphate acid-base buffers, and finally the membrane changes with blood plasma, where the chloride shift reaches a Gibbs-Donnan equilibrium and establishes chloride, bicarbonate and hydrogen ion activity ratios. Having an integrated model of blood gases and acid-base is crucial if we want the precise computational algorithms of the current state of a patient. These calculations could be used, for example, inside the next generation of medical devices to estimate not only blood properties, but also the connected properties of circulation [37] or metabolic functions [39].

In this article, we present a hemoglobin model that integrates O_2, CO_2 and H^+ binding. The model is not just empirical, but is based on sound theoretical principles, such as the competitive binding of CO_2 and H^+ on the valine-1 NH_2 terminus, the Bohr and Haldane effect [9,52] and on the principle of detailed balance. The principle of detailed balance is used for the first time in the Adair type of model. The advantages of this approach include explicit formulation of mass and energy conservation principles: The model accumulates substances and heat inside hemoglobin forms, making it very useful for integration in higher-scale dynamic models.

Acknowledgements

In addition, the authors want to thank doc. MUDr. Jiří Kofránek, CSc, and Prof. MUDr. Emanuel Nečas, DrSc, for critical review and Austin Schaefer for English language editing.

This work was supported by grants from the Ministry of Industry and Trade of the Czech Republic: MPO TIP TI3/869 – Virtual patient for medical education; and by The Ministry of Education, Youth and Sports via the grant SVV-2014-260033.

Declaration of interest: The authors report no conflict of interest. The authors alone are responsible for the content and writing of the paper.

References

[1] Matthew JB, Morrow J, Wittebort RJ, Gurd F. Quantitative determination of carbamino adducts of alpha and beta chains in human adult hemoglobin in presence and absence of carbon monoxide and 2, 3-diphosphoglycerate. J Biol Chem 1977;252:2234–44.

[2] Perutz M. Species adaptation in a protein molecule. Mol Biol Evol 1983;1:1–28.

[3] Perutz M, Kilmartin J, Nishikura K, Fogg J, Butler P, Rollema H. Identification of residues contributing to the Bohr effect of human haemoglobin. J Mol Biol 1980;138: 649–68.

[4] Rossi-Bernardi L, Roughton F. The specific influence of carbon dioxide and carbamate compounds on the buffer power and Bohr effects in human haemoglobin solutions. J Physiol 1967;189:1–29.

[5] Forster R, Constantine H, Craw MR, Rotman H, Klocke R. Reaction of CO2 with human hemoglobin solution. J Biol Chem 1968;243:3317–26.

[6] Stadie WC, O'Brien H. The carbamate equilibrium I. The equilibrium of amino acids, carbon dioxide, and carbamates in aqueous solution; with a note on the Ferguson-Roughton carbamate method. J Biol Chem 1936;112:723–58.

[7] Morrow J, Matthew J, Wittebort R, Gurd F. Carbon 13 resonances of 13CO2 carbamino adducts of alpha and beta chains in human adult hemoglobin. J Biol Chem 1976;251:477–84.

[8] Bohr C, Hasselbalch K, Krogh A. Concerning a biologically important relationship–the influence of the carbon dioxide content of blood on its oxygen binding. Skand Arch Physiolog 1904;16:402.

[9] Siggaard-Andersen O, Garby L. The Bohr effect and the Haldane effect. Scand J Clin Lab Invest 1973;31:1–8.

[10] Tyuma I. The Bohr effect and the Haldane effect in human hemoglobin. Jpn J Physiol 1984;34:205.

[11] Matthew J, Hanania G, Gurd F. Electrostatic effects in hemoglobin: hydrogen ion equilibria in human deoxy-and oxyhemoglobin A. Biochemistry 1979;18:1919.

[12] Monod J, Wyman J, Changeux J-P. On the nature of allosteric transitions: a plausible model. J Mol Biol 1965;12:88–118.

[13] Eaton WA, Henry ER, Hofrichter J, Bettati S, Viappiani C, Mozzarelli A. Evolution of allosteric models for hemoglobin. IUBMB Life 2007;59:586–99.

[14] Hill AV. The combinations of haemoglobin with oxygen and with carbon monoxide. I. Biochem J 1913;7:471.

[15] Siggaard-Andersen O, Siggaard-Andersen M. The oxygen status algorithm: a computer program for calculating and displaying pH and blood gas data. Scand J Clin Lab Invest 1990;50:29–45.

[16] Adair GS. The hemoglobin system VI. The oxygen dissociation curve of hemoglobin. J Biol Chem 1925;63:529–45.

[17] Dash RK, Bassingthwaighte JB. Erratum to: Blood HbO_2 and $HbCO_2$ dissociation curves at varied O_2, CO_2, pH, 2, 3-DPG and temperature levels. Ann Biomed Eng 2010; 38:1683–701.

[18] Severinghaus JW. Simple, accurate equations for human blood O_2 dissociation computations. J Appl Physiol 1979; 46:599–602.

[19] Christiansen J, Douglas C, Haldane J. The absorption and dissociation of carbon dioxide by human blood. J Physiol 1914;48:244–71.

[20] Perrella M, Bresciani D, Rossi-Bernardi L. The binding of CO2 to human hemoglobin. J Biol Chem 1975;250: 5413–8.

[21] Mateják M, Kulhánek T, Šilar J, Privitzer P, Ježek F, Kofránek J. Physiolibrary-Modelica library for Physiology. 10th International Modelica Conference; 2014 March 12; Lund, Sweden: Linköping University Electronic Press; 2014. pp. 499–505.

[22] Mateják M. Physiology in Modelica. Mefanet J 2014; 2:10–4.

[23] Siggaard-Andersen O. Oxygen-linked hydrogen ion binding of human hemoglobin. effects of carbon dioxide and 2, 3-diphosphoglycerate I. Studies on erythrolysate. Scand J Clin Lab Invest 1971;27:351–60.

[24] Bauer C, Schröder E. Carbamino compounds of haemoglobin in human adult and foetal blood. J Physiol 1972;227: 457–71.

[25] Reeves RB. The effect of temperature on the oxygen equilibrium curve of human blood. Respir Physiol 1980; 42:317–28.

[26] Kulhanek T, Matejak M, Silar J, Kofranek J. Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud. Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on 1–4 June 2014; pp. 712–5.

[27] Sendroy J, Dillon RT, Van Slyke DD. Studies of gas and electrolyte equilibria in blood XIX. The solubility and physical state of uncombined oxygen in blood. J Biol Chem 1934;105:597–632.

[28] Maas A, Rispens P, Siggaard-Andersen O, Zijlstra W. On the reliability of the Henderson-Hasselbalch equation in routine clinical acid-base chemistry. Ann Clin Biochem 1984;21:26.

