Journal of Applied Computer Science Methods
Published by University of Social Sciences

ACSM

Volume 9 Number 2 2017
University of Social Sciences, IT Institute
Społeczna Akademia Nauk
Łódź
Instytut Technologii Informatycznych
ISSN 1689-9636

International Journal of Applied Computer Science Methods

Journal of Applied Computer Science Methods

Published by University of Social Sciences

ACSM

Volume 9 Number 2 2017

University of Social Sciences, IT Institute

Społeczna Akademia Nauk
Łódź

Instytut Technologii Informatycznych
iti

INTERNATIONAL JOURNAL OF APPLIED COMPUTER SCIENCE METHODS (JACSM) is a semi-annual periodical published by the University of Social Sciences (SAN) in Lodz, Poland.

PUBLISHING AND EDITORIAL OFFICE:
University of Social Sciences (SAN)
Information Technology Institute (ITI)
Sienkiewicza 9
90-113 Lodz
Tel.: +48 42 6646654
Fax.: +48 42 6366251
E-mail: acsm@spoleczna.pl
URL: http://acsm.spoleczna.pl

Print: Mazowieckie Centrum Poligrafii, ul. Ciurlionisa 4, 05-260 Marki, www.c-p.com.pl, www.drukksiazek.pl

AIMS AND SCOPE:

The **International Journal of Applied Computer Science Methods is** a semi-annual, refereed periodical, publishes articles describing recent contributions in theory, practice and applications of computer science. The broad scope of the journal includes, but is not limited to, the following subject areas:

Knowledge Engineering and Information Management: *Knowledge Processing, Knowledge Representation, Data Mining, Machine Learning, Knowledge-based Systems, Knowledge Elicitation, Knowledge Acquisition, E-learning, Web-intelligence, Collective Intelligence, Language Processing, Approximate Reasoning, Information Archive and Processing, Distributed Information Systems.*

Intelligent Systems: *Intelligent Database Systems, Expert Systems, Decision Support Systems, Intelligent Agent Systems, Artificial Neural Networks, Fuzzy Sets and Systems, Evolutionary Methods and Systems, Hybrid Intelligent Systems, Cognitive Systems, Intelligent Systems and Internet, Complex Adaptive Systems.*

Image Understanding and Processing: *Computer Vision, Image Processing, Computer Graphics, Pattern Recognition, Virtual Reality, Multimedia Systems.*

Computer Modeling, Simulation and Soft Computing: *Applied Computer Modeling and Simulation, Intelligent Computing and Applications, Soft Computing Methods, Intelligent Data Analysis, Parallel Computing, Engineering Algorithms.*

Applied Computer Methods and Computer Technology: *Programming Technology, Database Systems, Computer Networks Technology, Human-computer Interface, Computer Hardware Engineering, Internet Technology, Biocybernetics.*

DISTRIBUTION:

Apart from the standard way of distribution (in the conventional paper format), on-line dissemination of the JACSM is possible for interested readers.

Contents

Andrzej Cader, Krzysztof Przybyszewski
Application of Deep Neural Networks to Mining the Macro-Characteristics of the Teaching Process from the Current Structure of the Student's Grades 5

Patrycja Pankiewicz
Using The MVC Pattern in the Design of Computer Games 23

Krzysztof Przybyszewski, Andrzej Cader
Rules of Knowledge Bases of it Systems Supporting Education Quality Management in Universities 35

Kamil Zakrzewski
Automation of Software Installation in Service Work 59

Mateusz Sztukowski
Offered Traffic vs Load Dispersion Factor Dimensioning of Mobile Networks 69

DE GRUYTER OPEN

JACSM 2017, Vol. 9, No. 2, pp. 5 - 21

10.1515/jacsm-2017-0006

APPLICATION OF DEEP NEURAL NETWORKS TO MINING THE MACRO-CHARACTERISTICS OF THE TEACHING PROCESS FROM THE CURRENT STRUCTURE OF THE STUDENT'S GRADES

Andrzej Cader, Krzysztof Przybyszewski

Information Technology Institute,
University of Social Sciences, Lodz, Poland
acader@san.edu.pl; kprzybyszewski@san.edu.pl

Abstract

This paper describes an attempt to use deep neural networks to assess the degree to which a student has attained learning outcomes. Assessing these aspects of the educational process is exceptionally tedious and burdensome. One possible solution is the automation of the assessment process through the use of Deep Learning Methods. The relatively small amount of data in the form of available grades, which are essential to train the neural network, is an obstacle. This paper proposes a new method of augmenting the data which solves this problem. It was demonstrated that very effective training for the neural network is possible based on this method, even with a small data group, and that this allows the level of learning outcomes attained by a student to be easily identified.

Keywords: deep learning; e-learning; deep neural networks; assessment of learning outcomes.

1 Introduction

Teaching in most European Union countries is based on the European Qualifications Frameworks (EQF), introduced by a European Council recommendation in April 2008 [1]. The aim is to enable the comparison of qualification levels among EU member states, and in this way to transfer qualifications gained at particular stages of the educational process, so that full mobility at each stage of education and employment is possible within the Union. For each level of learning, the EQF introduce three groups of descriptors, which define the standards which correspond to qualifications at a given level.

Knowledge – defines the extent to which information from a particular area of work or study has been assimilated during the learning process.

Skills – cognitive and practical.

Competencies – related to using knowledge and skills in personal and work-related activities.

The European Higher Education Area (EHEA), a part of the EQF, shapes the harmonisation of qualifications in higher education throughout the EU, on the basis of the Bologna Process [2,3].The EHEA was formally created in March 2010. The Bologna declaration, which began the process, sets out such targets as introducing a system of comparable levels and diplomas, supporting student mobility, guaranteeing quality education, tightening cooperation between educational sectors, research and innovation, and the development of interdisciplinary learning.

Poland introduced National Qualifications Frameworks, NQF (Polish Qualifications Frameworks) in March 2011 in accordance with the EQF, in the form the “Higher Education Law” („Prawo o szkolnictwie wyższym”) [4] and a range of implementing acts which precisely defined the qualification frameworks for individual areas and levels of education [5]. Higher education institutions were charged with describing each programme of education through learning outcomes for individual levels and fields of study.

In the above process, student qualifications are the formal result of the assessment process and validation, gained in a context in which the HE institution states that a student has reached learning outcomes according to particular standards, but does not define the level to which these outcomes have been achieved. Both the migration of students from one institution to another and qualifying graduates who are ready to enter the workplace takes place in the form of this binary classification. The lack of gradation makes it harder or impossible to give the student appropriate grades in a new institution, as assessment scales vary from country to country, and to use competition and selection in the hiring process. Assessing the degree to which individual education outcomes have been achieved would also enable a more precise personal and professional profile to be created – including the strengths and weaknesses of the student or applicant.

The main barrier preventing such gradation is created by the difficulties the teacher (or lecturer) has in defining the degree to which a particular learning outcome in a subject has been achieved. This demands a more in-depth analysis of many student behaviours **of different levels of significance** in the teaching process and of the multifaceted acquisition of knowledge and skills by the student. Teachers do not have time for this, or, frequently, enough psychological expertise. Traditional pedagogical expertise and subject-specific specialist knowledge may not be enough. Additionally, teachers are dealing with everyday teaching and organizational responsibilities. Nevertheless, establishing the extent to which learning outcomes have been achieved at this stage (subject outcomes) is a key element in identifying the overall learning outcome achievement level, which comes at the end (or the end of a particular stage) of a degree. With a grade for subject learning outcomes, it is relatively easy to establish a level for overall outcomes through a learning outcome matrix, for example, in the form of a weighted average, where the wights can be the ECTS points (expressed as fractions) of individual subjects.

One possible solution is to support the teacher at this stage through electronic tools in the form of artificial intelligence methods, particularly with the help of deep neural networks. This tool is exceptionally effective in solving this type of problem, on the condition that a sufficient dataset is available to train the neural network. This paper presents a solution of this kind, using the example of a particular subject. A unique model for the structure of the training data was developed, adjusted to the small dataset base, and two possible neural network architectures were suggested, and these allowed grades to be predicted and established the level of subject learning outcome attainment. The solutions were verified experimentally.

2 Application of neural networks in education

The development of new teaching methods necessitates the use of modern IT tools in this area. Conversely, the development of advanced computer methodology, particularly Artificial Intelligence Methods, stimulates the growth of innovative teaching strategies. In the contemporary teaching model, the emphasis is on the student gaining not just knowledge, but also certain skills which are used in the practical application of this knowledge in work and social relationships. This means that the system of subject assessment used up to now is no longer adequate. The degree to which the various competencies which arise from the knowledge acquired have been developed needs to be assessed. At this time, even the most advanced electronic educational management platforms do not make use of the opportunity created by IT data processing in this area, particularly the use of Artificial Intelligence Methods.

