Competitive Programming: a Case Study for Developing a Simulation-based Decision Support System

Norbert Bátfai, *Member, IEEE,* Péter Jeszenszky, *Member, IEEE,* András Mamenyák, Béla Halász, Renátó Besenczi, János Komzsik, Balázs Kóti, Gergely Kövér, Máté Smajda, Csaba Székelyhídi, Tamás Takács, Géza Róka and Márton Ispány, *Member, IEEE*

***Abstract*—FootballAvatar is an experimental industrial research and development subproject of the project *'SziMe3D–3D technological innovation in tourism, education and sport'*. FootballAvatar aims to produce a novel decision support information system based on simulations for professional football clubs. This paper establishes the notion of football avatar in the sense of information technology, though it has a strong mathematical background. However, we would like to apply it in several analytic and simulation software tools developed in our project. The main question is that how this notion could be implemented and used in several software environments including C++, Java, and R, or from an architectural viewpoint, on desktops, smart phones, and tablets, while the kinds of uses and the base definitions have often changed during the R&D phases. This changing of the precise interpretation of the notion of "football avatar" has a direct impact on selecting the software process model. For this reason, we have developed an own software methodology called Competitive Programming (CP), which will be presented in detail, as the main result of the present paper. Our main goal with CP was to create a methodology that allows us to work effectively even when the objectives to achieve are changing rapidly. As an example of the application of the methodology, the paper discusses the aforementioned FootballAvatar project.**

***Index Terms*—Competitive Programming, Agile Programming, Software Process Improvement, OSS Policy, Football Avatars**

I. INTRODUCTION

FootballAvatar is an experimental research and development project aimed at producing the next generation of soccer analysis programs. A major innovation of this project is the simulation-based decision-making, where simulations are organized around the notion of "football avatar". Basically, it is a mathematical concept introduced in Section III-I. However, we would like to apply it in various software environments including C++, Java, and R, or, from an architectural viewpoint, on various front-end platforms, i.e., desktops, smart phones, and tablets. One major challenge was that the precise definition of "football avatar" had been changed during the R&D phases. Since refined definitions had to be used in many software components, we had to develop our own software process methodology practice called Competitive Programming (or CP for short). CP is a competition-based methodology that extends the agile development processes and is based on a combination of eXtreme Programming and Rapid Application Development (RAD). At the heart of the methodology are the creation of an initial rapid prototype and the formation of small (typically, one or two member) developer teams that work on forks of the initial prototype in competition. CP incorporates gamification elements to motivate the competition among teams. The use of free and open source software is also an important element of CP, thus it provides support to implement an open source software policy.

The paper presents the FootbalAvatar project to demonstrate an application scenario of Competitive Programming. CP will be introduced in the first part of the paper, then, in the second part the FootballAvatar system is presented in detail.

A. A Brief History of the FootballAvatar Project and Earlier Work

The idea of the FootballAvatar project was born in July 2009. Then, with the help of the Silicon Field Regional IT Cluster we had found investors and won a tender for the project. Some related works (e.g., [1], [2], and [3]) were created before the FootballAvatar project started. We do not use any of the data or software components created from these works, the FootballAvatar software system has been made from scratch.

The article [1] introduced the notion of "football avatar" using an XML-based approach. [2] and [3] investigated the RoboCup soccer simulation. Since it is an existing and well-known soccer simulation model, it was necessary to examine it before the developement process was started.

B. Technological and Methodological Background

In the scientific literature, it is not uncommon that a company customizes a well-known methodology. For example, [4] presents a study of Toyota's software development process called Toyota Production System. In general, numerous scientific publications can be found about the relationship between Software Process Improvement (SPI) and agile methodologies.

N. Bátfai is with the Faculty of Informatics, University of Debrecen, P.O. Box 12, 4010 Debrecen, Hungary, and also with SziMe3D Ltd., Debrecen, Hungary (e-mail: batfai.norbert@inf.unideb.hu).

P. Jeszenszky, M. Ispány, A. Mamenyák, R. Besenczi, J. Komzsik, B. Kóti, G. Kövér, M. Smajda, Cs. Székelyhídi and T. Takács are with the Faculty of Informatics, University of Debrecen, Hungary, and also with SziMe3D Ltd., Debrecen, Hungary.

B. Halász is with SziMe3D Ltd., Debrecen, Hungary.

G. Róka is with DVSC Futball Szervező Zrt., Debrecen, Hungary.

Manuscript received May 12, 2015. Revised: December 8, 2015.

For example, [5] introduces a mobile development oriented SPI customization.

We are committed to the Agile Manifesto [6] and we are familiar with agile software development methodologies, such as Scrum ([7], [8]) or eXtreme Programming [9]. There are many examples of customized agile methods, for example, see [10]. As another customization example, Solo Scrum [11] may be interesting for us because our SPI uses one-member development teams in many competitions (competition plays an essential role in our SPI). Our methodology also incorporates gamification [12] elements.

In recent years it has become more and more common to write programs that run parallel on the GPU, thus outperforming equivalent CPU solutions ([13], [14]). This would not have been possible without the appearance of easy-to-use APIs, such as the NVIDIA CUDA toolkit [15]. CUDA is widely used for research and simulations, such as simulating artificial neural networks [16]. We believe in this new approach to programming, and to make use of the computational power of GPUs, we also implemented our simulations using CUDA.

C. A Review of Existing Soccer Simulation Models

The whole FootballAvatar system is organized around soccer simulations. Thus, soccer simulations with their theoretical and technical background are the most characteristic feature of the system to be developed.

In our approach, a soccer simulation is (1) realistic, if it has the appearance of a real soccer match, that is, for example, there are players, who have inertia and acceleration; (2) quasi-realistic, if it can be considered to be similar to a real soccer match in some way; (3) non-realistic, where the aim of the simulation is not to reproduce the course of the game itself. For example, a match between two sophisticated 2D RoboCup Soccer Simulation (RCSS) teams, such as HELIOS [17] and WrightEagle [18], is a realistic simulation. On the other hand, FerSML [1] simulations are quasi-realistic models, because FerSML does not use any realistic kinematic models for the motions of the players. In the following, quasi-realistic simulations are also referred to as FerSML-like simulations. The Quantum Consciousness Soccer Simulation (QCSS) [19] is another example for the quasi-realistic model, but it typically can work as a non-realistic model too. Finally, soccer betting prediction is typically based on non-realistic models, for example, statistical forecasting models. Some of these models focus on predicting tournament outcomes [20] or league positions [21] while other ones are concerned with predicting outcomes of individual matches [22].

In the case of realistic and quasi-realistic simulations, we usually use the "TV criterion" to characterize the appearance of simulated matches. It is a subjective criterion introduced in [23] that checks whether the flow of play looks like a real soccer match. It should be noted that the TV criterion is entirely based on subjective opinions of human observers and therefore it cannot be quantitatively described.

At the beginning of the research process we have investigated and understood [2] the 2D RCSS model. However, the possibility of using of the 2D RCSS in FootballAvatar was rejected which was the conclusion of our previously cited work. One of the reasons of the rejection was that the project management has chosen to apply a closed-source license. The other reason was that RCSS [24] very strongly focuses on Artificial Intelligence, which is not surprising since robot soccer is a standard AI task. In this paper, we are interested only in sport science simulations.

Contrarily, FerSML is already a sport science model but the usage of this simulation model was abandoned also from the same license cause. For the same reason, we cannot use QCSS too. However, in the case of QCSS it is essential to emphasize that soccer simulation is not a determining factor, because QCSS is a cognitive model for trying to investigate the emerging human consciousness.

Finally, it should be noticed that our simulators cannot be compared with popular products of the game industry because FootballAvatar as a product is intended for experts of professional football clubs rather than lay audiences.

II. SOFTWARE PROCESS IMPROVEMENT IN FOOTBALLAVATAR

A. Project Organization

SziMe3D Ltd. is located in Debrecen, Hungary in Central Europe. It is the project company of FootballAvatar. "Nagyerdei Gerundium" is a SziMe3D working group which mainly consists of contracted researchers from the University of Debrecen, including 4 PhD doctors, 2 postgraduate researchers, 1 PhD student, and 8 BSc students. The initial idea and the essential part of the design were developed by this group. In addition, the modeling, analyzing, and simulation parts of FootballAvatar are also developed by the "Nagyerdei Gerundium".

There is a representative of SziMe3D Ltd. in the working group "Nagyerdei Gerundium" who corresponds with the Product Owner in Scrum terminology, see [7] and [8]. We are familiar with Scrum, but do not follow it, for example, "Nagyerdei Gerundium" is not a Scrum team, it is divided into smaller subgroups that overlap each other during the development. Our Software Process Improvement (SPI) will be introduced in Sect. II-B.