[29] Antonini E, Wyman J, Brunori M, Fronticelli C, Bucci E, Rossi-Fanelli A. Studies on the relations between molecular and functional properties of hemoglobin V. The influence of temperature on the Bohr effect in human and in horse hemoglobin. J Biol Chem 1965;240:1096–103.

[30] Sauro HM. Enzyme kinetics for systems biology. Future Skill Software (Ambrosius Publishing); 2011. pp. 20–1.

[31] Naeraa N, Petersen ES, Boye E. The influence of simultaneous, independent changes in pH and carbon dioxide tension on the in vitro oxygen tension-saturation relationship of human blood. Scand J Clin Lab Invest 1963;15:141–51.

[32] Siggaard-Andersen O, Salling N. Oxygen-linked hydrogen ion binding of human hemoglobin. Effects of carbon dioxide and 2, 3-diphosphoglycerate. II. Studies on whole blood. Scand J Clin Lab Invest 1971;27:361–6.

[33] Lide DR. CRC handbook of chemistry and physics. Boca Raton, FL: CRC Press; 2004.

[34] Atha DH, Ackers GK. Calorimetric determination of the heat of oxygenation of human hemoglobin as a function of pH and the extent of reaction. Biochemistry 1974;13:2376–2382.

[35] Hester RL, Brown AJ, Husband L, Iliescu R, Pruett D, Summers R, Coleman TG. HumMod: a modeling environment for the simulation of integrative human physiology. Front Physiol 2011;2.

[36] Kofranek J, Matousek S, Rusz J, Stodulka P, Privitzer P, Matejak M, Tribula M. The Atlas of Physiology and Pathophysiology: Web-based multimedia enabled interactive simulations. Comput Methods Programs Biomed 2011;104: 143–53.

[37] Jarvis SS, Levine BD, Prisk GK, Shykoff BE, Elliott AR, Rosow E, Blomqvist CG, Pawelczyk JA. Simultaneous determination of the accuracy and precision of closed-circuit cardiac output rebreathing techniques. J Appl Physiol 2007; 103:867–74.

[38] Sun X-G, Hansen JE, Ting H, Chuang M-L, Stringer WW, Adame D, Wasserman K. Comparison of exercise cardiac output by the Fick Principle using oxygen and carbon dioxide. CHEST J 2000;118:631–40.

[39] Rees SE, Andreassen S. Mathematical models of oxygen and carbon dioxide storage and transport: the acid-base chemistry of blood. Crit Rev Biomed Eng 2005;33.

[40] Imai K, Yonetani T. PH dependence of the Adair constants of human hemoglobin. Nonuniform contribution of successive oxygen bindings to the alkaline Bohr effect. J Biol Chem 1975;250:2227–31.

[41] Adair G, Cordero N, Shen T. The effect of temperature on the equilibrium of carbon dioxide and blood and the heat of ionization of haemoglobin. J Physiol 1929;67:288–98.

[42] Chipperfield J, Rossi-Bernardi L, Roughton F. Direct calorimetric studies on the heats of ionization of oxygenated and deoxygenated hemoglobin. J Biol Chem 1967; 242:777–83.

[43] Weber RE, Campbell KL. Temperature dependence of haemoglobin-oxygen affinity in heterothermic vertebrates: mechanisms and biological significance. Acta Physiolog 2011; 202:549–62.

[44] Weber RE, Fago A, Campbell KL. Enthalpic partitioning of the reduced temperature sensitivity of O2 binding in bovine hemoglobin. Comparat Biochem Physiol A: Molec Integrat Physiol 2014;176:20–5.

[45] Szabo A, Karplus M. A mathematical model for structure-function relations in hemoglobin. J Mol Biol 1972; 72:163–97.

[46] Brunori M, Coletta M, Di Cera E. A cooperative model for ligand binding to biological macromolecules as applied to oxygen carriers. Biophys Chem 1986;23:215–22.

[47] Henry ER, Bettati S, Hofrichter J, Eaton WA. A tertiary two-state allosteric model for hemoglobin. Biophys Chem 2002;98:149–64.

[48] de Bruin SH, Rollema HS, Janssen LH, van Os GA. The interaction of chloride ions with human hemoglobin. Biochem Biophys Res Commun 1974;58:210–5.

[49] Rollema HS, de Bruin SH, Janssen L, Van Os G. The effect of potassium chloride on the Bohr effect of human hemoglobin. J Biol Chem 1975;250:1333–9.

[50] Benesch R, Benesch RE, Yu CI. Reciprocal binding of oxygen and diphosphoglycerate by human hemoglobin. Proc Natl Acad Sci USA 1968;59:526.

[51] Bunn HF, Briehl RW. The interaction of 2, 3-diphosphoglycerate with various human hemoglobins. J Clin Invest 1970; 49:1088.

[52] Siggaard-Andersen O, Salling N, Nörgaard-Pedersen B, Rørth M. Oxygen-linked hydrogen ion binding of human hemoglobin. effects of carbon dioxide and 2, 3-diphosphoglycerate: III. Comparison of the Bohr effect and the Haldane effect. Scand J Clin Lab Invest 1972;29:185–93.

Bibliography

[1] T. Kulhánek and M. Šárek. Processing of medical images in virtual distributed environment. In *Proceedings of the 2009 Euro American Conference on Telematics and Information Systems New Opportunities to increase Digital Citizenship - EATIS '09*, pages 1–3. ACM, 2009. doi:10.1145/1551722.1551732. (cited on p. 1, 2, 4, 37, 51)

[2] T. Kulhánek, M. Frič, and M. Šárek. Remote Analysis of Human Voice – Lossless Sound Recording Redirection. *Analysis of Biomedical Signals and Images. Proceedings of 20th International EURASIP Conference (BIOSIGNAL)*, pages 394–397, 2010. URL: http://bs2010.biosignal.cz/papers/1092.pdf. (cited on p. 1, 2, 4, 37, 55)

[3] T. Kulhánek. Infrastructure for Data Storage and Computation in Biomedical Research. *European Journal for Biomedical Informatics (EJBI)*, 6(1):55–58, 2010. URL: http://www.ejbi.org/img/ejbi/ejbi2010-1.pdf. (cited on p. 1, 2, 4, 37, 61)

[4] T. Kulhánek, M. Mateják, J. Šilar, and J. Kofránek. Parameter Estimation of Complex Mathematical Models of Human Physiology Using Remote Simulation Distributed in Scientific Cloud. In *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, pages 712–715, 2014. doi:10.1109/BHI.2014.6864463. (cited on p. 1, 3, 4, 38, 67)

[5] T. Kulhánek, J. Kofránek, and M. Mateják. Modeling of Short-term Mechanism of Arterial Pressure Control in the Cardiovascular System: Object-oriented and Acausal Approach. *Computers in Biology and Medicine*, 54:137–144, September 2014. doi:10.1016/j.compbiomed.2014.08.025. (cited on p. 1, 3, 4, 30, 73)