Deep Learning (DL) in particular, after its spectacular achievements in recent years, is starting to have an impact on the field of education as well [6, 7]. Recognizing images with great accuracy [8, 9, 10, 11], text analysis [12, 13], sound analysis [14, 15] are the most important areas in which DL dominates. Apart from classification tasks, DL methods achieve particularly good results in the prediction and modelling of phenomena. In this context, the widespread introduction of DL methods to various educational tasks is not surprising. This is most widespread in Data Mining and Machine Learning. Educational systems archive vast amounts of data, which can be used to obtain knowledge, useful for the further development and improvement of educational methods. DL tools are a natural fit for this area of activity. They are optimal in the context of very large amounts of unstructured data, which they can process with great efficiency.

Deep Learning (DL) methods are used in educational environments for a variety of purposes, most frequently in the analysis of educational quality, the development of new teaching strategies and educational systems. They support the functioning of e-learning platforms in obtaining information about levels of student activity and allow for an individual study programme to be profiled. The use of DL methods in teaching started to be widespread after 2010 [16, 17, 18]. Among the many papers on predicting learning results and forecasting progress in learning, Sharada et al. [19], Kim et al. [20] and Akram et al. [21] should be mentioned. A marked increase in the use of neural networks in teaching is observable over the last few years [22, 23, 24], including in the automation of student assessment [25, 26, 27]. Two areas where automation is relevant can be highlighted. The first is automated essay scoring [28] and the second is automatic short answer grading, which automatically classifies students answers as correct or incorrect, with reference to a collection of previous correct answers [29]. One of the most interesting areas involves attempts to use DL potential in the area of text analysis [30,31]. The most dominant methods here are those based on the use of recurrent neural networks [32, 33, 34], including bidirectional LSTM networks [20].

3 Comprehension and data preparation

Preparing a neural network to solve a particular problem, such as a classification problem, essentially involves implementing the correct models, representing the structure of the problem, among the weights of the network. The representation of the models has to be generalized enough for the network to be able to interpret new states, which do not appear at the network formation stage. This happens at the network training stage. The example presented here uses unidirectional networks based on supervised training, in which updating weights takes place using the error backpropagation and gradient descent optimisation method. This means that a set of training data is needed, consisting of input data (input vectors) and the corresponding output data (labels).

When assessing subject learning outcomes, the input data will be the grades obtained during the course of study, appropriately normalized – a spectrum of grades which the student receives at particular stages of subject acquisition. The output data, meanwhile, will be the grades given for each of the learning outcomes associated with the subject. During the development of the training data set, the teacher plays an essential role (in this case, a panel which was formed especially to implement the neural network), as he or she needs to carry out a manual classification at this stage. The subject “Physics”, covered as part of the first year course of an “Information Technology” degree, was selected for further study, in a group of students consisting of 80 people randomly selected for the training sample and 40 people for the test sample. The assessment panel was made up of experienced teachers of the subject, who also had considerable experience in the introduction of learning outcomes at the institution. It should be emphasized that the work put in by the assessment panel is only necessary at the neural network training stage. After training is complete, the prediction of grades for particular learning outcomes will automatically be carried out by the trained neural network. It was decided that the range of grades should be expanded – a 10 point scale was used, from 0 to 10 points (0 meaning no grade), both for subject grades and learning outcome grades.

In order to obtain an appropriately representative group of subject grades, 12 grade areas (categories) were suggested. These areas are characterised by method and range of assessment:

Kt_1. Ability to create written studies
Kt_2. Ability to prepare projects
Kt_3. Level of solving theoretical sentences
Kt_4. Ability to solve practical problems
Kt_5. Ability to solve tests
Kt_6. Substantive formulation of the oral answer
Kt_7. Participation in the discussion and substantive activity
Kt_8. Participation in consultations
Kt_9. Own work
Kt_10. Creativity
Kt_11. Cooperation as part of group tasks
Kt_12. Timeliness of tasks

Throughout the semester, a teacher could issue numerous grades in each subject category (Table1). The more grades, the more representative the input data set is. It is worth emphasizing that a large set of grades like this is only useful during network training. When using the trained network, only one grade in each category is necessary. If there are more, an average value should be used.

Table 1. Example of student assessments used to train the network

K_W02	K_U06	K_U07	K_K04	Kt_1	Kt_2	Kt_3	Kt_4	Kt_5	KT_6	Kt_7	Kt_8	Kt_9	Kt_10	Kt_11	Kt_12
9	9	9	8	9 8 8	10 8 9	9 9 9	8 8 10	9 8	9	7	5 7	6 8	9	9 8	9 9 8
8	8	8	7	8 9 9	9 6 10	8 5 10	7 8 8	9 10	9	3	5 8	8 7	5	6 7	9 10 9
3	3	3	2	1 2 2	3 4 2	2 3 1	5 4 6	1 9	2	1	1 2	2 1	2	1	5 4 3
4	4	5	5	7 5 5	5 3	5 4 3	7 6 5	4 6	4	4 2	2 4	3 4	4	5	7 8 5
7	7	6	7	7 5 8	6 6 7	6 7 6	7 8 7	7 5	7	6	6 5	7 6	5	7 7	8 10 9
3	4	4	3	2 4 5	4	5 2	4 5 5	4 8	3 4	2 3	2	4	3	3	3 5
3	4	4	3	5 4 6	5 3 4	4 1 6	6 5 3	4 7	3	3	3 2	4 6	2	1	8 9 9

In this situation, the input vectors are 12 dimensional vectors, which corresponds to the number of subject categories. It is clear that when many grades can be given to a particular student in a specific category, each combination of student grades from these 12 categories gives an independent input vector, as long as the grades in each category are independent of each other. Whether they are independent of each other or not is subject to doubt, since during the teaching process, earlier grades can influence each subsequent one, including influencing those outside their category. Nevertheless, these are dependencies which arise during the teaching process, while during final assessment, only the amorphous collection of bare grades without their associated context is significant. When seeking to give an objective assessment, one should be guided only by the grades, in isolation from their conditioning. From this point of view, the grades are independent of each other.

The above indicates that that, for this type of independent data, related to the feature values of the object, each combination of individual feature values may be an input vector for a neural network which classifies the state of the object. It should be explained that individual combinations correspond to particular states, while a sequence of feature values vi represents the generalized state. Thus, when presenting many specific vectors to a neural network, a representation of the generalized state is built. The number of input vectors for each set of data amounts to $\prod_{i=1}^{N} n_i$, where n_i is the number of elements in v_i.

In the examples studied, the panel of teachers suggested learning outcome grades which were used as output data for the network being trained (labels). The learning outcome matrix for Physics shows the following subject learning outcomes in specific groups:

Knowledge:

K_W02 – has basic knowledge of classical and quantum physics, essential in order to understand the principles of computer equipment functioning, and enables rational use of information technology

Skills:

K_U06 – can use mathematical, physical and technical knowledge acquired to describe and simulate processes, create models, save algorithms and other IT tasks.

K_U07 – uses mathematical, physical and technical knowledge to optimize programming solutions; can use appropriate analytical methods and computational experiments to complete IT tasks.

Social skills:

K-K04 – is aware of being responsible for his or her own work, and willing to follow teamwork guidelines, and to take responsibility for collaborative tasks.

In order to establish grades, the panel first developed guidelines to be followed. For every learning outcome, a hierarchy of categories was suggested, which defines the extent of influence on a particular outcome. For example, for outcome K-W02 this influence is defined as (the order indicates to what extent a particular category influence the outcome, from greatest to least influence): Kt_3; Kt_6; Kt_1; Kt_9; Kt_4; Kt_5; Kt_2; Kt_10; Kt_7; Kt_8; Kt_11..

Examples of the grades that were issued are given in the first four columns of Table 1.

The preparation of the training data (from 80 students) consisted of the following stages:

Putting together all grade combinations from the Kt_1 to Kt_12 fields (with one grade from each field), giving each combination the same label, separately for each student.

Random shuffling of each combination.

Dividing the set into train_data and train_labels and standard preparation of batch data, normalising train_data to the range [0,1]. Normalisation consisted of dividing all values by the maximum value, 10.