B. Competitive Programming

In research projects it is natural that clearly defined development targets cannot be established until we have enough experience in the area to be researched. In our case, this area has a strong mathematical flavour, simply because football avatar is a mathematical statistics-based concept (see Section III-I).

It is clear that choosing the software development methodology is an essential element for industrial projects. Taking into consideration that our project is a research project as well, where the software requirements are not clearly understood, it therefore seems appropriate to choose an agile methodology for the development of the FootballAvatar project.

In addition, it is important to note that the quintessence of the development of FootballAvatar is that it takes place in a university environment. This gives a unique opportunity for introducing some innovations in the software development

process. Since FootballAvatar is an industrial project one of the most fundamental issues is the interest of investors. However, the project is embedded in the University of Debrecen, since members of the core developer team (including all but one of the authors) who are contracted employees or apprentices of the project company are also with the university as a researcher or a student. In this environment we deeply believe in the Agile Manifesto [6], and what is more, the management of the project company also support it. The choice of this environment brings several benefits to the development. The main strength of our project is that we have the possibility to involve the best students in the software development. It is a very important aspect from the point of view of cost-efficiency, and it enables us to introduce a competition-based method for extending the agile development processes.

Fig. 1 shows the general model of our own software development process based on a combination of eXtreme Programming (XP) ([9], [25]) and Rapid Application Development (RAD) [26], where the competition among the forked rapid prototypes of XP programming pairs is focused. The first step of our approach is the creation of an initial rapid prototype to identify the major features of the application to be developed. This proto is based on user's and developer's stories presented in weekly project meetings and is typically created by a guru programmer (see Subsection II-B3). It is important to emphasize the role of the developers because they have no soccer-specific preconceptions. Even the absence of these preconceptions can allow us to create entirely new software products and services. Naturally, this has to be done in close cooperation with soccer and business experts. This iteration is a usual agile iteration which is shown by a dotted line in Fig. 1. Our competition-based agile software development practice presented in this paper is referred to as *Competitive Programming*.[1] The formal documents of this method can be found at http://footballavatar.hu/CP. Two of the most significant developer documents are the competition form and the OSS policy form that are shown in Table I and II. We maintain these forms in DocBook XML as part of our documentation process that is presented in more detail in Sect. III-J.

Table I shows the layout of our competition form. The lower part of the form supports the evaluation process that can be iterated many times depending on the result of the previous evaluation.

Table II shows the layout of our OSS submission form. The detailed description of our OSS policy process is presented in Sect. II-C.

Below we briefly survey the main competing areas, such as MABSA and FANM. The former acronym stands for MultiAgent-Based Server Architecture, the latter stands for FANM is Not MABSA. These will be detailed in sections III-E and III-F.

[1]It should be noted that the term *competitive programming* is also used broadly in the context of programming contests, where it is used to describe competitions in which participants compete with each other in solving various programming tasks [27]. There is also a Wikipedia article with the title *Competitive programming* [28] devoted to the topic. In our terminology, this term is used to denote a software development methodology.

TABLE I: Competition form

TO BE FILLED AT TEAM FORMATION	
Date of team formation	
Team name	
Member #1	
Member #2	
Supervisor of the team	
Short description of the task	
For which part of the system is the task related to	
Deadline of first iteration	
Repository location	
Names of competing teams	
Comment	
TO BE FILLED AT EVALUATION	
Date	
OSS policy verdict	approve/approve with limitations (see comment)/cancel
Meeting verdict	approve/cancel/suspend
Comment	

TABLE II: Open source software submission form

TO BE FILLED AT SUBMISSION	
Name of submitter	
Team of submitter	
Date of submission	
Name of software	
URL to obtain the software	
Type of intended usage	use as library/internal developer tool/runtime environment/content
For which parts of the system is the software intended to be used?	
Will the software be distributed with the final product?	yes/no
Name of license(s)	
Location where the license is indicated (e.g., LICENSE file included in package, web page)	
TO BE FILLED AT EVALUATION	
Person responsible for the verdict	
Verdict (approve or reject intended use)	approve/approve with limitations (see comment)/reject
Comment	

- Competing MABSA Implementations: MABSA is our internal research simulation platform which will be discussed in detail in Sect. III-E. For the development of this platform three development teams were allocated, namely: FBA One C++ (FBA is an acronym for

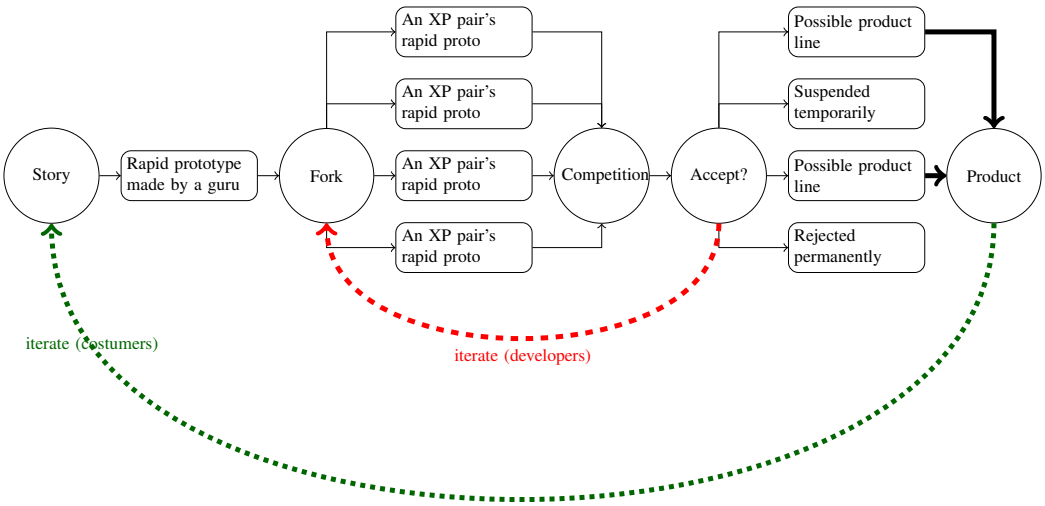


Fig. 1: A general model of the competition based agile software development process. The redundant paths (marked by bold line) are integrated into the product line.

"FootballAvatar"), Tunneled Footballers, and Hungarian Phoenix FC (see Fig. 2). At first, the development team Hungarian Phoenix (MABSA-HPFC) was suspended, and later it was cancelled. For a possible reason for this, see Sect. III-F1.

- Competing FANM Implementations: In our terminology, the term FANM stands for soccer simulation implementations that are easy to reuse across multiple applications and platforms. It will be introduced in Sect. III-F. The competing FANM development teams are shown in Fig. 3.
- Competing CUDA ports of FANM Implementations: A part of the FootballAvatar simulations can be computed on a Linux PC equipped with NVIDIA GPU. Porting FANM models to CUDA is a very challenging task but the result can be very effective. The object of the contest is to maximize the number of parallel threads of soccer matches in a CUDA block.

Finally, we note that similar competition-based methodological approaches are used in our other projects. These approaches have grown out of the first author's competition based teaching techniques. However, the idea of competing rapid prototypes is unequivocally rooted in the FootballAvatar project, because the success of the developed soccer simulation teams can be naturally measured by the results of the matches between them. In addition, we knew and understood very well from the beginning that it will be hard to find successful soccer algorithms that can reproduce the distributions observed in reality. The reason for introducing competitions into our SPI was that we wanted to support this search.

1) Incorporating Gamification Elements: Competitions can be interpreted as games between developers, where the winning itself is the direct reward of competitions. In this sense, stating the most challenging research problems and development tasks as competitions can be regarded as gamification, or rather, as ludification ([12], [29]).

As a classic gamification element, we have developed a point system to indicate the difficulty and also the monetary value of competition tasks. We represent values with quaternary (base 4) numbers, where digits are symbolized by balls. In our notation, a "classic soccer ball" denotes 0, a "silver ball" denotes 1, a "gold ball" denotes 2, and finally, a "fire ball" denotes 3.

2) Selection of the Winner: A question that must be answered is how a winner is selected from a set of competing rapid prototypes. In the case of soccer team simulation algorithms (ie., MABSA and FANM soccer teams) the winner can be naturally determined by simulating a tournament among the competing teams. In general, selecting the winner is straightforward in the case of competing simulation algorithms, ie., the winner is the one that produces results closest to reality. For this and other reasons our intuition suggests that developing simulation algorithms is such an area where the CP methodology can applied very effectively.

In other cases (eg. porting FANM soccer teams to CUDA) the goal of the competition is to minimize or maximize a predefined objective function (ie. the number of parallel threads of soccer matches in a CUDA block). If that is the case, the selection of the winner is straightforward.

On the other hand, there are cases where the winner is determined by some other mechanism. For example, the logo of the FootballAvatar project was also selected in a competition by a voting procedure (see Section III-H).