[6] T. Kulhánek, M. Tribula, J. Kofránek, and M. Mateják. Simple Models of the Cardiovascular System for Educational and Research Purposes. *MEFANET Journal*, 2(2):56–63, 2014. (cited on p. 1, 3, 4, 30, 83)

[7] M. Mateják, T. Kulhánek, and S. Matoušek. Adair-based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity. *Scandinavian Journal of Clinical \& Laboratory Investigation*, 2014. doi:10.3109/00365513.2014.984320. (cited on p. 1, 3, 4, 38, 39, 41, 93)

[8] T. Kulhánek, J. Šilar, M. Mateják, P. Privitzer, J. Kofranek, and M. Tribula. Distributed Computation and Parameter Estimation in Identification of Physiological Systems. *VPH conference*, 2010. (cited on p. 1, 3)

[9] T. Kulhánek, F. Ježek, M. Mateják, J. Šilar, P. Privitzer, M. Tribula, and J. Kofránek. RESTful Web Service to Build Loosely Coupled Web Based Simulation of Human Physiology. *Transactions of Japanese Society for Medical and Biological Engineering*, 51(Supplement):R–32, 2013. doi:10.11239/jsmbe.51.R-32. (cited on p. 1, 3, 42)

[10] J. Kofránek, M. Mateják, P. Privitzer, M. Tribula, T. Kulhánek, J. Šilar, and R. Pecinovský. HumMod-Golem Edition: Large Scale Model of Integrative Physiology for Virtual Patient Simulators. In *Proceedings of World Congress in Computer Science 2013 (WORLDCOMP'13), International Conference on Modeling, Simulation and Visualisation Methods (MSV'13)*, pages 182–188, 2013. (cited on p. 1, 3, 30)

[11] M. Mateják, T. Kulhánek, J. Šilar, P. Privitzer, F. Ježek, and J. Kofránek. Physiolibrary - Modelica Library for Physiology. In *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*, pages 499–505. Linköping University Electronic Press, 2014. doi:10.3384/ecp14096499. (cited on p. 1, 3)

[12] T. Kulhánek and M. Šárek. Virtualizace a integrace v gridovém pacs systému (cze) virtualization and integration in grid pacs system (eng). In *Mefanet 2008*. MSD Brno, 2008. URL: http://www.mefanet.cz/res/file/mefanet2008/prispevky/25_kulhanek.pdf. (cited on p. 1, 2)

[13] M. Šárek and T. Kulhánek. Nové směry medicínských aplikací sdružení CESNET. In *sborník příspěvků MEDSOFT*, pages 145–148, 2009. (cited on p. 1, 2)

[14] T. Kulhánek. Virtual Distributed Environment for Exchange of Medical Images. In D. Kuželová, editor, *Proceeding of PhD. Conference 2009 was published as: Doktorandské dny '09*, pages 62–64. Institute of Computer Science/MatfyzPress, 2009. URL: http://www.cs.cas.cz/hakl/doktorandsky-den/files/2009/sbornik-dd-2009.pdf. (cited on p. 1, 2)

[15] T. Kulhánek, M. Frič, and M. Šárek. Vzdálený přístup k virtuálním výukovým a výzkumným aplikacím - podpora foniatrických vyšetření. In D. Schwarz, J. Majerník, L. Dušek, S. Štípek, and V. Mihál, editors, *Mefanet 2009*. MSD Brno, 2009. URL: http://www.mefanet.cz/res/file/mefanet2009/prispevky/kulhanek_full.pdf. (cited on p. 1)

[16] T. Kulhánek, M. Frič, and M. Šárek. Vzdálená analýza lidského hlasu - bezeztrátové nahrávání zvuku přes ip sítě. In *sborník příspěvků MEDSOFT*, pages 96–101, 2010. (cited on p. 1, 2)

[17] T. Kulhánek, J. Šilar, J. Kofránek, M. Mateják, P. Privitzer, and M. Tribula. Od výukového modelu k identifikaci fyziologického systému. In D. Schwarz, M. Komenda, J. Majerník, S. Štípek, V. Mihál, and L. Dušek, editors, *Mefanet 2010*. MSD Brno, 2010. URL: http://www.mefanet.cz/res/file/mefanet2010/prispevky/kulhanek-full.pdf. (cited on p. 1)

[18] T. Kulhánek. From Educational Models Towards Identification of Physiological Systems. *Mefanet Report*, 04:69–72, 2011. URL: http://www.mefanet.cz/res/file/reporty/mefanet-report-2011.pdf. (cited on p. 1, 3)

[19] T. Kulhánek. Infrastructure for Data Storage and Computation in Biomedical Research. In D. Kuželová, editor, *Proceeding of PhD. Conference 2011 was published as: Doktorandské dny '11*, pages 90–94. Institute of Computer Science/MatfyzPress, 2011. URL: http://www.cs.cas.cz/hakl/doktorandsky-den/files/2011/sbornik-dd-2011.pdf. (cited on p. 1, 2)

[20] T. Kulhánek, M. Frič, and J. Hrb. Vzdálená analýza lidského hlasu v reálném čase. In *sborník příspěvků MEDSOFT*, pages 180–184, 2012. URL: http://www.creativeconnections.cz/medsoft/2012/Medsoft_Kulhánek_Tomáš.pdf. (cited on p. 1, 2)

[21] T. Kulhánek, M. Mateják, J. Šilar, P. Privitzer, M. Tribula, F. Ježek, and J. Kofránek. Hybridní architektura pro webové simulátory (cze) Hybrid architecture for web simulators (en). In *sborník příspěvků MEDSOFT*, pages 115–121, 2013. URL: http://www.creativeconnections.cz/medsoft/2013/Medsoft_2013_Kulhanek.pdf. (cited on p. 1, 3, 42)

[22] T. Kulhánek, M. Mateják, J. Šilar, and J. Kofránek. Identifikace fyziologických systémů. In *sborník příspěvků MEDSOFT*, pages 148–153, 2014. URL: http://www.creativeconnections.cz/medsoft/2014/Medsoft_2014_Kulhanek.pdf. (cited on p. 1, 3)

[23] T. Kulhánek and M. Frič. Remote Analysis of Human Voice - Environment for Voice Training and ORL Medicine. In *EGI Technical Forum, Prague*, Prague, 2012. (cited on p. 1, 38)

[24] J. Kofránek, P. Privitzer, M. Tribula, J. Šilar, T. Kulhánek, M. Mateják, and F. Ježek. Virtual Patient Simulator, CZ Utility model. Application number: 2014-30329. Issue number: 27613, 2014. URL: http://isdv.upv.cz/portal/pls/portal/portlets.pts.det?xprim=10082946. (cited on p. 1, 3)