This procedure allowed 560,688 training data to be obtained. At the stage of choosing the network model and tuning it with training data, a set of 160,000 validating data was temporarily extracted. Test data was prepared on the basis of grades from a separate group of 40 students.

4 Results of the experiment

Two neural network architectures were selected for the research experiment. The first (Arch_1) is based on a sequential model, in which the output layer neurons represent the particular learning outcomes. At output, they can take on a set of discrete values in the range of 1 to 10 (the value of the given learning outcome grade). Of necessity, these neurons do not have an activation function. The input layer was composed of 12 neurons (corresponding to the number of subject categories). The sequential model was based on dense layers - *keras.layers.Dense*. During the testing of this model, the most effective optimiser "rmsprop", the hidden neuron activation function "relu", the loss function – mean squared error (mse), and the metric - mean absolute error (mae) were established. The model was tuned experimentally in terms of the number of hidden layers and the number of neurons in these layers. The optimal version turned out to be the use of a fully connected neural network with five hidden dense layers (Figure 1) and training using mini-batch size = 100. During the training process, it was established that there is no need to use regularisation techniques.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 12)                156
_________________________________________________________________
dense_1 (Dense)              (None, 64)                832
_________________________________________________________________
dense_2 (Dense)              (None, 128)               8320
_________________________________________________________________
dense_3 (Dense)              (None, 256)               33024
_________________________________________________________________
dense_4 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_5 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_6 (Dense)              (None, 4)                 260
=================================================================
Total params: 83,744
Trainable params: 83,744
Non-trainable params: 0
```

Figure 1. Neural network model with Arch_1 architecture in its final version. The number of neurons in individual layers is shown.

Results of the final experiments are shown in Figure 2 and Figure 3.

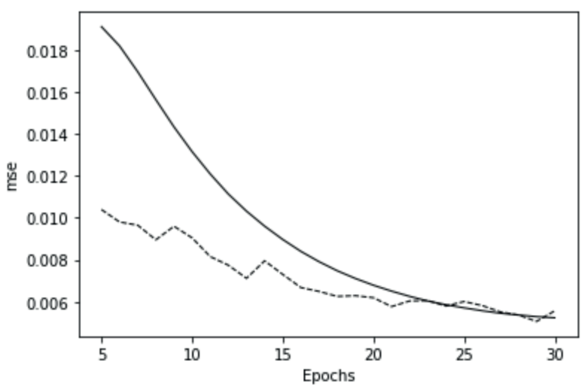

Figure 2. Loss function (mean squared error mse) for the sequential model. The broken line is the loss function validation.

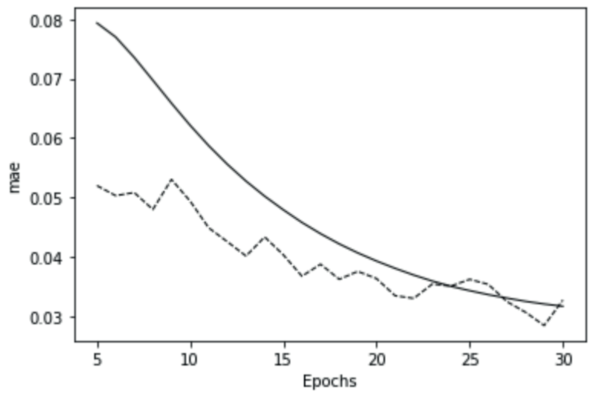

Figure 3. Metric (mean absolute error mae) for the sequential model. The broken line is the metric validation.

As shown above, the model mirrors the patterns trained very successfully. After 22-23 epochs, the learning starts to be unstable, and the first effects of overtraining can be observed. It appears that 22 epochs is the optimal training period for the network. The precision obtained then is very good *mse*=0,0075 and *mae* = 0,038.

The second model (Arch_2) is based on a multi-output network (in this case, 4 output), in which each output refers to a particular learning outcome (Figure 4). Parallel output branches are made up of sequential dense layers, and the output layers in each branch are made up of 11 neurons, which correspond to the possible grades of a given learning outcome (from 0 to 10). The final model – after many precision tests, contains a sequential component – 64 layers, 128, 256, 128 neurons. 4 parallel branches of the network are identical, which means equivalent grades for the individual learning outcomes and are made up of 64 and 32 neurons as appropriate. As a result of testing, the most effective optimiser was selected as *"adam",* the loss function = *categorical_crossentropy*, the metric *"accuracy"*, the loss function for the hidden layer neurons *"relu"* and for the output layers *"softmax"*.

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 12)]         0
__________________________________________________________________________________________________
dense_20 (Dense)                (None, 64)           832         input_2[0][0]
__________________________________________________________________________________________________
dense_21 (Dense)                (None, 128)          8320        dense_20[0][0]
__________________________________________________________________________________________________
dense_22 (Dense)                (None, 256)          33024       dense_21[0][0]
__________________________________________________________________________________________________
dense_23 (Dense)                (None, 128)          32896       dense_22[0][0]
__________________________________________________________________________________________________
dense_24 (Dense)                (None, 64)           8256        dense_23[0][0]
__________________________________________________________________________________________________
dense_25 (Dense)                (None, 64)           8256        dense_23[0][0]
__________________________________________________________________________________________________
dense_26 (Dense)                (None, 64)           8256        dense_23[0][0]
__________________________________________________________________________________________________
dense_27 (Dense)                (None, 64)           8256        dense_23[0][0]
__________________________________________________________________________________________________
dense_28 (Dense)                (None, 32)           2080        dense_24[0][0]
__________________________________________________________________________________________________
dense_29 (Dense)                (None, 32)           2080        dense_25[0][0]
__________________________________________________________________________________________________
dense_30 (Dense)                (None, 32)           2080        dense_26[0][0]
__________________________________________________________________________________________________
dense_31 (Dense)                (None, 32)           2080        dense_27[0][0]
__________________________________________________________________________________________________
Kw1 (Dense)                     (None, 11)           363         dense_28[0][0]
__________________________________________________________________________________________________
Ku1 (Dense)                     (None, 11)           363         dense_29[0][0]
__________________________________________________________________________________________________
Ku2 (Dense)                     (None, 11)           363         dense_30[0][0]
__________________________________________________________________________________________________
Kk (Dense)                      (None, 11)           363         dense_31[0][0]
==================================================================================================
Total params: 117,868
Trainable params: 117,868
Non-trainable params: 0
```

Figure 4. Functional model with four outputs corresponding to individual learning outcomes.

The loss function in the case of the functional model is a collective function, since the gradient descent algorithm necessitates the minimalization of the scalar value. Thus the values of the individual loss functions from the parallel outputs are joined into one – in the example above a simple sum with no weights has been used (Figure 5).

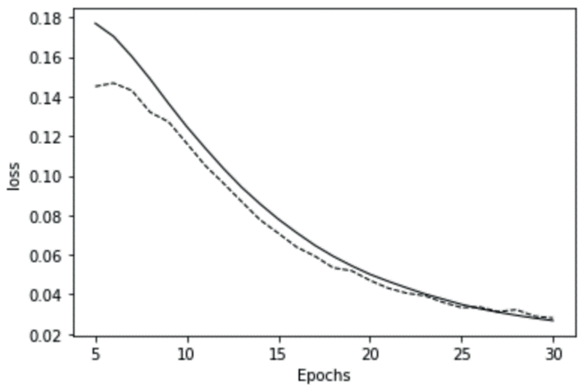

Figure 5. Loss function (categorical_crossentropy) for the Arch_2 model. The broken line is the loss function validation.

The metric (accuracy) for each of the parallel outputs is shown in turn on Figure 6 – Figure 9.

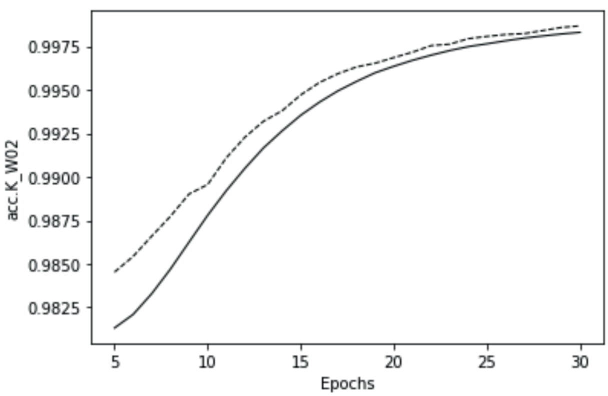

Figure 6. The course of the accuracy metric during training and validation (broken line) of the Arch_2 model for the output layer corresponding to learning outcome K_W02.