3) Our Best Practices to Set Up Competing Development Teams: As shown in Table III, we have organized competitions in ten different research and development fields that are named in the first column and will be detailed in Sections III-E, III-F, III-G and III-H. Starting from the second column, the

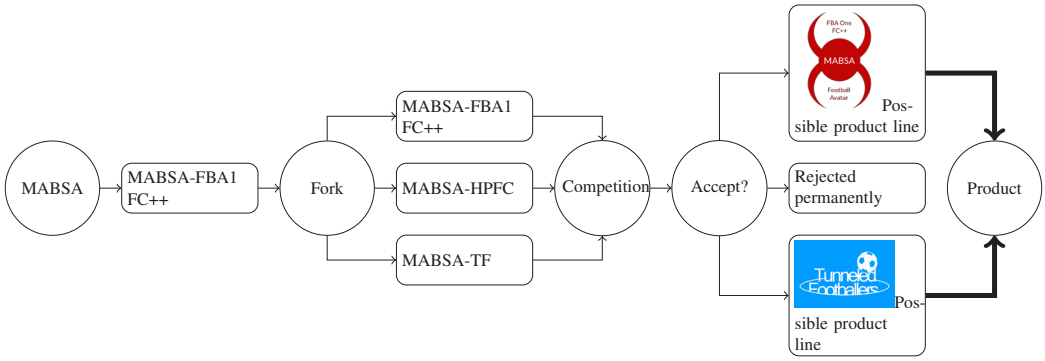


Fig. 2: An arrangement of MABSA rapid prototypes in our competition programming flow chart. MABSA acronyms are explained in detail in Sect. III-E.

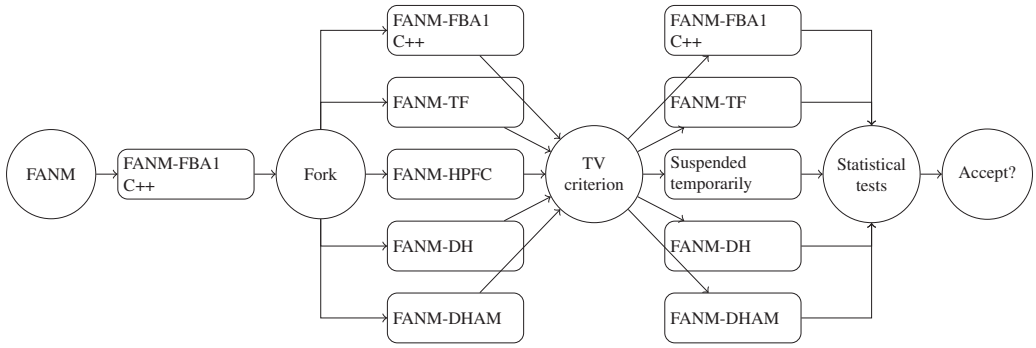


Fig. 3: An arrangement of FANM rapid prototypes in our competition programming flow chart. FANM acronyms are explained in detail in Sect. III-F.

activities of our researchers and developers are shown. In the table, an "1" denotes that the person represented by the column successfully took part in at least one competition in the field. "0" denotes non-participation, while an underlined "0" stands for unsuccessful participation.

Our experience shows that the most successful development teams consist of one or two members. All competing teams with more than two members were unsuccessful, and their activities had to be reorganized or, in several cases, had to be suspended temporarily or cancelled permanently.

In addition, our experience suggests that there is a strong correlation between taking part in the work of the successful teams and the number of commits (shown in the last row of Table III). It is not surprising, as we maintain all source code and documentation under version control. We also noticed that the activity of our researchers and developers follows a Pareto-like distribution in that sense that 80 percent of the commits were made by 20 percent of the members (see Fig. 4).

We recommend that the whole project team should include two guru programmers, two mathematicians, about 10 software engineering students, and a project owner together with further technical experts from the target area of the system to be developed (in our case, they are soccer experts). The development must be managed by the project leader who should be one of two programming gurus. The project leader should continuously monitor the activity of the others. This monitoring process is based on the number of commits per month and until it follows the early mentioned Pareto-like distribution, the project leader makes proposals to include new members in order to help, reorganize or replace members who are in the tail of the Pareto-like distribution.

C. Using Open Source Software and Licensing Issues

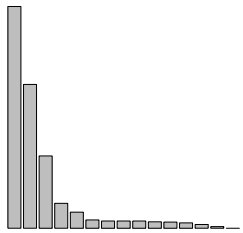
Open source software have become a key factor in the IT industry. According to a Gartner survey conducted in 2008 [30], 85% of companies already used open source software, while the remaining 15% percent expected to do so in the next 12 months. (A total of 274 companies took part in the survey from around the world.) In their more recent report [31] they expect open source software to continue to broaden its presence.

Open source has also become a business, a number of companies are specifically set up to develop and distribute open source software. Even terms such as *professional open source (POS)* [32], *OSS 2.0* [33], and *second-generation open source (OSSg2)* [34] were coined to refer to commercially developed open source software. Enterprises can also benefit a lot from using open source software to develop their own software products (for example, see [35]).

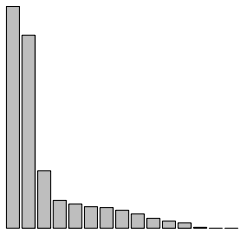
Therefore, to exploit inherent advantages we use a lot of open source software in the development of the FootballAvatar system.

TABLE III: Competition in the FootballAvatar project. The rows show different research and development fields, and the columns represent the activities of our researchers and developers, see text for detailed explanation.

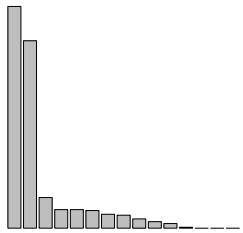
				*		*	*	*		*					
MABSA	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
FANM	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0
FANM+	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0
FANM (CUDA)	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0
2D	1	1	1	0	0	1	0	0	0	0	0	0	0	1	0
3D	1	0	0	0	0	1	0	1	0	1	0	0	1	1	0
MOBILE	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
ANALY "0"	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
ANALY "−1"	1	1	1	0	1	1	0	0	1	0	1	0	0	0	0
BRAND	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0
# commits in $[T_1, T_2]$	835	538	270	31	14	27	23	24	16	<10	20	60	93	27	<10
# commits in $[T_3, T_4]$	250	216	64	31	14	27	23	24	16	<10	20	<10	11	<10	<10



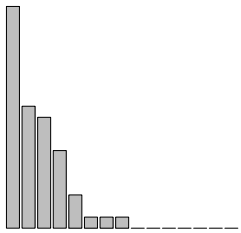
(a) The total number of commits in our repositories in a given time interval $[T_1, T_2]$.



(b) The total number of commits in our repositories in a time interval $[T_3, T_4]$, soon after when new developers joined to the project, where $[T_3, T_4] \subset [T_1, T_2]$.



(c) The number of commits in our source code repository in a given time interval $[T_1, T_2]$.



(d) The number of commits in our source code repository in a time interval $[T_3, T_4]$, soon after when new developers joined to the project, where $[T_3, T_4] \subset [T_1, T_2]$.

Fig. 4: Anonymized bar charts showing developer activity, where each bar represents the number of commits by a developer. Here the shapes of the distributions are significant, rather than the exact number of commits. In figures 4(b) and 4(d), it is shown that performance shifts slightly towards a distribution with a heavier tail after new members had been included. Our project management also uses these charts for the assessment of the work done by the developers.

1) OSS Policy: It is a common practice among enterprises to develop and maintain an *open source software policy* (or OSS policy in short), in which they specify the accepted uses of open source components and their license requirements for third-party open source software (for a comprehensive treatment of this topic, see [36]). It should be noted that, according to the Gartner survey conducted in 2008 [30], 69 percent of companies examined had no such a formal policy. The study also concluded that companies should have an OSS policy.

To assure that their licensing terms will not interfere with our closed source business model we maintain a list of all third-party software used for the development of the system. Developers have to name each third-party software they would like to use and our licensing experts decide on a case-by-case basis if their licensing terms comply with our licensing policies. If the licensing experts reject the use of a third-party software they try to suggest appropriate alternatives too.

We call the organizational process described above as *License Approval Process* (LAP). The process also includes legal consultation with third-party software vendors (see later).

2) An insight into our OSS policy: Since the majority of the system will be proprietary (non-free) software it limits the use of free and open source components. For example, any code released under a copyleft license like the GNU GPL cannot be incorporated into non-free software. Therefore, we prefer using free and open source libraries distributed under commercial friendly open source licenses, such as the Apache License, the X11 License, or the various BSD licenses. (For a comprehensive overview of the popular free and open source licenses see [36].) We use GPL'd code only as a last resort for standalone components that we make available as free and open source. The project company allows open source developments only in certain areas of FootballAvatar, but these components are independent from the simulation core. Note that the GNU GPL does not limit the use of the output of a program distributed under its terms. (See the question "In what cases is the output of a GPL program covered by the GPL too?" in [37].) Thus, we can use GPL'd software in the development of FootballAvatar. For example, we use Blender and GIMP to create content.