[25] J. Kofránek, R. Pecinovský, and P. Novák. Kopenograms – Graphical Language for Structured Algorithms. In *International Conference Foundations of Computer Science*, pages 90–96, 2012. (cited on p. 5)

[26] R. Bayer and E. M. McCreight. Organization and Maintenance of Large Ordered Indexes. *Acta Informatica*, 1(3):173–189, 1972. doi:10.1007/BF00288683. (cited on p. 6)

[27] R. Bayer. Symmetric Binary B-Trees: Data Structure and Maintenance Algorithms. *Acta Informatica*, 1(4):290–306, 1972. doi:10.1007/BF00289509. (cited on p. 6)

[28] A. Cobham. The Intrinsic Computational Difficulty of Functions. In *Proc. 1964 Congress for Logic,Methodology and Philosophy of Science.*, pages 24–30, 1964. (cited on p. 6)

[29] S. A. Cook. An Overview of Computational Complexity. *Communications of the ACM*, 26(6):400–408, June 1983. doi:10.1145/358141.358144. (cited on p. 6)

[30] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). *Computers and Intractability*, 1979. (cited on p. 6, 7)

[31] S. A. Cook. The Complexity of Theorem-Proving Procedures. In *STOC '71 Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971. doi:10.1145/800157.805047. (cited on p. 7)

[32] R. Karp and \. Reducibility among Combinatorial Problems", in R. *Miller and J. Thatcher, eds. Complexity of Computer Computations , Plenum Press*, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9. (cited on p. 7)

[33] R. E. Korf. Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence*, 27:97–109, 1985. (cited on p. 7)

[34] J. Pearl and R. E. Korf. Search Techniques. *Annual Review of Computer Science*, 2:451–467, 1987. doi:10.1146/annurev.cs.02.060187.002315. (cited on p. 7)

[35] Planck Collaboration, P. a. R. Ade, N. Aghanim, C. Armitage-Caplan, M. Arnaud, et al. Planck 2013 Results. I. Overview of Products and Scientific Results. *Physics*, 6:107, March 2013. arXiv:1303.5062, doi:10.1051/0004-6361/201321529. (cited on p. 7)

[36] C. Papadimitriou. *Computational Complexity*. Addison-WesleyPublishing Company, 1995. (cited on p. 7)

[37] M. Sipser. *Introduction to the Theory of Computation*. 2012. (cited on p. 7)

[38] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach, 3rd edition*. 2009. doi:10.1017/S0269888900007724. (cited on p. 8)

[39] B. F. J. Manly. *Randomization, bootstrap and Monte Carlo methods in biology*. Chapman and Hall/CRC, 2007. URL: http://www.loc.gov/catdir/toc/fy0702/2006047407.html, doi:10.2307/1271419. (cited on p. 8)

[40] G. M. Amdahl. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. In *Proceedings of AFIPS Conference*, pages 483–485, 1967. (cited on p. 9, 40)

[41] J. L. Gustafson. Reevaluating Amdahl's law. *Communications of the ACM*, 31(5):532–533, May 1988. doi:10.1145/42411.42415. (cited on p. 10)

[42] D. R. Butenhof. *Programming with POSIX Threads*. 1997. (cited on p. 12)

[43] B. Chapman, G. Jost, and R. van der Pas. *Using OpenMP*, volume 10. MIT press, 2008. (cited on p. 12)

[44] P. S. Pacheco. *Parallel Programming with MPI*. Morgan Kaufmann, 1997. (cited on p. 12)

[45] J. Diaz, C. Muñoz Caro, and A. Niño. A Survey of Parallel Programming Models and Tools in the Multi and Many-core Era. *IEEE Transactions on Parallel and Distributed Systems*, 23(8):1369–1386, 2012. doi:10.1109/TPDS.2011.308. (cited on p. 12)

[46] C. Moler. Matrix Computation on Distributed Memory Multiprocessors. *Hypercube multiprocessors*, pages 181–195, 1986. (cited on p. 12)

[47] I. Foster. *Designing and Building Parallel Programs*. Addison-Wesley Publishing Company, 1995. (cited on p. 12, 13)

[48] E. Cantu-Paz. *Designing Efficient and Accurate Parallel Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1999. (cited on p. 13)

[49] P. H. Madden. Dispelling the Myths of Parallel Computing. *IEEE Design \& Test of Computers*, (April):58–64, 2012. doi:10.1109/MDT.2012.2230391. (cited on p. 13)

[50] D. Culer, J. P. Singh, and A. Gupta. *Parallel Computer Architecture*. Morgan Kaufmann, 1998. (cited on p. 13)

[51] T. Rauber and G. Rünger. *Parallel Programming: For Multicore and Cluster Systems*. Springer, 2013. (cited on p. 13)

[52] A. S. Tanenbaum and M. van Steen. *Distributed Systems: Principles and Paradigms, 2/E*. 2007. (cited on p. 13)

[53] T. Erl. *SOA Principles of Service Design*. Prentice Hall, 2008. (cited on p. 14)

[54] R. T. Fielding. Chapter 5: Representational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures, Dissertation*, 2000. (cited on p. 15)

[55] S. Vinoski. REST Eye for the SOA Guy. *IEEE Internet Computing*, 11:82–84, 2007. doi:10.1109/MIC.2007.22. (cited on p. 15)

[56] M. Fowler. *Patterns of Enterprise Application Architecture*. Pearson Education, Inc., 2003. (cited on p. 15)

[57] J. Nilsson. *Applying Domain-Driven Design and Patterns*. Pearson Education, Inc., 2006. (cited on p. 15)

[58] G. Hohpe and B. Woolf. *Enterprise Integration Patterns*. Pearson Education, Inc., 2002. (cited on p. 15)

[59] V. Novák and J. Vojtěch. Deployment of a DWDM System with CLA Optical Amplifiers in the CESNET2 Network. *Networking Studies*, page 27, 2007. (cited on p. 16)

[60] I. Foster and C. Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004. (cited on p. 16)

[61] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International journal of high performance computing applications*, 15(3):200–222, 2001. (cited on p. 16)

[62] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, 23(3):187–200, 2000. (cited on p. 16)

[63] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Service Architecture for Distributed Systems Integration. *Grid Computing: Making the Global Infrastructure a Reality*, pages 217–249, 2003. (cited on p. 17)

[64] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A Security Architecture for Computational Grids. In *Proceedings of the 5th ACM Conference on Computer and Communications Security - CCS '98*, pages 83–92, New York, New York, USA, November 1998. ACM Press. doi:10.1145/288090.288111. (cited on p. 17)

[65] I. Bird, B. Jones, and K. Kee. The Organization and Management of Grid Infrastructures. *Computer*, 42(1):36–46, January 2009. doi:10.1109/MC.2009.28. (cited on p. 17)