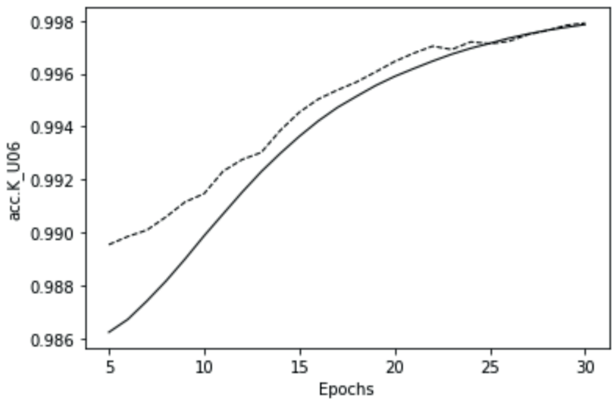

Figure 7. The course of the accuracy metric during training and validation (broken line) of the Arch_2 model for the output layer corresponding to learning outcome K_U06.

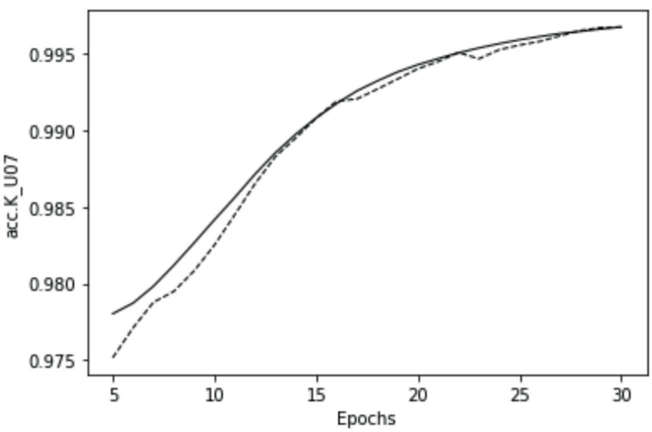

Figure 8. The course of the accuracy metric during training and validation (broken line) of the Arch_2 model for the output layer corresponding to learning outcome K_U07.

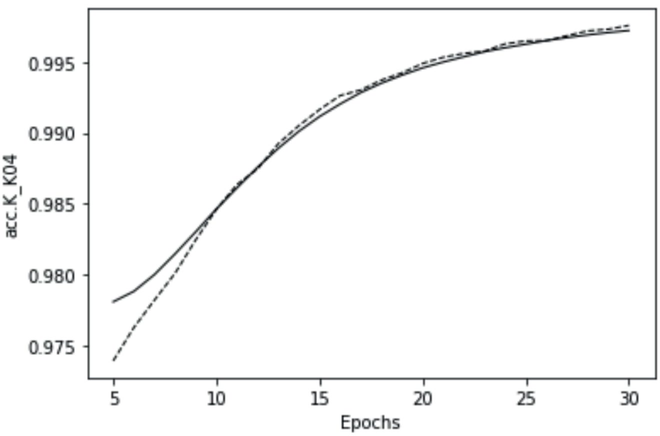

Figure 9. The course of the accuracy metric during training and validation (broken line) of the Arch_2 model for the output layer corresponding to learning outcome K_K04.

As can be seen, the functional multi-output model Arch_2, similarly to the sequential model, very successfully mirrors the training patterns. The level of precision is similar to that of the sequential model. The first effects of over-training can be observed after 24 epochs. The precision obtained then is very good, with validation *accuracy* 0,997 to 0,998 and loss function 0,04. Comparing both models, it can be stated that both can be used in the project. The only difference is a slightly longer training time for model 2. However, this is not a significant difference. Meanwhile, the functional model's architecture allows it to adjust the precision for each learning outcome separately. The more important and more significant learning outcomes can be tuned better by increasing the number of layers, or the number of neurons in the layers for those particular branches.

For both models, grade prediction based on the trained neural network was carried out at the testing stage for a randomly selected group of 40 students. As the network makes predictions based on the presentation of an individual grade vector, two methods of preparing the test data are possible. The first consists of duplicating the algorithm which was used for the training data, or in other words creating many input vectors with combinations of individual grades and estimating all the network's answers. The second involves preparing one input vector, for example by defining an average value for individual grades in particular categories and gaining one final answer. The second method was preliminarily chosen, based on the assumption that the multiplication of vectors is most suited to the network training stage.

During testing, the prediction results for each network model were compared with the grades proposed by the tutors. In the functional model, it was the winning neuron which decided on the final grade for a particular learning outcome. Since the output layer with *softmax* activation function creates the probability distribution for individual categories (grades), the winning category is taken to be the one which corresponds to the greatest probability value. In the sequential model, full conformity between the predictions and the tutor grades was obtained in 35 cases out of 40. In four cases, the prediction value differed from the tutor grade by one point in one learning outcome, and in one case, also by one point in two additional learning outcomes. In the functional method, a slightly better result was obtained, as only in four cases was a one-point difference noted for one learning outcome.

It should be remembered that during the training process, the neural network created and remembered patterns based on assessment methods developed by teachers, who also assessed the test cases.

5 Conclusion

The experiment demonstrated the possibility of using a deep neural network to predict learning outcome grades based on a standardized set of subject grades. This allows the process of defining the level to which learning outcomes have been achieved by a student to be automated. Suggested neural networks with two different architectures gained a very high level of precision in this area, despite an extremely small number of data (2056) being available during the training stage. The use of data augmentation allowed 560,688 relevant training vectors to be generated. Independent data in individual subject grade categories were used in this method. A middle-sized neural network can be trained using this method, which can effectively assess the level of learning outcome achievement. This creates the possibility of quickly and easily creating modules to automatically validate educational processes in education management systems in higher education institutions. It should be emphasized that the suggested data augmentation method can be used in many other areas, both in classification and regression tasks, as long as the data being processed are segmentally independent.

References

1. European Parliament and Council recommendation of 23rd April 2008 on creating European qualification frameworks for lifelong learning, government office file UE C111 6th May 2008
2. Kraśniewski A., *Proces boloński. To już 10 lat,* Fundacja Rozwoju Systemów Edukacji, Warszawa 2009

3. Chmielecka E., *Proces boloński i krajowe ramy kwalifikacji dla szkolnictwa wyższego,* Studia BAS, nr 3(35), 2013
4. 18th March 2011 *Act on amendments to the act: higher education law*, academic qualifications and titles and degrees and titles in art, and changes to certain acts.
5. *Referencing report. Relating the Polish qualification frameworks on lifelong learning to the European qualification frameworks*, 2013.
6. H. Aldowah, H. Al-Samarraie, and W. M. Fauzy, *Educational data mining and learning analytics for 21st century higher education: Are view and synthesis,* Telematics and Informatics, vol.37, pp. 13–49, 2019
7. A. Hernandez-Blanco, B. Herrera-Flores, D. Tomas and B. Navarro-Colorado, *A Systematic Review of Deep Learning Approaches to Educational Data Mining*, Complexity, vol. 2019, pp. 1-22,
8. https ://doi.org/10.1155/2019/1306039
9. M. Clément, C. Kurtz, L. Wendling, (2018) *Learning spatial relations and shapes for structural object description and scene recognition. Pattern Recognit 84* pp.197–210. https://doi.org/10.1016/J.PATCOG.2018.06.017
10. A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis (2018) *Deep Learning for Computer Vision: A Brief Review. Comput Intell Neurosci* 2018:1–13. https://doi.org/10.1155/2018/7068349
11. C. C. Park, Y. Kim, and G. Kim, *Retrieval of sentence sequences for an image stream via coherence recurrent convolutional networks,* IEEE transactions on pattern analysis and machine intelligence, vol. 40, pp. 945-957, 2018
12. G. Zeng, J. Zhou, X. Jia, et al (2018) *Hand-Crafted Feature Guided Deep Learning for Facial Expression Recognition.* In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, pp 423–430
13. D. W. Otter, J. R. Medina, and J. K. Kalita, 2018, *A Survey of the Usages of Deep Learning in Natural Language Processing.* 1, 1 (July 2018), 35
14. T. Young, D. Hazarika, S. Poria, E. Cambria, (2018). *Recent Trends in Deep Learning Based Natural Language Processing [Review Article].* IEEE Computational Intelligence Magazine. 13. 55-75.
15. H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Y. Chang, T. Sainath, *Deep Learning for Audio Signal Processing*, IEEE Journal of Selected Topics in Signal Processing, Vol: 13, Issue: 2 , May 2019
16. H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, *Deep Learning for Audio Signal Processing,* IEEE J. Sel. Topics Signal Process., vol. 13, no. 2, pp. 206–219, Apr. 2019
17. C. Romero and S. Ventura, *Educational data mining: A review of the state of the art,* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 40, no. 6, pp. 601–618, 2010
18. C. Romero and S. Ventura, *Data mining in education,* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 3, no. 1, pp. 12–27, 2013