Some free and open source licenses require special attention

in non-free projects, one such license is the GNU LGPL. Although it is a so-called weak copyleft license that is intended to be appropriate for non-free applications, it is not without problems (for example, see [36]). Developers must permit the modification of any LGPL'd libraries they use and also the reverse engineering of their own code for debugging such modifications. Since reverse engineering is undesirable in closed source projects, we try to avoid GNU LGPL and use LGPL'd software only as a last resort when there is no appropriate alternative (an example is the Qt framework). We use LGPL'd code for free and open source components. If our non-free code must be linked with LGPL'd code we permit reverse engineering for parts of our system.

Licensing also requires special care because in one of our sales model we will distribute the FootballAvatar system in binary form bundled together with all required third-party software, even with a complete Linux operating system. Fedora has been chosen as our primary free and open source Linux distro. We consulted with the legal team of the Fedora Project and they have authorized us to distribute our system together with Fedora under the name Fedora Remix.

a) Example of Rejection: For example, the use of Ubuntu Touch as a target platform for our mobile interface was rejected. Ubuntu Touch is an Ubuntu Linux distribution for touchscreen mobile devices developed by Canonical Ltd. At present, it is experimental software for evaluation purposes only that is available free for non-commercial use. Although it is a promising mobile platform its current licensing terms are not appropriate for us since they prohibit any commercial use.

3) Types of Third-party Software: It is worth noting that we distinguish the following four category of software in our list of third-party software maintained within the LAP: (1) libraries (such as Mesa and Qt), (2) runtime environments (such as R and WordPress), (3) developer tools (such as Apache Maven and Blender), (4) and other content (such as fonts and artwork). These categories represent slightly different uses of third-party software. For example, developer tools include build automation software used by our developers to build the FootballAvatar system from sources and graphics software used by our artists to create original artwork, such as product logos. These tools don't have to be bundled with the final product.

III. An Overview of the FootballAvatar System

The results of the research and development activities of FootballAvatar can be grouped into following three basic categories:

(1) Actually, software components in the *FBA Core* category constitute the FootballAvatar system. They are distributed under a closed source proprietary license. (2) The elements of the category *FBA Add-Ons* are additional components whose terms of use are different from than those of the *FBA Core*. For example, *FBA Add-Ons* contains open source software. (3) The elements classified in category *FBA Exper* shall not be included in the FootballAvatar product, for example, because of licensing problems.

In the following, we review the FootballAvatar Soccer Simulator Collection that we also refer to as the Multi-speed Simulator.

A. A Quick Glance at the Multi-speed Simulator

The main functionality of the FootballAvatar system is organized into successive simulation levels. We refer to this layered architecture as *Simulation Oriented Architecture*, or SimOA for short. The following three main levels (or speeds) can be distinguished in the FootballAvatar Soccer Simulation Collection: (1) On the level labeled "-1" simulation algorithms use only publicly available or estimated data based on objective and/or subjective observations. (2) Level "0" uses dedicated equipment such as video cameras and sensors to gather data (this infrastructure is provided and operated by our partners). (3) Higher ("$+$") levels can be built on the lower ones.

The software elements on each layer can typically operate in the following three additional modes: standalone mode, analyzer mode, and avatar simulation mode. (1) *Standalone* simulators and analyzers operate independently on both the data provided professionally and available publicly. One important use of the standalone elements is to generate test data. (2) Software in the *analyzer* mode can be used to examine test data or real soccer data. This mode uses advanced data mining and statistical techniques such as bivariate Poisson regression models, see also [38]. (3) The *avatar simulation* mode is the basis of the comparison of real and simulated soccer matches.

The main difference between the standalone and the avatar simulations is that the former has no input at all or its input is not decisive. However, in the case of avatar simulations the representation and behaviour of the players and teams depend very heavily on the input.

In addition, from an architectural viewpoint, we distinguish the following three environments: (1) *MABSA* (Multi-Agent-Based Server Architecture) is a multi-agent system to simulate and analyze soccer matches. This environment is for research purposes only. (2) *FANM* (as opposed to MABSA) software elements neither use networking nor agent technology. These models purely focus on soccer simulation algorithms. (3) *FANM+* elements supplement FANM with additional features. For example, FANM+ contains FANM algorithms ported to Nvidia's CUDA platform.

Finally, from the aspect of implementations, we have several reference implementations, such as FBA One, Tunneled Footballers, and Hungarian Phoenix, that represent potential product lines.

In the next paragraphs, we give a detailed insight into the multi-speed simulator. It is important to emphasize that all presented models and components have been developed in the developer competition stage.

B. Speed "-1"

The level "-1" of the FootballAvatar system consists of simulation algorithms that use only publicly available or estimated data based on objective and/or subjective observations. Among others, the algorithms on this level make prediction of

the outcomes of various events that occur during matches for the coaches of a football team. These events are, for example, the result (win, draw, or lose) of a match, the number of goals, faults, yellow and red cards. It should be emphasized that we do not want to compete against betting offices by developing winning strategies for the gambling market. The main goal of the level "−1" is to help a football team providing useful information such as the comparison of the strengths of the teams who will be playing at the weekend or whether the next opponent has a taste for more faults than usual.

C. Speed "0"

Several performance metrics are used in professional football, of which the best known is the EA SPORTS Player Performance Index (PPI). According to [39], PPI is used in the Barclays Premier League to analyze the performance of players. PPI's strength is its ability to measure the performance of players independently of their playing position. It is also interesting to notice that PPI is based on a published research paper [40].

In the FootballAvatar project we are working on a conceptually similar index but its development is in a very early stage. We are experimenting with different kinds of indices. For instance, we have some promising results with the use of the Similarity Metric [41]. However, the software components in question are experimental and not part of *FBA Core*, they are assigned to *FBA Exper*. "Decision Processor" is a standard *FBA Core* coaching staff's tool in FootballAvatar that can help to classify the decisions of players in a discrete time scale. The output from this tool can be used as a basis for building performance indices. We experimented with several *FBA Exper* solutions based on the data produced by "Decision Processor".

D. Higher Speeds

Higher speeds are based on FANM simulations. Typical use cases for the higher speeds are the following: (1) Simulating full championships, such as national cups, the Champions League, or the World Cup, including the creation of match calendars (match scheduling). (2) Extending simulations by incorporating additional models, for example, physiological ones. Currently, the system uses two such extensions, a player stamina model and a foul model. The above mentioned stamina and foul models affect simulations by modifying the properties of players (also referred to as avatar properties) as a function of time. These models will be discussed in detail in a further paper that is currently under development. The modifications of the properties are implemented via AOP (Aspect-Oriented Programming) [42] aspects.

E. MABSA: MultiAgent-Based Server Architecture

MABSA is an acronym for MultiAgent-Based Server Architecture. It is a TCP/IP-based client-server framework developed in C++ from scratch by the first author. MABSA is based on the Berkeley Socket API and uses IO multiplexing to control both the server and the clients. The main feature of this architecture is that simulation algorithms can be connected to the server in the same way as client agents (see Fig. 5).

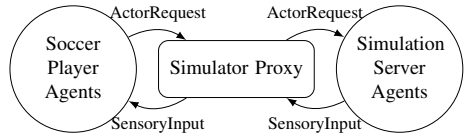


Fig. 5: The MABSA simulation architecture. Simulation algorithms are also implemented as agents, therefore they can be easily plugged in or replaced. This flexibility makes the search for good simulation algorithms easier.

MABSA soccer teams and simulation algorithms have been developed in competition. In the following, we will refer to a MABSA soccer team and a simulation algorithm together as a *MABSA implementation*. Initially, we had three competing implementations that we refer to as MABSA reference implementations.

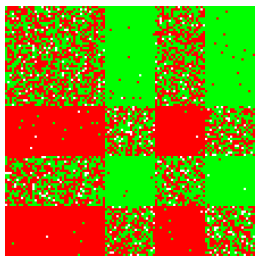
1) The MABSA Communication Protocol, Simulation Algorithms and Soccer Teams: The implementation of the communication protocol between simulation and analyzer algorithms and also simulated players is based on Google's Protocol Buffers [43]. The protocol was developed in accordance with our competition based methodology. Originally, MABSA development teams used their slightly modified versions of an initial protocol. Currently, developers use a unified protocol that represents a consensus among them and is a result of an iterative development. The protocol is organized around two communication classes, namely, *SensoryInput* and *ActorRequest*: the former encapsulates all inputs available to clients (i.e., players), while the latter their responses to their input. The exchange of *SensoryInput* and *ActorRequest* objects between agents (i.e., players and simulation algorithms) is via a simulator proxy.

a) MABSA-FBA1 FC++: It is the first author's C++-based reference implementation which has been developed as a rapid prototype. Directly or indirectly, the other development teams used it as a basis for their own MABSA implementation. MABSA-FBA1 FC++ uses a discrete-time simulation algorithm that will be detailed in a further paper.