[66] D. Adamova. Computing Framework for the LHC: Current Status and Challenges of the High Luminosity LHC Future. In *52 International Winter Meeting on Nuclear Physics, Bormio,Italy*, number January, 2014. URL: http://pos.sissa.it/archive/conferences/212/021/Bormio2014_021.pdf. (cited on p. 17)

[67] M. Litzkow and M. Livny. Experience with the Condor Distributed Batch System. *IEEE Workshop on Experimental Distributed Systems*, 1990. doi:10.1109/EDS.1990.138057. (cited on p. 18)

[68] I. Foster and C. Kesselman. Globus: a Metacomputing Infrastructure Toolkit, 1997. doi:10.1177/109434209701100205. (cited on p. 18)

[69] E. Laure, S. Fisher, A. Frohner, C. Grandi, P. Kunszt, et al. Programming the Grid with gLite. *Computational Methods in Science and Technology*, 12:33–45, 2006. (cited on p. 18)

[70] M. Riedel, E. Laure, L. Field, J. P. Navarro, J. Casey, et al. Interoperation of World-wide Production e-Science Infrastructures. In *Concurrency and Computation: Practice and Experience 21.8*, number March, pages 961–990. 2009. doi:10.1002/cpe. (cited on p. 18)

[71] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@home: An Experiment in Public-Resource Computing. *Communications of the ACM*, 45(11):56–61, November 2002. doi:10.1145/581571.581573. (cited on p. 18)

[72] D. Anderson. Boinc: A System for Public-resource Computing and Storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*. IEEE, 2004. doi:10.1109/GRID.2004.14. (cited on p. 18, 35)

[73] Z. Balaton, G. Gombas, P. Kacsuk, A. Kornafeld, J. Kovacs, et al. SZTAKI Desktop Grid: A Modular and Scalable Way of Building Large Computing Grids. In *Parallel and Distributed Processing Symposium*. IEEE, 2007. doi:10.1109/IPDPS.2007.370665. (cited on p. 18)

[74] P. Kacsuk, J. Kovacs, Z. Farkas, A. C. Marosi, G. Gombas, and Z. Balaton. SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System. *Journal of Grid Computing*, 7(4):439–461, September 2009. doi:10.1007/s10723-009-9139-y. (cited on p. 18)

[75] G. Fedak, F. Cappello, and U. P. Sud. XtremWeb : A Generic Global Computing System. In *Proceedings of First IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 582–587, 2001. doi:10.1109/CCGRID.2001.923246. (cited on p. 18)

[76] W. Herr, D. I. Kaltchev, F. Schmidt, and E. Mcintosh. Large Scale Beam-beam Simulations for the CERN LHC Using Distributed Computing. Technical Report June 2006, 2006. URL: http://cds.cern.ch/record/972345. (cited on p. 18)

[77] N. Hø imyr, J. Blomer, P. Buncic, M. Giovannozzi, A. Gonzalez, et al. BOINC Service for Volunteer Cloud Computing. *Journal of Physics: Conference Series*, 396(3):032057, December 2012. doi:10.1088/1742-6596/396/3/032057. (cited on p. 18)

[78] P. Kacsuk, Z. Farkas, and G. Fedak. Towards Making BOINC and EGEE Interoperable. *2008 IEEE Fourth International Conference on eScience*, pages 478–484, December 2008. doi:10.1109/eScience.2008.111. (cited on p. 19)

[79] E. Urbah, P. Kacsuk, Z. Farkas, G. Fedak, G. Kecskemeti, et al. EDGeS: Bridging EGEE to BOINC and XtremWeb. *Journal of Grid Computing*, 7(3):335–354, September 2009. doi:10.1007/s10723-009-9137-0. (cited on p. 19)

[80] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, et al. Xen and the Art of Virtualization. *ACM SIGOPS Operating Systems Review*, 37:164, 2003. doi:10.1145/1165389.945462. (cited on p. 19)

[81] L. Youseff, R. Wolski, B. Gorda, and C. Krintz. Paravirtualization for HPC Systems. *Frontiers of High Performance Computing and Networking â€“ ISPA 2006 Workshops*, pages 474–486, 2006. doi:10.1007/11942634_49. (cited on p. 19)

[82] M. Ruda, Z. Šustr, J. Sitera, D. Antoš, L. Hejtmánek, P. Holub, and M. Mulač. Virtual Clusters as a New Service of MetaCentrum, the Czech NGI. In *Cracow Grid Workshop'09*, pages 66–71, 2009. (cited on p. 19)

[83] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud Computing and Grid Computing 360-degree Compared. *Grid Computing Environments ...*, 2008. doi:10.1109/GCE.2008.4738445. (cited on p. 19, 21)

[84] P. Mell and T. Grance. The NIST Definition of Cloud Computing. 2011. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. (cited on p. 19)

[85] A. Anjum, R. Hill, R. McClatchey, N. Bessis, and A. Branson. Glueing Grids and Clouds Together: a Service-oriented Approach. *International Journal of Web and Grid Services*, 8(3):248–265, 2012. doi:10.1504/IJWGS.2012.049169. (cited on p. 20, 47)

[86] P. Kacsuk. P-GRADE Portal Family for Grid Infrastructures. *Concurrency Computation Practice and Experience*, 23:235–245, 2011. doi:10.1002/cpe.1654. (cited on p. 20, 21, 35)

[87] D. Milojicic, I. M. Llorente, and R. S. Montero. OpenNebula: A Cloud Management Tool. *IEEE Internet Computing*, 15(2):11–14, 2011. doi:10.1109/MIC.2011.44. (cited on p. 20)

[88] R. Kumar, N. Gupta, S. Charu, K. Jain, and S. Jangir. Open Source Solution for Cloud Computing Platform Using OpenStack. 3(5):89–98, 2014. URL: http://www.ijcsmc.com/docs/papers/May2014/V3I5201427.pdf. (cited on p. 20)

[89] S. Ortiz. The Problem with Cloud-computing Standardization. *Computer*, 44(July):13–16, 2011. doi:10.1109/MC.2011.220. (cited on p. 20)

[90] G. Hager and G. Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. 2010. doi:10.1201/EBK1439811924. (cited on p. 20)

[91] J. Levesque and G. Wagenbreth. *High Performance Computing: Programming and Applications*. CRC Press, 2010. (cited on p. 20)

[92] I. Raicu, I. T. Foster, and Y. Zhaoyo. Many-task computing for grids and supercomputers. In *2008 Workshop on Many-Task Computing on Grids and Supercomputers, MTAGS 2008*, 2008. doi:10.1109/MTAGS.2008.4777912. (cited on p. 20)

[93] I. Raicu. *Many-task Computing: Bridging the Gap between High-throughput Computing and High-performance Computing*. PhD thesis, 2009. (cited on p. 20)