19. A. Pena-Ayala, *Educational data mining: A survey and a data mining-based analysis of recent works,* Expert Systems with Applications, vol. 41, no. 4, pp. 1432–1462, 2014

20. N. Sharada, M. Shashi, and X. Xiong, *Modeling student knowledge retention using deep learning and random forests,* Journal of Engineering and Applied Sciences, vol. 13, no. 6, pp. 1347–1353, 2018

21. B. Kim, E. Vizitei, and V. Ganapathi, *Gritnet: Student performance prediction with deep learning,* 2018, https://arxiv.org/ abs/1804.07405

22. B. Akram, W. Min, E. N. Wiebe, B. W. Mott, K. Boyer, and J. C. Lester, *Improving stealth assessment in game-based learning with lstm-based analytics,* in Proceedings of the 11th International Conference on Educational Data Mining, 2018

23. B. Hu, *Teaching quality evaluation research based on neural network for university physical education,* In: IEEE, pp 290–293, 2017, https ://doi.org/10.1109/ICSGE A.2017.1

24. U. Kose, A. Arslan, *Optimization of self-learning in computer science engineering course: an intelligent software system supported by artificial neural network and vortez optimization algorithm,* Comput. Appl. Eng. Educ. 25, pp. 142–156, 2017

25. E. T. Lau, L. Sun, Q. Yang, *Modelling, prediction and classification of student academic performance using artificial neural networks,* SN Applied Sciences, 1:982, 2019, https://doi.org/10.1007/s42452-019-0884-7

26. S. Zhao, Y. Zhang, X. Xiong, A. Botelho, and N. Heffernan, *A memory-augmented neural model for automated grading,* in Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale, pp. 189–192, ACM, Cambridge, Massachusetts, USA, April 2017

27. J. G. Alvarado, H. A. Ghavidel, A. Zouaq, J. Jovanovic, and J. McDonald, *A comparison of features for the automatic labeling of student answers to open-ended questions,* in Proceedings of the 11th International Conference on Educational Data Mining, 2018

28. A. Sales, A. Botelho, T. Patikorn, and N. T. Heffernan, *Using big data to sharpen design-based inference in A/B tests,* in Proceedings of the 11thInternational Conference on Educational Data Mining, 2018

29. K. Taghipour and H. T. Ng, *A Neural Approach to Automated Essay Scoring,* in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1882– 1891, Association for Computational Linguistics, Austin, Texas, November 2016

30. Y. Zhang, R.S hah, and M. Chi, *Deep learning + student modeling + clustering: a recipe fo reffective automatic short answer grading,* in Proceedings of the 9th International Conference on Educational Data Mining, pp. 562–567, 2016

31. H. Choi, Z. Wang, C. Brooks, K. Collins-Thompson, B. G. Reed, and D. Fitch, *Social work in the classroom? A tool to evaluate topical relevance in student writing,* in Proceedings of the 10th International Conference on Educational Data Mining, 2017

32. Neethu George, Sijimol PJ, Surekha Mariam Varghese, *Grading Descriptive Answer Scripts Using Deep Learning,* International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 5, pp. 991-996, 2019
33. S. Tang, J. C. Peterson, and Z. A. Pardos, *Deep neuraln etworks and how they apply to sequential education data,* in Proceedings of the Third ACM Conference on Learning @ Scale (L@S'16), pp . 321–324, ACM, New York, USA, April 2016
34. F. Okubo, T. Yamashita, A. Shimada, and H. Ogata, *A neural network approach for students' performance prediction,* in Proceedings of the Seventh International Learning Analytics & Knowledge Conference (LAK'17), pp. 598-599, ACM., New York, USA, March 2017
35. L. Wang, A. Sy, L. Liu, and C. Piech, *Learning to represent student knowledge on programming exercises using deep learning,* in Proceedings of the 10th International Conference on Educational Data Mining, pp. 201–204, Cambridge, Mass, USA, April 2017

DE GRUYTER OPEN

JACSM 2017, Vol. 9, No. 2, pp. 23 - 34

10.1515/jacsm-2017-0007

USING THE MVC PATTERN IN THE DESIGN OF COMPUTER GAMES

Patrycja Pankiewicz

Information Technology Institute,
University of Social Sciences, Lodz, Poland
ppankiewicz@spoleczna.pl

Abstract

Designing computer games is a process of forming their principles and plot. It starts with the development of a general concept, until creation of detailed and complete technical documentation to enable the implementation of the game. Application of the MVC pattern in the design of computer games facilitates the whole process by organizing the game structure, where each layer (Model, View, Controller) is responsible for a specific task. Such division increases the legibility of the code, especially when creating large applications, and its physical segregation allows you to avoid problems with finding a specific functionality.

Keywords: MVC Pattern, Model-View-Controller, design pattern, game engine

1 Introduction

Designing computer games is the creation of their scenario and binding rules. This process leads to the general concept of developing precise and complete technical documentation that will allow the implementation of the game. This paper presents an analysis of the possibilities of using selected computer game programming environments in the aspect of the MVC pattern.

Any computer game development can be considered from the point of view of at least two aspects: View (graphics of the game) and algorithmics (implementation of the planned scenario). This leads directly to the use of the MVC design model (Model-View-Controller). The MVC pattern is now more often used in the design of computer games. Separation of the three layers ensures free access to modification of the code without its unnecessary change in other places. Such solution is extremely beneficial during the team work, where one group is responsible for the game's appearance, the next one for its physics, and another for user interaction programming. However, the pattern must be well thought over. The game design is the key to success. On its basis, appropriate tasks and functions are assigned to each of the layers. However, not all environments are well adapted to implement this pattern; it is worth to get acquainted with available technologies and capabilities of particular environments before working on the game's project and software.

The paper will present a method of comparing the effectiveness of two selected environments for the design of computer games (Godot, Unity) in the aspect of the MVC pattern.

2 Features of the MVC Pattern

The most popular pattern currently used in probably every technology used to implement web applications is the MVC pattern. Its three basic components are:

Model - is responsible for business logic, this term is related to the functionality of the way in which the application stores data. It should be treated as the only part of the application for data storage in a permanent way. It is important that implementation details regarding the way the data is stored by the model were hidden from the rest of the application. The method of storing information is arbitrary, but most often it is a database. The appropriate way to implement the model is to build a separate class for each logical object.

View - is responsible for the presentation, displays the data to the user. The most common output format for web applications is HTML. The view uses the model to get the data to display. It should create instances of the model classes and call those methods that are responsible for getting the right data. It should not change the state of the application. If it is intended to display a list of users, it cannot delete them. The view is mainly a code that draws the appropriate data from the model.

Controller - this is the most important of layers. It handles the input, where information is received directly from the user. Controls the model and view and exchanges information with them. Decides which operations should be carried out and which view to display. It is an essential part of every library implementing MVC [6, 7, 8].

The MVC pattern is best suited for large and complex applications, where many people are involved, who should be designated competence areas.

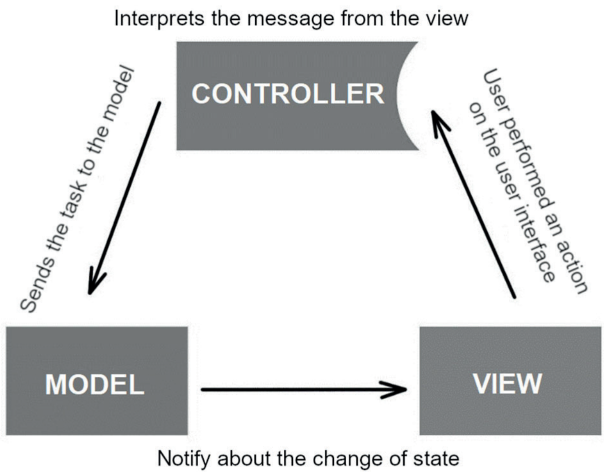

Figure 1. Operation scheme of the MVC pattern.

3 Environments of Game Design and Implementation

Godot is a MIT-based license game engine, one of the simplest and most liberal open source licenses. The program allows creating 2D and 3D games. Godot works with Windows, Linux, OS X and BSD systems. This environment allows you to create games for consoles and for PC and mobile platforms. Thanks to it, it is also possible to set texture resolution and compression for all platforms that support the game. The name Godot is related to art, called "Waiting for Godot", written by Samuel Beckett due to the chronic need to add new elements in the environment that could lead to the creation of an exhausting product, but this does not happen [5].