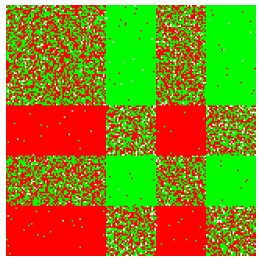
b) Hungarian Phoenix FC (HPFC): Because this development team failed to comply with the time-limits for the development of a working MABSA soccer team, it was suspended temporarily. Then, after an unsuccessful reorganization it was cancelled. The lack of success of HPFC may be explained by the "guru problem" in the sense of [44], since members had not enough experience with network programming and soccer modeling.

c) Tunneled Footballers: It is the third author's C++-based MABSA implementation. It uses a discrete-time algorithm that calculates the motion of the ball based on the Runge-Kutta method. The players can interact with the ball and with their environment with the following commands: *stand*, *move*, *kick*, *catch* and *tackle*. They also have attributes that determine the effectiveness of their actions, namely, speed, stamina, power, ball-control, dribbling, tackling.

The developer has created an own display program called



(a) This figure shows the result of 24,200 simulated matches on a GeForce GTX 560 Ti card, organized into 484 (22 × 22) threads per CUDA block.



(b) This figure shows the result of 51,200 simulated matches on a GeForce GTX 660 Ti card, organized into 1024 (32 × 32) threads per CUDA block.

Fig. 6: The Hungarian flag notation for visualizing CUDA simulation results, where a red pixel denotes a win, a white one a draw, and a green one a defeat (for the home team).

FootballEye, which can be seen in action in Fig. 10. This program serves as a testing and debugging tool for the developer.

F. FANM: FANM is Not MABSA

FANM is a recursive acronym for FANM is Not MABSA. The term denotes the second of our two architectures for soccer simulation. While the MABSA architecture primarily serves as an internal research platform, FANM is intended to be used in the end product. It supports easy embedding of simulations algorithms to the system. FANM algorithms can be ported to run in extremely parallel computing environments, such as CUDA GPUs.

1) FANM Simulation Algorithms and Teams: Similarly to MABSA, FANM implementations have also been developed in competition. We have five different competing FANM development teams, namely, FANM-FBA1 FC++, FANM-TF, FANM-HPFC, FANM-Debrecen Handsomes, and FANM-Debrecen HardAsMuscle.

a) FANM-FBA1 FC++: The name stands for a FANM reference implementation written in C++ by the first author. It is a rapid prototype that is easily portable to CUDA. Its simulation algorithm is developed from scratch and uses a FerSML-like control in that sense that it has been organized around the motion of the ball. A skeleton of the CUDA ported version of this implementation is shown in Listing 1. Fig. 6 shows our visualization technique called Hungarian flag notation, where the columns correspond to the line-ups of the away team and rows correspond to the line-ups of the home team. In this experiment both teams used the same five line-ups, but the 3rd and 5th line-ups were detuned to get better results for the home team. In this figure each pixel represents the result of a match. The figure itself reflects that the results are correct.

b) FANM-TF: It is a FANM implementation written in C++ by the third author. FANM-TF is easily portable to CUDA. Its simulation algorithm was taken from the MABSA implementation called Tunneled Footballers, but it uses neither sockets nor agents.

Listing 1: CUDA skeleton code for the FANM reference implementation FANM-FBA1 FC++.

```
__device__ void
fanmsimu ( /* blockIdx.x, blockIdx.y, dcurandinit,
 dfavatars, dlineups, dpasses, dresults */ )
{
  // GPU
  // A SOCCER SIMULATION
}
__global__ void
setupfanmkernel ( /* dcurandinit */ )
{
  // GPU
  ...
  curand_init ( ... );
}

__global__ void
fanmkernel ( ... )
{
  // GPU
  ...
  fanmsimu ( /* blockIdx.x, blockIdx.y, dcurandinit,
   dfavatars, dlineups, dpasses, dresults */ );
}

int
main ( int argc, char *argv[] )
{
  // CPU
  ...
  dim3 grid ( 5, 10 );
  dim3 tgrid ( 32, 32 );
  ...
  setupfanmkernel <<< grid, tgrid >>> ( /* inic. */ );

  fanmkernel <<< grid, tgrid >>> ( /* drndinic, dfavatars,
   dlineups, dpasses, dresults */ );
  ...
}
```

c) FANM-HPFC: Although it was intended to be a complete FANM implementation, this software works in the analyzer mode only. The development of the standalone and avatar simulation modes was cancelled due to problems inherited from MABSA-HPFC. Another reason for the failure was that, unlike other FANM implementations (e.g., FANM-Debrecen Handsomes) FANM-HPFC does not reuse code from other implementations, it was intended to be written in Java from scratch.

d) FANM-Debrecen Handsomes: This FANM implementation is currently under development by the fifth and ninth authors. It is written in C++ and extends FANM-FBA1 FC++ with the use of a stamina model mentioned in Sect. III-D. Several factors affect the stamina of a player, these factors can be classified into two groups. First, any activity (e.g., movement, passing, dribbling) by the player has a direct impact on the stamina. Second, physiological properties (e.g., blood pressure, blood chemistry characteristics) also modify it.

e) FANM-Debrecen HardAsMuscle: This FANM implementation is developed by the eighth author. It is written in C++ and is based on FANM-FBA1 FC++. There are a few, but significant differences compared to the other FANM implementations. For example, the simulation algorithm uses a foul model that assigns the probability of committing a foul to each player.

G. User Interface

Two main types of users are distinguished in the FootballAvatar system, end users and a special user called FootballAvatar Operator, or FAO for short. Every software product has end users. In our case, they are coaches, managers, and executives of a football club. The FAO is an expert user with a deep knowledge and understading of the FootballAvatar system, whose main task is to provide end users with all information they require, such as simulation results.

Consistent with the above, we distinguish the following two user interface levels: (1) The first UI level is specifically for the FAO. It is a PC-only interface, and its main goal is to provide functionality to run simulations and to provide information for end users. (2) The second UI level is specifically for end users. It is built on the information provided by the FAO. UI implementations must be portable and easy to use, so this interface is only available on portable devices (i.e., on tablets and smartphones).

The organization of the user interface reflects the SimOA introduced in Sect. III-A, i.e., the FAO must choose one of the simulation levels before use.

2D and 3D display applications, mobile solutions, and data analysis applications are typical elements of the user interface. Similarly to MABSA and FANM implementations, they are also being developed in competition with each other. We have five 3D, four 2D, and four mobile-based competing display applications, and eight competing data analysis applications. Only some of them will be presented here.

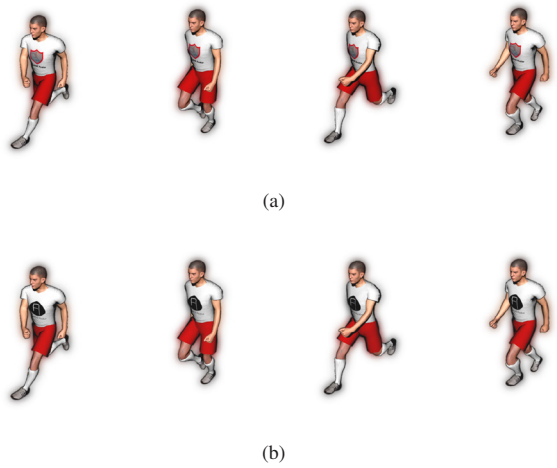
f) Competing 3D Display Applications: Basically, two kinds of 3D display applications are being developed to be used in the FootballAvatar system: an anthropomorphic that uses 3D animated human models, and a schematic that uses buttons to represent players. The latter solution is also referred to as "button soccer display". For example, *3D Model Animation (3DMA)* is an anthropomorphic display application developed by the eleventh author. Fig. 7 shows a 3D model of this application, that was created by Blender [45] and MakeHuman [46].

We have experimented with several "button soccer" type displays. For example, an OpenGL-based display application made by the development team FBA1 is shown in Fig. 8.

g) Competing 2D Display Applications: We have four different competing solutions for 2D display and match analysis. For example, *Multi-Display Player*, or MDP for short, was designed to display multiple visualizations at the same time using screen splitting. It is intended to be used in tactic rooms equipped with a big screen or a projector. MDP provides a flexible layout architecture to which additional visualizations (e.g., new charts) can be added easily.