[94] I. Raicu, I. Foster, M. Wilde, Z. Zhang, K. Iskra, et al. Middleware Support for Many-task Computing. *Cluster Computing*, 13:291–314, 2010. doi:10.1007/s10586-010-0132-9. (cited on p. 20)

[95] J. Pasley. How BPEL and SOA Are Changing Web Services Development. *IEEE Internet Computing*, 9(June):60–67, 2005. doi:10.1109/MIC.2005.56. (cited on p. 21)

[96] J. Yu and R. Buyya. A Taxonomy of Scientific Workflow Systems For Grid Computing. *ACM SIGMOD Record*, 34(3):44, 2005. doi:10.1145/1084805.1084814. (cited on p. 21)

[97] Y. Zhao, I. Raicu, and I. Foster. Scientific Workflow Systems for 21st Century e-Science, New Bottle or New Wine? *IEEE Congress on Services - Part I, 2008.*, pages 467 – 471, 2008. arXiv:0808.3545. (cited on p. 21)

[98] V. Curcin and M. Ghanem. Scientific Workflow Systems - Can One Size Fit All? *2008 Cairo International Biomedical Engineering Conference*, 2008. doi:10.1109/CIBEC.2008.4786077. (cited on p. 21)

[99] P. Kacsuk, T. Kiss, and G. Sipos. Solving the Grid Interoperability Problem By P-GRADE Portal at Workflow Level. *Future Generation Computer Systems*, 24:744–751, 2008. doi:10.1016/j.future.2008.02.008. (cited on p. 21)

[100] N. Wilkins-Diehr. Special Issue: Science Gateways—Common Community Interfaces to Grid Resources. *Concurrency and Computation: Practice and Experience*, 19(6):743–749, April 2007. URL: http://doi.wiley.com/10.1002/cpe.1098, doi:10.1002/cpe.1098. (cited on p. 21)

[101] M. Pierce, S. Marru, B. Demeler, R. Singh, and G. Gorbet. The Apache Airavata Application Programming Interface: Overview and Evaluation with the UltraScan Science Gateway. In *2014 9th Gateway Computing Environments Workshop*, pages 25–29. IEEE, November 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/

wrapper.htm?arnumber=7021845, doi:10.1109/GCE.2014.15. (cited on p. 21)

[102] S. Memon, M. Riedel, F. Janetzko, B. Demeler, G. Gorbet, et al. Advancements of the UltraScan Scientific Gateway for Open Standards-based Cyberinfrastructures. *Concurrency and Computation: Practice and Experience*, 26(13):2280–2291, September 2014. doi:10.1002/cpe.3251. (cited on p. 21)

[103] P. Kacsuk, Z. Farkas, M. Kozlovszky, G. Hermann, A. Balasko, K. Karoczkai, and I. Marton. WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities. *Journal of Grid Computing*, 10:601–630, 2012. doi:10.1007/s10723-012-9240-5. (cited on p. 21)

[104] I. N. Bankman, editor. *Handbook of Medical Imaging; Processing and Analysis*. 2000. (cited on p. 24)

[105] K. Slavicek, O. Dostal, M. Javornik, and M. Drdla. MEDIMED - Regional Centre for Medicine Image Data Processing. *2010 Third International Conference on Knowledge Discovery and Data Mining*, pages 310–313, January 2010. doi:10.1109/WKDD.2010.133. (cited on p. 25, 27, 43)

[106] P. Ross, R. Sepper, and H. Pohjonen. Cross-border Teleradiology – Experience from Two International Teleradiology Projects. *European journal of radiology*, 73(1):20–5, January 2010. doi:10.1016/j.ejrad.2009.10.016. (cited on p. 25)

[107] V. Saliba, H. Legido-Quigley, R. Hallik, A. Aaviksoo, J. Car, and M. McKee. Telemedicine Across Borders: A Systematic Review of Factors That Hinder or Support Implementation. *International journal of medical informatics*, 81(12):793–809, December 2012. doi:10.1016/j.ijmedinf.2012.08.003. (cited on p. 25)

[108] S. G. Erberich, M. Bhandekar, M. D. Nelson, A. Chervenak, and C. Kesselman. DICOM Grid Interface Service for Clinical and Research PACS: A Globus Toolkit Web Service for Medical Data Grids. *International Journal of Computer Assisted Radiology and Surgery*, 1(ii):100–102, 2006. doi:10.1007/s11548-006-0013-0. (cited on p. 25, 26)

[109] S. G. Erberich, J. C. Silverstein, A. Chervenak, R. Schuler, M. D. Nelson, and C. Kesselman. Globus MEDICUS – Federation of DICOM Medical Imaging Devices into Healthcare Grids. *Studies in health technology and informatics*, 126:269–278, 2007. (cited on p. 25, 26, 43)

[110] H. Duque, J. Montagnat, J. Pierson, L. Brunie, and I. E. Magnin. DM 2 : A Distributed Medical Data Manager for Grids. In *Biogrid'03 workshop*, pages 138–147, Tokyo, Japan. (cited on p. 25)

[111] J. Montagnat, A. Frohner, D. Jouvenot, C. Pera, P. Kunszt, et al. A Secure Grid Medical Data Manager Interfaced to the gLite Middleware. *Journal of Grid Computing*, 6(1):45–59, July 2007. doi:10.1007/s10723-007-9088-2. (cited on p. 25)

[112] D. Krefting, J. Bart, K. Beronov, O. Dzhimova, J. Falkner, et al. MediGRID: Towards a User Friendly Secured Grid Infrastructure. *Future Generation Computer Systems*, 25(3):326–336, 2009. doi:10.1016/j.future.2008.05.005. (cited on p. 25)

[113] D. Krefting, M. Vossberg, A. Hoheisel, and T. Tolxdorff. Simplified Implementation of Medical Image Processing Algorithms into a Grid Using a Workflow Management System. *Future Generation Computer Systems*, 26(4):681–684, 2010. doi:10.1016/j.future.2009.07.004. (cited on p. 25)

[114] K. Skaburskas, F. Estrella, J. Shade, D. Manset, J. Revillard, et al. Health-e-Child: A Grid Platform for European Paediatrics. *Journal of Physics: Conference Series*, 119:082011, 2008. doi:10.1088/1742-6596/119/8/082011. (cited on p. 25)

[115] S. Benkner, A. Arbona, G. Berti, A. Chiarini, R. Dunlop, et al. @neurIST: Infrastructure for Advanced Disease Management through Integration of Heterogeneous Data, Computing, and Complex Processing Services. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1365–1377, 2010. doi:10.1109/TITB.2010.2049268. (cited on p. 25)

[116] M. Kuba, O. Krajícek, P. Lesný, J. Vejvalka, and T. Holecek. Grid Empowered Sharing of Medical Expertise. *Studies in health technology and informatics*, 120:273–282, 2006. (cited on p. 25)