The current stable version of the Godot 3.1.1 and it is available for downloading at godotengine.org. This engine allows developers to create a game from scratch without the need to use separate tools apart from those used to create the content. Using the Godot engine, you can create games in C #, C ++ or GDScript scripting language. The latter is a high level programming language similar to Python. Godot includes a built-in physical engine for 2D and 3D aspects of the engine that supports collision detection. After unpacking, the program takes only 41 MB of disk space. Games that were created using the Godot engine are e.g. Running Nose, Anthill, or Tank of Freedom [4, 5, 10].

Another game engine for the design and implementation of games is Unity. It is used by a wide spectrum of users, starting with hobbyists and ending with large companies. It enables creating games and interactive applications for web browsers, portable devices, personal computers or consoles. Created games can be run on Microsoft Windows, macOS and Linux. The games that were created using the Unity engine include: Assassin's Creed Identity, Deus Ex: The Fall, or Wasteland 2 [3, 9].

Unity is a very flexible and complete game engine. It allows to go through all the stages of building the game. Mechanisms in which the engine is equipped provide a wide range of possibilities. The wizard contains many intuitive tools to design the game interface, advanced options for animating and shading models based on physical simulations. For the first time, Unity was presented in 2005 at the Apple's Worldwide Developers Conference. The program also allows you to use the so-called Asset Store. It is a resource store in which the user can use paid or free components. There are, among others, models for 2D and 3D games, various types of plugins, textures, ready surroundings or levels. Unity's environment has been designed so that it can be used not only by large studios, but also by independent developers. It allows you to prepare the game for distribution in platforms such as the Microsoft Store, PlayStation Network or Steam [3]. The current stable version of the program is Unity 2019.1.6 and it is available for downloading at https://unity3d.com [2].

4 Integration of the MVC Pattern With Selected Game Design Environments

Game's structure in both environments was designed on the basis of the MVC design pattern, which divides it into three independent layers, thanks to which each logical part is separated from each other, and by changing in one place, there is no need to make more changes in other parts.

The model is a computer representation of the problem under consideration It is a pure simulation of the world and does not know anything about the data entered by the user. The model contains objects used to perform all operations that are related to the implementation of the application's functionality It is a set of rules in the game world. In both games the model will be responsible for the movement of physics. The created functions will later be available from the Controller level.

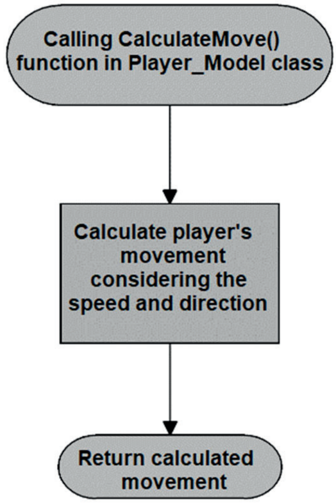

Figure 2. Scheme of an example function in the Model script – Godot engine (own execution).

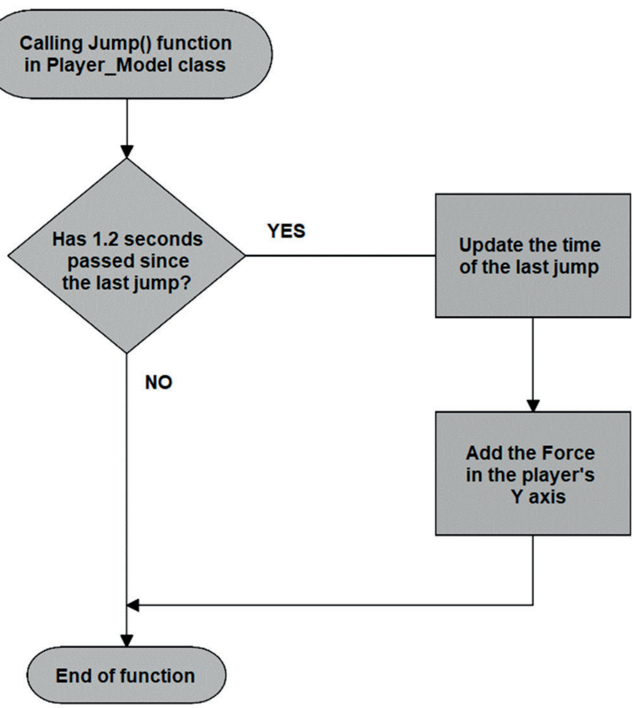

Figure 3. Scheme of an example function in Model script – Unity engine (own execution).

The view is a layer of presentation of data displayed to the user within the graphical interface. It is separated from logic, thanks to which it allows for non-invasive graphic changes at any time In designing and creating games, it is responsible for visualization, i.e. color, texture, or animation. In both games, the role of the view will be to allow the player's color to be changed by modifying the material superimposed on the rendered object's mesh.

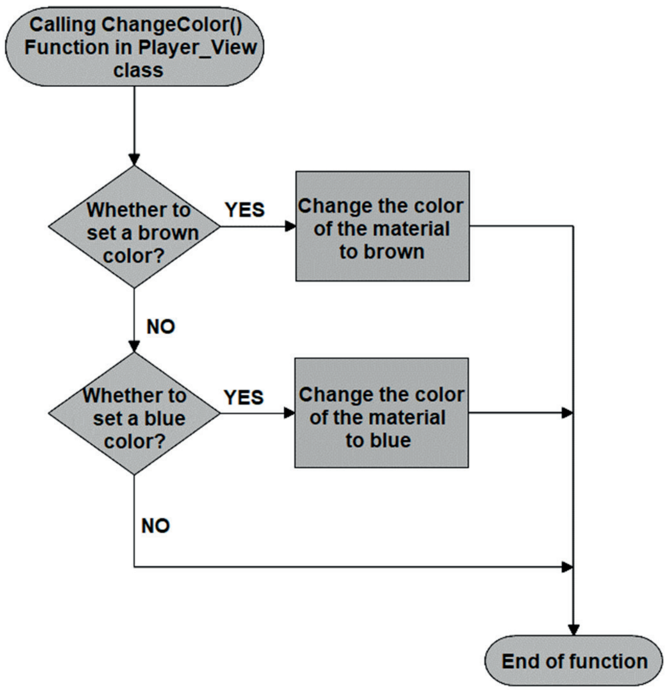

Figure 4. Scheme of an example function in View script – Godot engine (own execution).

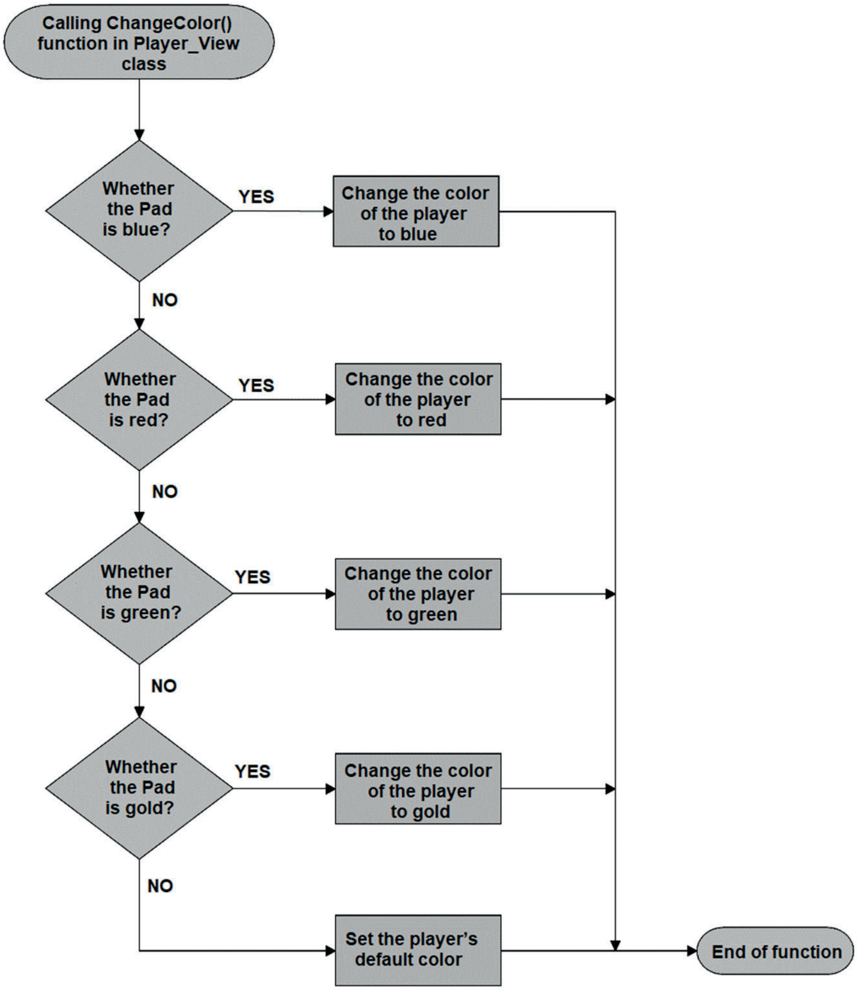

Figure 5. Scheme of an example function in View script – Unity engine (own execution).