As another example, Fig. 10 shows a Qt-based display and analytic application that works with both the MABSA and the FANM architecture too and was very successful in the developer competition phase.

h) Competing Mobile Solutions: Fig. 11 shows a screenshot from our four competing mobile solutions. For example, *View 2D Entity (V2DE)* is one of the MDP modules designed to display a football pitch in 2D and to perform basic analytics.



(a)

(b)

Fig. 7: 3D models of a football player used by the *3D Model Animation (3DMA)* display program. The two models differ only in the shirt logo.

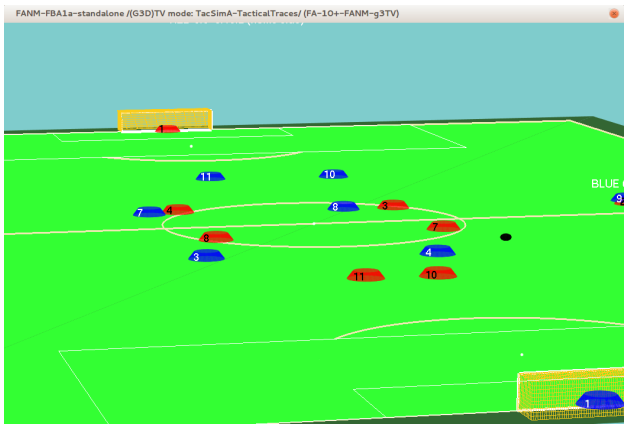


Fig. 8: The OpenGL-based display program of the development team FBA1.

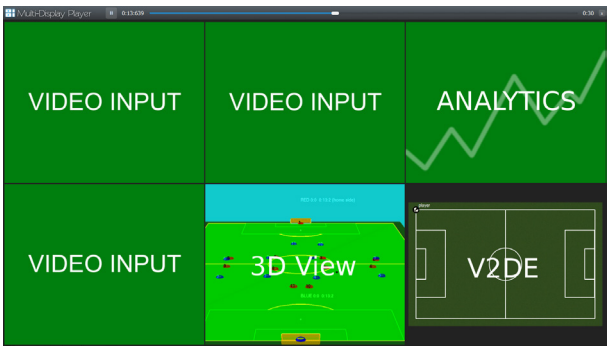


Fig. 9: The modular layout of Multi-Display Player (MDP).

V2DE can run detached from MDP as an independent cross-platform application on Windows, Linux, and also on Android tablets.

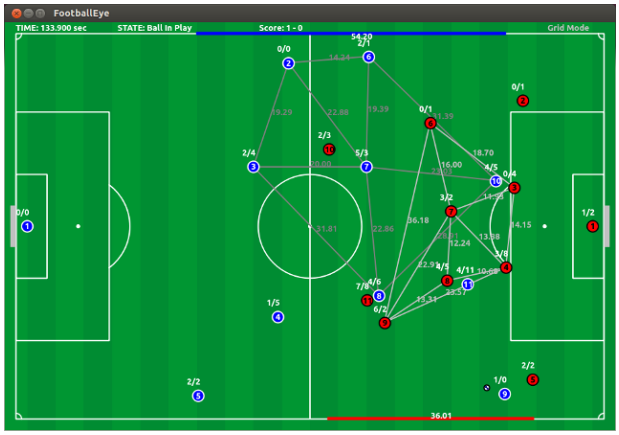


Fig. 10: FootballEye, a QT-based display program of the development team Tunneled Footballers.

(a) The 2D player called FA-droid.

(b) The 2D player called View2DEntity.

(c) Tactic Design Table.

(d) FootballEye with QT5 running on Ubuntu Touch.

Fig. 11: Our four tablet-based solution proposals in the developer competition stage.

H. Project Logo

In accordance with our methodological approach, CP, two competing development teams were formed for the creation of the FootballAvatar logo. We call the resulting logos the *Minimal FootballAvatar Logo* and the *Shield FootballAvatar Logo*.

i) The Minimal FootballAvatar Logo: This logo was created by a one-member team consisting of the sixth author. The logo depicts a letter "F" and a letter "A" that are merged together. The design goal of the artist was to create a logo that is simple and easy to remember.

j) The Shield FootballAvatar Logo: This logo was created by a two-member team consisting of the seventh and the eleventh authors. The design goal was to create a minimalist logo that looks similar to football club logos. The football player shown in the logo is based on a 3D model by the eleventh author.

(a) The minimal FootballAvatar logo.

(b) The shield FootballAvatar logo.

Fig. 12: The two competing FootballAvatar logos.

I. Football Avatars and Avatar-based Simulations

In this section a heuristic mathematical definition is given for the notion of football avatar. The definition is based on statistical hypothesis testing and goes back to [23]. It should be noted that the authors of this paper are currently working on a separate paper devoted to the concept of football avatar.

Let $x = (x_1, \ldots, x_p)$ be a p-dimensional random variable, where x_i, $i = 1, \ldots, p$, are quantities characterizing soccer matches from particular aspects. There are typically such quantities that depend on chance, but their values are known in reality, such as the number of passes of a given player in a match, or the total number of goals scored by a team or a player in a season. The coordinates of x are referred to as probabilistic properties of the soccer. Realizations of this random vector x (known from real soccer) will be called *a-priori observations*. Let $\mathcal{A}$ be a soccer simulation algorithm based on various player, pitch and/or referee features as avatar features that computes independent *simulated observations* for x.

Definition (A Heuristic Definition of Football Avatar)**.** *The pair $(x, \mathcal{A})$ is referred to as football avatar with respect to a given set of probabilistic properties if the probability distribution of the a-priori observations and the probability distribution of the simulated realizations of x are equal.*

The equality of these two probability distributions can be verified by appropriate methods of hypothesis testing.

Statement. *The TF-FANM-NB1 algorithm is a football avatar with respect to the total number of goals scored in a season.*

Following the example of [23], in order to verify the statement we have used the Wald–Wolfowitz and Mann–Whitney tests to determine whether the two distributions are the same or not. The significance level α for all following tests is chosen as $\alpha = 0.05$. In Table IV we have collected the total number of goals scored during seasons 2004/2005–2012/2013 in the Hungarian National Championship.

Season	4/5	5/6	6/7	7/8	8/9	9/10	10/11	11/12	12/13
Goals	681	707	677	746	710	707	690	648	639

TABLE IV: The total number of goals scored during seasons 2004/2005–2012/2013 in the Hungarian National Championship ($\overline{x} = 689.44$, $s_n^* = 33.02$).

Then, we have simulated nine seasons for this championship using the TF-FANM-NB1 algorithm, for which the total number of goals scored obtained from the simulations are shown in Table V. We refer to this sequence of simulations as a.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	test stat.
a	684	696	735	775	780	709	693	695	705	10/24
b	692	720	703	648	689	712	680	708	697	13/34

TABLE V: The total number of goals scored in our simulations ($\overline{x}_a = 719.11$, $s^*_{n_a} = 36.08$ and $\overline{x}_b = 694.33$, $s^*_{n_b} = 21.28$). The last column shows the values of test statistics in form of Wald–Wolfowitz/Mann–Whitney. The structure of the table is the same as the previous one.

Another sequence of simulations, b was run with different parameters in comparison with a. Finally, it should be noted that all these simulations have been computed on the level "-1" in the standalone mode.

1) Simulation-based Decision Making Support: Based on "avatar simulations" introduced in the previous section, we can answer many questions that may be of interest to the coaching staff, such as the following: (1) Which starting eleven should be chosen? (2) Which line-up should be chosen? (3) What is the likely impact of given tactical orders? (4) What is the most likely match result? Details of these topics will be discussed in further works.

J. Project Documents and the Documentation Process

The FootballAvatar project uses DocBook as its primary documentation format. DocBook [47] is an XML vocabulary for writing technical documentation. It is an open standard that is maintained by the OASIS DocBook Technical Committee. Docbook is popular and widely used in the industry. For example, the following projects use DocBook for their documentation: the Fedora Documentation Project, FreeBSD, GNOME, KDE, PHP, PostgreSQL, the Linux Documentation Project. The main advantages of DocBook are the following: (1) it is a platform and vendor independent plain-text based format, (2) DocBook documents can be transformed into various presentation formats, including HTML and PDF.

An important additional advantage comes from its plain-text nature: DocBook documents can be stored under version control and developers can work concurrently on the same document. Therefore, we keep all documentation in our SCM repository, similar to source code. This allows us to track changes and also the activity of our developers.

We use dblatex [48] to create high quality PDF documentation from DocBook XML sources. It is a free and open source tool distributed under the GNU GPL that uses the LaTeX typesetting system to transform DocBook into PostScript or PDF.

It is mentioned here that weekly team meetings and other project activities are recorded on video as part of our documentation process.