[117] M. Kuba, D. Kouřil, and M. Procházka. Sealed Grid with Downloadable Services. In T. Priol and M. Vanneschi, editors, *Towards Next Generation Grids*, pages 83–91. Springer US, 2007. doi:10.1007/978-0-387-72498-0_8. (cited on p. 26)

[118] I. Foster. Globus Toolkit Version 4: Software for Service-oriented Systems. *Journal of Computer Science and Technology*, 21(4):513–520, 2006. (cited on p. 26)

[119] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster. The Globus Striped GridFTP Framework and Server. *Proceedings of the ACM/IEEE 2005 Supercomputing Conference, SC'05*, 2005(c):1–11, 2005. doi:10.1109/SC.2005.72. (cited on p. 26)

[120] J. Cooley and J. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of computation*, pages 297–301, 1965. URL: http://www.jstor.org/stable/2003354. (cited on p. 27)

[121] M. Frigo and S. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. doi:10.1109/JPROC.2004.840301. (cited on p. 27)

[122] W. DeLeo LeBorgne and B. D. Weinrich. Phonetogram Changes for Trained Singers over a Nine-month Period of Vocal Training. *Journal of Voice*, 16(1):37–43, 2002. doi:10.1016/S0892-1997(02)00070-X. (cited on p. 27)

[123] F. L. Wuyts, L. Heylen, F. Mertens, M. D. Caju, R. Rooman, P. H. de Heyning, and M. De Bodt. Effects of Age, Sex, and Disorder on Voice Range Profile Characteristics of 230 Children. *ANNALS OF OTOLOGY RHINOLOGY AND LARYNGOLOGY*, 112(6):540–548, 2003. (cited on p. 27)

[124] J. G. Švec and H. K. Schutte. Videokymography: High-speed Line Scanning of Vocal Fold Vibration. *Journal of Voice*, 10(2):201–205, 1996. doi:10.1016/S0892-1997(96)80047-6. (cited on p. 27)

[125] J. G. Svec, F. Sram, and H. K. Schutte. Videokymography in Voice Disorders: What to Look for? *The Annals of otology, rhinology, and laryngology*, 116(3):172–80, March 2007. URL: http://europepmc.org/abstract/MED/17419520. (cited on p. 27)

[126] M. Frič. Parametrizovaný fonetogram obecných řečových a hlasových projevů - ParVRP, 2007. URL: http://zvuk.hamu.cz/vyzkum/dokumenty/TL12x.pdf. (cited on p. 27, 28, 43)

[127] M. Frič, T. Kulhánek, and J. Hrb. Systém pro vzdálené přístupnou analýzu hlasu RealVoiceLab, 2012. URL: http://zvuk.hamu.cz/vyzkum/dokumenty/TL46x.pdf. (cited on p. 27, 28)

[128] P. Kohl, E. J. Crampin, T. A. Quinn, and D. Noble. Systems Biology: An Approach. *Clinical pharmacology and therapeutics*, 88:25–33, 2010. doi:10.1038/clpt.2010.92. (cited on p. 29, 46)

[129] P. J. Hunter and M. Viceconti. The VPH-Physiome Project: Standards and Tools for Multiscale Modeling in Clinical Applications. *IEEE Reviews in Biomedical Engineering*, 2:40–53, 2009. doi:10.1109/RBME.2009.2036204. (cited on p. 29)

[130] A. C. Guyton, T. G. Coleman, and H. J. Granger. Circulation: Overall Regulation. *Annual review of physiology*, 34:13–46, 1972. doi:10.1146/annurev.ph.34.030172.000305. (cited on p. 29)

[131] R. L. Hester, R. Iliescu, R. Summers, and T. G. Coleman. Systems Biology and Integrative Physiological Modelling. *The Journal of physiology*, 589(Pt 5):1053–60, March 2011. doi:10.1113/jphysiol.2010.201558. (cited on p. 29)

[132] R. L. Hester, A. J. Brown, L. Husband, R. Iliescu, D. Pruett, R. Summers, and T. G. Coleman. HumMod: A Modeling Environment for the Simulation of Integrative Human Physiology. *Frontiers in physiology*, 2(April):12, January 2011. doi:10.3389/fphys.2011.00012. (cited on p. 29)

[133] E. Butterworth, B. E. Jardine, G. M. Raymond, M. L. Neal, and J. B. Bassingthwaighte. JSim, an Open-source Modeling System for Data Analysis. *F1000Research*, 2, May 2014. doi:10.12688/f1000research.2-288.v2. (cited on p. 30)

[134] P. J. Hunter. The IUPS Physiome Project: A Framework for Computational Physiology. *Progress in biophysics and molecular biology*, 85(2-3):551–69, January 2004. doi:10.1016/j.pbiomolbio.2004.02.006. (cited on p. 30)

[135] T. Yu, C. M. Lloyd, D. P. Nickerson, M. T. Cooling, A. K. Miller, et al. The Physiome Model Repository 2. *Bioinformatics (Oxford, England)*, 27(5):743–744, 2011. doi:10.1093/bioinformatics/btq723. (cited on p. 30)

[136] M. Hucka, A. Finney, B. J. Bornstein, S. M. Keating, B. E. Shapiro, et al. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. *Systems biology*, 1(1):41–53, 2004. (cited on p. 30)

[137] N. Le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, et al. BioModels Database: A Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems. *Nucleic acids research*, 34:D689–D691, 2006. doi:10.1093/nar/gkj092. (cited on p. 30)

[138] J. Kofránek and J. Rusz. Restoration of Guyton´s Diagram for Regulation of the Circulation as a Basis For Quantitative Physiological Model Development. *Physiological research / Academia Scientiarum Bohemoslovaca*, 59(6):897–908, January 2010. (cited on p. 30)

[139] J. Kofránek, M. Mateják, and P. Privitzer. Hummod – Large Scale Physiological Models in Modelica. In *Proceedings 8th Modelica Conference, Dresden, Germany*, pages 713–724, 2011. (cited on p. 30, 38, 39, 41)

[140] J. Fernandez de Canete, P. del Saz-Orozco, D. Moreno-Boza, and E. Duran-Venegas. Object-oriented Modeling and Simulation of the Closed Loop Cardiovascular System by Using SIMSCAPE. *Computers in biology and medicine*, 43(4):323–33, May 2013. doi:10.1016/j.compbiomed.2013.01.007. (cited on p. 30)

[141] J. Fernandez de Canete, J. Luque, J. Barbancho, and V. Munoz. Modelling of Long-term and Short-term Mechanisms of Arterial Pressure Control in the Cardiovascular System: An Object-oriented Approach. *Computers in biology and medicine*, 47:104–112, April 2014. doi:10.1016/j.compbiomed.2014.01.006. (cited on p. 30)