The controller is responsible for the processing, analysis and receiving of user input data. Refreshes the view, changes the state of the model, and also switches control to another controller. The application may have several controllers, but at the moment only one of them can control the application [1].

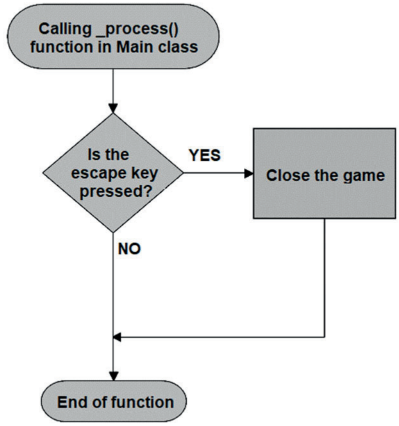

Figure 6. Scheme of an example function in Controller script – Godot engine (own execution).

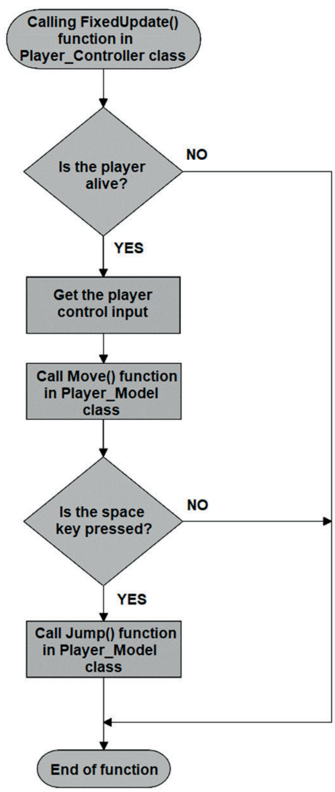

Figure 7. Scheme of an example function in Controller script – Unity engine (own execution).

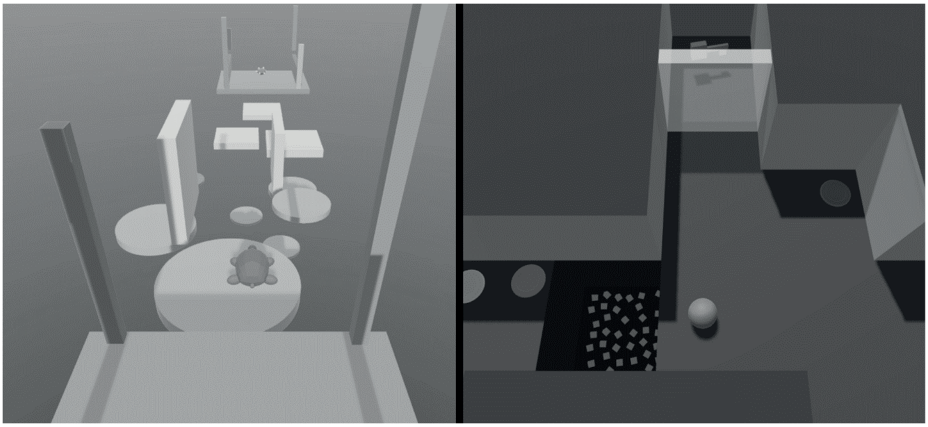

Figure 8. Comparison of frames from games in game environments: Godot (left) and Unity (right) (own execution).

5 Method of Comparing Game Environments

Unity and Godot environments have a number of functionalities that facilitate work on the design of computer games. Their interfaces have been designed so that the user has easy access to the most important tools. In the case of Unity, individual panels can be adjusted in any way by dragging and dropping them to a specific location. Unity also has a preview of the game in real mode and the search engine of objects on the scene. These elements are not found in the Godot engine. However, it has other functions, such as: built-in editor for writing scripts in GDScript language and the ability to change the interface language. Godot also offers visual programming that allows you to create games by arranging blocks representing specific variables and functions.

Materials that have been applied to the view layer are much simpler to add in the Unity environment. In the game design window, by means of the right mouse button one should choose the following options: Create and Material, and then assign the appropriate color to the material in the Inspector tab. Such prepared component can be simply dragged onto any object on the scene to give it the right color. If material is modified on all objects to which it has been assigned, it will be automatically updated. Adding materials in the Godot environment is not possible through the use of drag & drop technology. The material can be assigned and modified only from the level of creation of the object, so when changing the hue, you should go to the editing of the given object and adjust its color in its properties.

Programming of the controller layer in both environments was similar due to the built-in functions FixedUpdate, Start and Update in the Unity environment, which correspond to the functions _physics_process, _ready and _process in the Godot environment. The FixedUpdate and _physics_process functions are independent of the number of frames per second and are used in physics-related calculations. Start and _ready are always performed at the beginning before the first frame is updated, when the object is created on the scene. Update and _process are functions that are always performed before the rendered frame of the game.

The table below provides a detailed list of the functionalities of both environments.

Table 1. Comparison of Godot and Unity engines part 1 (own execution).

	Godot	Unity
Platforms		
Windows	x	x
Linux	x	x
Mac	x	x
iOS	x	x
Android	x	x
Xbox One	-	x
PlayStation 4	-	x
Technologies		
2D games	x	x
3D games	x	x
VR games	x	x
Properties of the game engine		
Simulation in real time	-	x
Physics	x	x
Particle system	x	x
Supported languages	GDscript, C#, C++	C#
Visual programming	x	-
Usage of MVC pattern	Difficult	Full
User interface		
Supports drag&drop	-	x
Built-in language editor	x	-
User interface customization	Templates	Freely

Table 2. Comparison of Godot and Unity engines part 2 (own execution).

	Godot	Unity
Import of 3D objects		
.3ds	-	x
.blend	-	x
.obj	x	x
.fbx	-	x
.dae	x	x
Audio		
.mp3	-	x
.ogg	x	x
.wav	x	x
Other		
Licence	Free (MIT)	Free
Size	41 MB	3.86 GB
Monetization of created game	x	x

6 Conclusions

The possibility of applying the MVC pattern in Godot and Unity environments differs significantly from one to the other. In the case of Unity, each of the written scripts has free access to other scripts attached to the object or its parameters. It is enough to enter only the name of the GetComponent function, which is the same as the class in it. In Godot environment, it's much harder to access from one script to another because it has a node structure. Without a thorough knowledge of the game engines documentation, the user is not able to implement the MVC pattern in the game design. Finding the right reference to the script takes much more time in this case.

The MVC pattern in a simple way allows to expand projects with new functionalities by independently developing and updating separated components. However, improper use of the pattern can lead to creating unnecessary amount of code. The project will begin to grow, and breaking it down into smaller and smaller parts will be difficult to modify. Therefore, it is necessary to think over the implementation of the MVC pattern in your project, as well as to familiarize yourself with the possibilities of individual environments. It may turn out that the given environment is not well adapted for implementation of this pattern.