Developer's Guide: We follow the principles of agile documentation [49]. In order to implement these, we maintain only one document simply called "Developer's Guide", which includes the Conceptual Plan, the Software Requirements Specification, and the System and Implementation Plan together. It also documents our competition and open source policy related activities using the competition and the open source submission forms shown in Tables I and II.

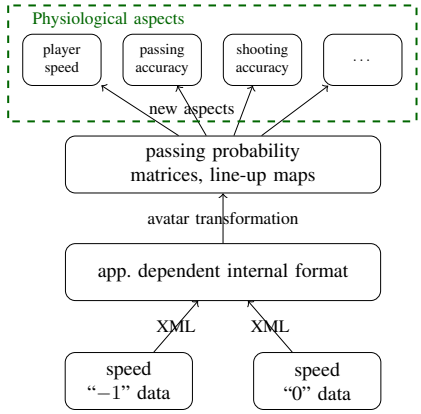


Fig. 13: An avatar is a cross-cutting aspect that can be used to prepare the input files for the simulation algorithms.

IV. THE IT MANIFESTATION OF AVATARS

In Sect. III-I we showed that the notion of "football avatar" is a mathematical definition that can help to evaluate the goodness of soccer simulation algorithms, where the evaluation is based on the comparison of the simulation results and real data. Such a comparison was done in the statement of Sect. III-I where the passing probability matrices and the line-up maps were used. However, it was necessary to include additional properties into the simulation in order to satisfy the TV criterion and statistical requirements. In all cases, the cause of introducing additional properties is that we have applied new approaches to the given simulations. Therefore, an avatar is simply an aspect from an AOP-based [42] viewpoint. To be more precise, avatars are the results of applying avatar transformation aspects to the input of simulation algorithms, as is shown in Fig. 13. In our experiments, the most basic football avatars consist of passing probability matrices and line-up maps. These properties can be extended by introducing new aspects, in which case the appropriate functions of the simulation algorithms must also be overridden to handle new input data by the newly introduced aspects. A few examples for such additional aspects are the following: (1) Physiological aspects: incorporate additional physiological properties into the simulation that affect the performance of players, eg., a stamina model. (2) Environmental aspects: incorporate environmental properties into the simulation, such as current weather conditions, or the type of playing surface. (3) Referee aspects: incorporate properties of the referees into the simulation.

V. VALIDATION OF SIMULATIONS

The following questions naturally arise: (1) How can FootballAvatar simulations be validated? (2) What kind of results

can be expected from the FootballAvatar system? (3) Who knows what can be expected from a computer program in the field of soccer? We are looking for answers to these and similar questions. To be able to answer them, we have to take into account that what can be expected at all from a human coach. Thus, we asked for help from football clubs. At our request, they provided us with preliminary tactical scenarios prepared for their matches. We would like to ground our answers to the above questions on the comparison of the scenarios from football clubs and the scenarios that occur in our simulations. This requires further work, that will be presented in a later publication.

The concept of football avatar, in the strict sense of the definition, by itself, implies a validation criterion (i.e., a statistical test). Some examples of the simulations referred in the definition of football avatar were presented in detail in Sect. III-I.

VI. Conclusion

Finding successful soccer simulations algorithms that can reproduce the distributions observed in reality have proved to be a challenging research and development task. We have known this since the start of the project. To help the search for suitable simulation algorithms we have developed our own methodology that we named as Competitive Programming, or CP for short. We hope that CP can be successfully applied in R&D projects, in which sufficient number of developers are available to allocate multiple competing teams to a specific task. This is often the case when the R&D activity happens fully or partially in a university environment. In order to support the adoption of CP we have made our standard project document templates available at http://footballavatar.hu/CP.

On the basis of the results shown in this paper, it is clear that our CP-based efforts have been successful, because we have found simulation algorithms that can fulfill the definition of football avatar. Thus, in the strict sense the research purpose has been achieved.

VII. Ongoing and Future Work

This section briefly summarizes ongoing and future work mentioned in the paper. (1) We are working on a performance measure that is similar to the EA SPORTS Player Performance Index (PPI) but its development is in a very early stage (see Sect. III-C). (2) The stamina and foul models mentioned in Sect. III-D, and also MABSA simulation algorithms mentioned in Sect. III-E are discussed in detail in another paper that is devoted to the concept of football avatar and has been submitted for publication [38]. (3) We are planning a paper that will address questions regarding the validation of FootballAvatar simulations (see Sect. V).

Acknowledgment

The authors would like to thank the members of the research group "World Football—Modeling and Visualization" at the University of Debrecen for the meetings and their useful comments that help them better understand soccer. During the development of the FootballAvatar system authors worked in cooperation with other project partner companies (namely, U1 Research Ltd., IQRS Ltd., and Satrax Ltd.) and they would like to thank them for their contributions that will be important in the operation of the FootballAvatar system at real football clubs. We, the authors would like to express our special thanks to Tamás Sándor, Péter Szakály, Elemér Kondás, Ferenc Frida and Sándor Szilágyi. Last, but not least, thanks to all members (especially Prof. György Terdik, Tibor Balla and Piroska Biró) of the "Nagyerdei Gerundium" working group of SziMe3D Ltd. for their continued help and support.

The publication was supported by the GOP-1.2.1-11-2012-0005 (*SziMe3D – 3D-s technológiai innovációk a turizmus, oktatás és sport területén*, *SziMe3D–3D technological innovation in tourism, education and sport*) project. The project has been supported by the European Union.

References

[1] N. Bátfai, "Footballer and Football Simulation Markup Language and Related Simulation Software Development," Journal of Computer Science and Control Systems, vol. 3, no. 1, pp. 13–18, 2010.

[2] N. Bátfai, R. Dóczi, J. Komzsik, A. Mamenyák, Cs. Székelyhídi, J. Zákány, M. Ispány, and Gy. Terdik, "Applications of a simplified protocol of RoboCup 2D soccer simulation," Infocommunications Journal, vol. 5, no. 1, pp. 15–20, 2013.

[3] N. Bátfai and Gy. Terdik, "The application of the data of RoboCup 2D soccer simulation league to test several sport science results," 2012, (in manuscript). [Online]. Available: http://robocup.inf.unideb.hu/fersml/assr/

[4] K. Furugaki, T. Takagi, A. Sakata, and D. Okayama, "Innovation in software development process by introducing Toyota Production System," FUJITSU Scientific & Technical Journal, vol. 43, no. 1, pp. 139–150, 2007. [Online]. Available: http://www.fujitsu.com/downloads/MAG/vol43-1/paper16.pdf

[5] O. Salo and P. Abrahamsson, "Integrating agile software development and software process improvement: a longitudinal case study," in Proceedings of the 2005 International Symposium on Empirical Software Engineering (ISESE 2005). IEEE, 2005, pp. 193–202.

[6] M. Fowler and J. Highsmith, "The agile manifesto," Software Development Magazine, vol. 9, no. 8, pp. 29–30, 2001.

[7] K. Schwaber, "Scrum development process," in Proceedings of the 10th Annual ACM OOPSLA, 1995, pp. 117–134.

[8] K. Schwaber and M. Beedle, Agile Software Development with Scrum. Prentice Hall, 2001.

[9] K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd ed. Addison-Wesley, 2004.

[10] B. Fitzgerald, G. Hartnett, and C. K., "Customising agile methods to software practices at Intel Shannon," European Journal of Information Systems, vol. 15, no. 2, pp. 200–213, 2006.

[11] P. Bell. (2007, Jun. 17) Solo Scrums. blog post. Accessed: 2015-12-04. [Online]. Available: http://www.pbell.com/index.cfm/2007/6/17/Solo-Scrums

[12] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon, "Gamification: Using game design elements in non-gaming contexts," in CHI '11 Extended Abstracts on Human Factors in Computing Systems, ser. CHI EA '11. ACM, 2011, pp. 2425–2428.

[13] O. Schenk, M. Christen, and H. Burkhart, "Algorithmic performance studies on graphics processing units," Journal of Parallel and Distributed Computing, vol. 68, no. 10, pp. 1360–1369, 2008.

[14] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, "A performance study of general-purpose applications on graphics processors using CUDA," Journal of Parallel and Distributed Computing, vol. 68, no. 10, pp. 1370–1380, 2008.