[142] P. Fritzson and P. Bunus. Modelica – A General Object-oriented Language for Continuous and Discrete-event System Modeling and Simulation. In *Simulation Symposium, 2002. Proceedings. 35th Annual*, pages 365–380. IEEE, 2002. (cited on p. 30)

[143] M. Tiller. Modelica by Example, 2014. URL: http://book.xogeny.com. (cited on p. 30)

[144] M. Hofmann. On the Complexity of Parameter Calibration in Simulation Models. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 2(4):217–226, October 2005. doi:10.1177/154851290500200405. (cited on p. 31)

[145] P. Eykhoff. *Trends and Progress in System Identification*. IFAC series for graduates, research workers and practising engineers;vol. 1. Pergamon, Oxford, 1981. (cited on p. 31)

[146] M. C. K. Khoo. *Physiological Control Systems*. IEEE press, 2000. (cited on p. 31, 34)

[147] C. G. Moles, P. Mendes, and J. R. Banga. Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods. *Genome Research*, pages 2467–2474, 2003. doi:10.1101/gr.1262503. (cited on p. 31)

[148] W. A. Pruett, L. D. Husband, G. Husband, M. Dakhlalla, K. Bellamy, T. G. Coleman, and R. L. Hester. A Population Model of Integrative Cardiovascular Physiology. *PloS one*, 8(9):e74329, 2013. (cited on p. 31)

[149] S. Takahashi, K. Sakawa, Y. Shiraishi, H. Miyashita, and S. Technology. Modeling, Simulation and Parameter Estimation of the Cardiovascular System by Using Model Based Approach. pages 493–500, 2013. (cited on p. 31)

[150] M. a. Abbass, E. ElSamahy, and A. Genedy. An Optimized Cardiovascular Model: A Pattern Search Approach. *Proceedings - 2012 International Conference on Biomedical Engineering and Biotechnology, iCBEB 2012*, 1:45–48, 2012. doi:10.1109/iCBEB.2012.62. (cited on p. 31)

[151] S. Maffioletti, R. Murri, B. Jonen, and S. Scheuring. Computational Workflows with GC3Pie. In *Proceedings of the EGI Community Forum*, 2012. (cited on p. 31)

[152] M. Humphrey, N. Beekwilder, J. L. Goodall, and M. B. Ercan. Calibration of Watershed Models Using Cloud Computing. *2012 IEEE 8th International Conference on E-Science*, (1048222):1–8, October 2012. doi:10.1109/eScience.2012.6404420. (cited on p. 31)

[153] J. H. Holland. *Adaptation in Natural and Artificial Systems*, volume Ann Arbor. 1975. URL: http://www.citeulike.org/group/664/article/400721. (cited on p. 32)

[154] T. Blochwitz, M. Otter, M. Arnold, and C. Bausch. The Functional Mockup Interface for Tool Independent Exchange of Simulation Models. In *Proceeding of 8th Modelica Conference, Dresden*, 2011. (cited on p. 33, 43)

[155] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie. Survey of Sampling-based Methods for Uncertainty and Sensitivity Analysis. *Reliability Engineering and System Safety*, 91:1175–1209, 2006. doi:10.1016/j.ress.2005.11.017. (cited on p. 34)

[156] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity Analysis in Practice*. 2004. doi:10.1002/0470870958. (cited on p. 34)

[157] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis. The Primer*. 2008. doi:10.1002/9780470725184. (cited on p. 34)

[158] W. van Meurs. *Modeling and Simulation in Biomedical Engineering: Applications in Cardiorespiratory Physiology*. McGraw-Hill Professional, 2011. (cited on p. 38, 39, 41)

[159] A. L. Chervenak, T. G. M. Van Erp, C. Kesselman, M. D'Arcy, J. Sobell, et al. A System Architecture for Sharing De-identified, Research-ready Brain Scans and Health Information Across Clinical Imaging Centers. *Studies in Health Technology and Informatics*, 175:19–28, 2012. doi:10.3233/978-1-61499-054-3-19. (cited on p. 43)

[160] S. Gotshall and B. Rylander. Optimal population size and the genetic algorithm. *Proceedings On Genetic And Evolutionary Computation Conference*, 2000. (cited on p. 44)

[161] K. Tø ndel, U. G. Indahl, A. B. Gjuvsland, J. O. Vik, P. Hunter, S. W. Omholt, and H. Martens. Hierarchical Cluster-based Partial Least Squares Regression (HC-PLSR) is an Efficient Tool for Metamodelling of Nonlinear Dynamic Models. *BMC systems biology*, 5(1):90, 2011. URL: http://www.biomedcentral.com/1752-0509/5/90, doi:10.1186/1752-0509-5-90. (cited on p. 44)

[162] P. Hunter, T. Chapman, P. V. Coveney, B. de Bono, V. Diaz, et al. A Vision and Strategy for the Virtual Physiological Human: 2012 Update. *Interface Focus*, 3(2), February 2013. URL: http://rsfs.royalsocietypublishing.org/content/3/2/20130004.abstract. (cited on p. 46)

[163] L. Itu, P. Sharma, K. Ralovich, V. Mihalef, R. Ionasec, et al. Non-invasive Hemodynamic Assessment of Aortic Coarctation: Validation with in Vivo Measurements. *Annals of Biomedical Engineering*, 41(4):669–681, 2013. doi:10.1007/s10439-012-0715-0. (cited on p. 46)

[164] T. H. Vines, A. Y. K. Albert, R. L. Andrew, F. Débarre, D. G. Bock, et al. The Availability of Research Data Declines Rapidly with Article Age. *Current Biology*, 24:94–97, 2014. arXiv:1312.5670, doi:10.1016/j.cub.2013.11.014. (cited on p. 46)

[165] P. Bryan Heidorn. Shedding Light on the Dark Data in the Long Tail of Science. *Library Trends*, 57(2):280–299, 2008. doi:10.1353/lib.0.0036. (cited on p. 46)

[166] C. Anderson. The Long Tail: How Endless Choice Is Creating Unlimited Demand. *Word Journal Of The International Linguistic Association*, page 256, 2006. (cited on p. 46, 47)

[167] C. Weinhardt, A. Anandasivam, B. Blau, N. Borissov, T. Meinl, W. Michalk, and J. Stöß er. Cloud Computing – A Classification, Business Models, and Research Directions. *Business \& Information Systems Engineering*, 1:391–399, 2009. doi:10.1007/s12599-009-0071-2. (cited on p. 47)

[168] S. Crouch, N. C. Hong, S. Hettrick, M. Jackson, A. Pawlik, et al. The Software Sustainability Institute: Changing Research Software Attitudes and Practices. *Computing in Science and Engineering*, 15(6):74–80, 2013. doi:10.1109/MCSE.2013.133. (cited on p. 47)