[15] NVIDIA Corporation. NVIDIA CUDA Toolkit. Accessed: 2015-12-04. [Online]. Available: https://developer.nvidia.com/cuda-toolkit

[16] J. Pendlebury, H. Xiong, and R. Walshe, "Artificial neural network simulation on CUDA," in Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, ser. DS-RT '12. IEEE, 2012, pp. 228–233.
[17] H. Akiyama, H. Shimora, T. Nakashima, Y. Narimoto, and T. Okayama, "HELIOS2011 team description," 2011.
[18] A. Bai, G. Lu, H. Zhang, and X. Chen, "WrightEagle 2D Soccer Simulation Team Description 2011," 2011. [Online]. Available: http://ai.ustc.edu.cn/en/robocup/2D/tdps/WrightEagle2011_2D_Soccer_Simulation_Team_Description_Paper.pdf
[19] N. Bátfai, "Quantum consciousness soccer simulator," CoRR, vol. abs/1211.2719, 2012. [Online]. Available: http://arxiv.org/abs/1211.2719
[20] R. Koning, M. Koolhaas, G. Renes, and G. Ridder, "A simulation model for football championships," European Journal of Operational Research, vol. 148, no. 2, pp. 268–276, 2003.
[21] R. Koning, "Balance in competition in Dutch soccer," Journal of the Royal Statistical Society: Series D (The Statistician), vol. 49, no. 3, pp. 419–431, 2000.
[22] A. C. Constantinou, N. E. Fenton, and M. Neil, "pi-football: A Bayesian network model for forecasting association football match outcomes," Knowledge-Based Systems, vol. 36, pp. 322–339, 2012.
[23] N. Bátfai, "The socceral force," CoRR, vol. abs/1004.2003, 2010. [Online]. Available: http://arxiv.org/abs/1004.2003
[24] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The robot world cup initiative," in Proceedings of the first international conference on Autonomous agents, ser. AGENTS'97. ACM, 1997, pp. 340–347.
[25] D. Wells. XP flow chart. Accessed: 2015-12-04. [Online]. Available: http://www.extremeprogramming.org/map/project.html
[26] J. Martin, Rapid Application Development. Macmillan Publishing Co., 1991.
[27] S. Halim and F. Halim, Competitive Programming 3: The New Lower Bound of Programming Contests, 3rd ed. Lulu, 2013. [Online]. Available: https://sites.google.com/site/stevenhalim/
[28] Wikipedia, "Competitive programming — Wikipedia, the free encyclopedia," 2013. [Online]. Available: en.wikipedia.org/wiki/Competitive_programming
[29] M. Bouca, "Mobile communication, gamification and ludification," in Proceedings of the 16th International Academic MindTrek Conference, ser. MindTrek '12. ACM, 2012, pp. 295–301.
[30] L. F. Wurster, "User survey analysis: Open-source software, worldwide," Gartner, Inc., Tech. Rep., 2008. [Online]. Available: http://www.gartner.com/id=757916
[31] A. Raina and L. F. Wurster, "Market trends: Application development software, worldwide, 2012–2016," Gartner, Inc., Tech. Rep., 2012. [Online]. Available: http://www.gartner.com/id=2098416
[32] R. T. Watson, D. Wynn, and M.-C. Boudreau, "JBOSS: The evolution of professional open source software," MIS Quarterly Executive, vol. 4, no. 3, pp. 329–341, Sep. 2005.
[33] B. Fitzgerald, "The transformation of open source software," MIS Quarterly, vol. 30, no. 3, pp. 587–598, Sep. 2006.
[34] R. T. Watson, M.-C. Boudreau, P. T. York, M. E. Greiner, and D. Wynn, Jr., "The business of open source," Communications of the ACM, vol. 51, no. 4, pp. 41–46, Apr. 2008.
[35] M. Ruffin and C. Ebert, "Using open source software in product development: A primer," IEEE Software, vol. 21, no. 1, pp. 82–86, Jan. 2004.
[36] H. J. Meeker, The Open Source Alternative: Understanding Risks and Leveraging Opportunities. John Wiley & Sons, 2008.
[37] Free Software Foundation. (2013) Frequently asked questions about the GNU licenses. Accessed: 2015-12-04. [Online]. Available: http://www.gnu.org/licenses/gpl-faq.html
[38] N. Bátfai, A. Mamenyák, P. Jeszenszky, G. Kövér, M. Smajda, R. Besenczi, B. Halász, Gy. Terdik, and M. Ispány, "Sport Science Soccer Simulations," 2014, submitted manuscript.
[39] EA SPORTS Player Performance Index. Football Association Premier League Limited. Accessed: 2015-12-04. [Online]. Available: http://www.premierleague.com/en-gb/players/ea-sports-player-performance-index/
[40] I. G. McHale, P. A. Scarf, and D. E. Folker, "On the development of a soccer player performance rating system for the English Premier League," Interfaces, vol. 42, no. 4, pp. 339–351, 2012.
[41] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi, "The similarity metric," IEEE Transactions on Information Theory, vol. 50, no. 12, pp. 3250–3264, 2004.
[42] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin, "Aspect-oriented programming," in ECOOP'97 – Object-Oriented Programming, ser. Lecture Notes in Computer Science, vol. 1241. Springer-Verlag, 1997, pp. 220–242.
[43] Google Inc. Protocol Buffers. Accessed: 2015-12-04. [Online]. Available: https://developers.google.com/protocol-buffers/
[44] A. A. Janes and G. Succi, "The dark side of agile software development," in Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software. ACM, 2012, pp. 215–228.
[45] Blender Foundation. Blender. Accessed: 2015-12-04. [Online]. Available: http://blender.org/
[46] The MakeHuman team. MakeHuman. Accessed: 2015-12-04. [Online]. Available: http://makehuman.org/
[47] N. Walsh and L. Muellner, DocBook: The Definitive Guide. O'Reilly Media, 1999. [Online]. Available: http://www.docbook.org/tdg/en/html/docbook.html
[48] B. Guillon. dblatex: DocBook to LaTeX publishing. Accessed: 2015-12-04. [Online]. Available: http://dblatex.sourceforge.net/
[49] A. Rüping, Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects. John Wiley & Sons, Inc., 2003.

Norbert Bátfai is working as an assistant professor in Faculty of Informatics at the University of Debrecen, Hungary. He received his M.Sc. (summa cum laude) in Computer Science in 1998 from the Kossuth Lajos University (KLTE), Debrecen, Hungary. In 1999, he won the first prize in the Java Programming Contest organized by Hungarian Java Alliance: Sun, IBM, Oracle, Novell and IQSoft. In 2004, his company won the first prize in the Hungarian Mobile Java Developer Contest organized by Sun Hungary and Nokia Hungary. In 2008, the Hungarian Chief Information Officers' Association awarded him the IT trainer of the year title. He received his Ph.D. degree in 2011. He won the Pollák–Virág award from the Scientific Association for Infocommunications, Hungary in 2012.

Péter Jeszenszky received his Ph.D. degree in 2012 from the University of Debrecen, Hungary. Currently, he is an assistant professor at the Department Technology at the University of Debrecen. He won the the Pollák–Virág award from the Scientific Association for Infocommunications, Hungary in 2012.

András Mamenyák is majoring in Engineering Information Technology BSc at the University of Debrecen, Hungary. In 2013, he won the first prize in the XDA Tablet Z Development Competition organized by XDA Developers.

Béla Halász is working as the project manager for the FootballAvatar project at SziMe3D Ltd. He has extensive working experience in managing various development projects in different IT fields, from Health Informatics to Software Licence Protection.

Renátó Besenczi is majoring in Engineering Information Technology BSc and working as a research team member at the University of Debrecen, Hungary.

Balázs Kóti is majoring in Software Information Technology BSc at the University of Debrecen, Hungary. He started his studies at the University in 2012. He recieved technician degree from geodesy and geographical information systems at Vásárhelyi Pál Engineering High School and Technicum.

Gergely Kövér is majoring in Software Information Technology BSc at the University of Debrecen, Hungary. He started his studies at the University in 2012.

Máté Smajda is majoring in Software Information Technology BSc at the University of Debrecen, Hungary. He started his studies at the University in 2012, and he is also an experienced football player, winner of the Hungarian National Football Olympiad for Students.

Csaba Székelyhídi is majoring in Software Information Technology BSc at the University of Debrecen. He is a member of World Football Modelling and Visualizing Research Group and FootballAvatar Project. He started his studies at the University in 2011.

Tamás Takács is majoring in Software Information Technology BSc at the University of Debrecen, Hungary. He started his studies at the University in 2012.

János Komzsik is majoring in Engineering Information Technology BSc at the University of Debrecen, Hungary.

Géza Róka is the club director of DVSC Futball Szervező Zrt., the football club of Debrecen, which has won the Hungarian Championship and the Hungarian National Cup six times, the Hungarian Super Cup five times, and has participated in the UEFA Champions League and Europa League group phases. His main area of expertise is sports law, particularly international football law. He is currently working on his Ph.D. thesis, which examines the impact of the EU legal system to the international regulation of football.

Márton Ispány received his Ph.D. degree in 1997 from the Kossuth Lajos University, Hungary. He is an associate professor at the Department of Information Technology at the University of Debrecen, Hungary. Dr. Ispány's research areas include integer valued time series analysis, branching processes, and data mining. He won the Alexits György award from the Hungarian Academy of Sciences and the Pollák–Virág award from the Scientific Association for Infocommunications, Hungary in 2012