References

1. CodeWithChris, *How To Build a Match Game - Lesson 2 (MVC Design Pattern).* May 2019, https://www.youtube.com/watch?v=j8nmtC4NWdA
2. ElHady H., *Top Game Engines.* May 2019, https://instabug.com/blog/game-engines/
3. Geig M., *Unity. Przewodnik projektanta gier.* Helion, 2015
4. Kowasz K., *Godot — nowy silnik do tworzenia gier udostępniony wraz z kodem źródłowym*. May 2019, https://www.dobreprogramy.pl/Godot-nowy-silnik-do-tworzenia-gier-udostepniony-wraz-z-kodem-zrodlowym,News,52172.html
5. Linietsky J., Manzur A., Godot community, *Godot Docs – 3.1 branch*. May 2019, https://docs.godotengine.org/pl/latest/getting_started/scripting/gdscript/gdscript_basics.html
6. Mallawaarachchi V., *Wzorce architektoniczne w pigułce.* May 2019, https://bulldogjob.pl/news/422-wzorce-architektoniczne-w-pigulce
7. Piechowiak J., *MVC — wprowadzenie do wzorca*. May 2019, https://blog.helion.pl/mvc-wprowadzenie-wzorca/
8. Rak T., Żydzik K., *C# 6.0 i MVC 5. Tworzenie nowoczesnych portali internetowych.* Helion, 2015
9. Ross E., Ross J., *Unity i C#. Podstawy programowania gier*. Helion, 2018
10. Tyser, *Tutorial przedstawiający składnię gdscripta.* May 2019, http://troman.pl/news/godot-podstawy-gdscripta_ID98

DE GRUYTER
OPEN

JACSM 2017, Vol. 9, No. 2, pp. 35 - 58
10.1515/jacsm-2017-0008

RULES OF KNOWLEDGE BASES OF IT SYSTEMS SUPPORTING EDUCATION QUALITY MANAGEMENT IN UNIVERSITIES

Krzysztof Przybyszewski, Andrzej Cader

Information Technology Institute,
University of Social Sciences, Lodz, Poland
kprzybyszewski@san.edu.pl, acader@san.edu.pl

Abstract

The study presents the assumptions of the project of an IT system supporting the quality management of education at the University. The specification and characteristics of the basic elements of the system as well as the basic rules of the system knowledge base are described. Using the given rules, exemplary calculations of the levels of achieved learning outcomes were made. The MinervaEDU system, in which the described rules were used, was also presented.

Keywords: IT system, quality of education, quality management, knowledge base, expert systems

1 Introduction

The overriding goal of any educational system is to provide the participant with appropriate competences (in terms of: knowledge, skills and social competences) needed to participate in the next stage of activity [9-13]. These can be: next stages of education (e.g. schools of any level, qualification or vocational courses) or work requiring specific qualifications.

It is very important to define the estimators (parameters) that allow a participant in the education process to be issued with an appropriate certificate confirming the acquisition of a set of competences, i.e. a certificate confirming the obtained qualifications. Examples of such certificates are: school certificate, secondary school-leaving examination certificate, graduation diploma with an appropriate supplement, master's diploma, driving licenses, etc.

In some certificates, values that define the degree to which a given person has achieved the previously assumed learning outcomes are given, and in each case a minimum value is provided for the issuance of a certificate. The values are given in various representations: descriptive (linguistic), graded scale, relative scales (the most popular is the stanine or standard five scale) or other [11-13].

Specification of competences achieved in a specific educational process and related with them and the values of parameters (estimators) allow to determine the didactic profile of a given educated or graduated person.

Taking into account the purpose of the conducted education process, it should be assumed that the achieved learning outcomes specified in the didactic profile should be consistent with the requirements for graduates in terms of participation in the next stage of activity. It is simple: the values of the didactic profile parameters are direct indicators of the student's suitability for this stage of education and, in this sense, they can be used for selection during enrollment.

The situation is completely different in the case of a graduate taking up employment after completing a certain stage of education. In this case, the presentation of the didactic profile may not be sufficient: on their basis, the future employer should be able to determine the suitability of the applicant for the offered job position. For this purpose, it is necessary to define the competency profile of the offered job position, i.e. a set of competences required to take up a job. It would also be advisable to specify the level of required competencies, although this is not necessary.

Taking into account everything that has been presented earlier, it seems obvious to define the process of managing the quality of education in any educational unit (system) as adjusting the assumed didactic profiles (or actually education programs) to the potential competency profiles of professions (jobs) or the required competency profiles for the next stages of education. The adjustment process should be continuous and measurable.

When we define the education quality management process in such a way, we encounter a number of difficulties related mainly to the lack of standards and the lack of dictionaries common to all educational units and employers that define competences (both in the didactic profile and in the competency profile).

In the case of undertaking education at the next stages of education, the problems have been largely resolved by the Ministry of National Education by [10, 16, 17] by:

1. Introduction of uniform and externally assessed competency tests and examinations at all stages of education, including upper secondary schools.

2. The application of standards with regard to determining the learning outcomes by the use of the national competence framework in the form of: core curriculum for education, result programs and education standards (tests and examinations).

In the case of vocational education and lifelong learning, the activities leading to the reduction of the described problems consisted mainly in:

1. Development of National Professional Qualification Standards (NPQS) [1, 5-8]. The list of standard descriptions developed so far can be found on the website of the Ministry of Labor and Social Policy: www.kwalifikacje.praca.gov.pl.

2. Determining the binding rules for building a framework for vocational lifelong learning [8, 9].

In 2012, the Ministry of Science and Higher Education introduced a normative [20] National Qualifications Framework (NQF) for higher education standardizing (or actually defining) competences in the didactic profile by defining the area-based learning outcomes for 9 areas of education. The implementation of the Qualifications Framework in the Polish education system is an international obligation that is result of the implementation of the Bologna Declaration, the provisions of the Bergen Conference of 2005, attended by ministers of higher education from 46 countries, and result of the Recommendation of the European Parliament and the Council of 23 April 2008 [25]. The Ordinance of the Minister of Science and Higher Education also specifies the principles of building the Domain Learning Outcomes - DLO (and even several model sets were given), which are to constitute the basis for the description of the student / graduate of first and second cycle studies. The publications [4] contain guidelines on how to build higher education programs in order to comply with the assumptions of the NQF. Unfortunately, so far it has not been specified how to take into account the requirements of the labor market (e.g. specified in the NPQS) in the construction of the DLO (i.e. the didactic profile) or what transformation should be applied between the DLO and the requirements of the competency profile of professions (e.g. specific in NPQS).

The above-mentioned Regulation was supplemented by the Regulation of the Ministry of Science and Higher Education of 30 September 2016 on Polish Qualification Framework (PQF) second-cycle characteristics typical for qualifications obtained in higher education after obtaining qualifications at level 4 - levels 6-8 [21].

After such a short description of the conditions and definitions of the management of the educational process, we can begin to try to define the assumptions of the IT model supporting the system and the role of the rule base in this system. Due to the type of our work and interests, it is obvious that the considerations will concern the system supporting the quality management of education in universities.

In line with the previously proposed definition of the education quality management process in any educational unit, it will take into account appropriately defined elements (student / graduate didactic profile, curriculum for a given field of study or level of education, competency profile of the profession / job position) and the rules for matching these elements. The matching rules will be the estimators used in managing the quality of education at the university.

Because in the didactic profile, the acquired competences are described with the language of learning outcomes (they are given as learning outcomes - LO), and in the competency profile of the profession / job position they are described in the language of professional competences (they are given as professional competences), the description languages for both profiles differ significantly, it seems necessary to define an intermediate competency profile into which the parameters of both profiles will be transformed and which will be used in the profile matching optimization process.

It seems to us that the most appropriate will be the application of a systemic approach based on the analysis of the education system and a set of potential job positions treated as an information systems [15, 22].

2 Student's (Graduate's) Didactic Profile

According to the definition of the information system, in the case of universities, the set of information system objects is a set of students and graduates or a set of subjects / modules, the set of features characterizing the objects is the set of achieved learning outcomes, and the set of values that can be assigned to the features is the set of possible grades expressed: in a numerical scale adopted in a given university or a percentage scale or a two-state scale (pass / not pass), or a descriptive scale (word descriptors). The information function is the rule for assigning grades to subjects / modules, i.e. a set of rules for grading subjects specified in each card (description, syllabus) of the subject (module).

A graphic illustration of the definition of the education system as an information system along with a representation of an exemplary object (Student X) is shown in Figure 1.

In the case of universities, grades are obtained by the student when completing a subject (module) or after passing the exam in this subject (module). Grades may be assigned to each form of activity or a total grade may be given. Each item can be treated as an information subsystem. In this case, the set of objects is a set of subjects, the collection of features is a collection of competences (Subject Learning Outcomes - SLO), and the set of values is a set of partial grades of a given subject most often assigned to specific SLO.

On the basis of the grades from the modules ($grade_i$) and the matrix of filling a given Field of Study Learning Outcomes (FSLO) by subjects, it is possible to determine the level of achievement of a given FSLO (k_m). The algorithm for converting subject grades into values characterizing the levels of fulfilment of a given educational effect has already been described in the publication of our co-authorship [14]. The scheme of determining the values that define the levels of FSLO achieved by the student is shown in Figure 2.

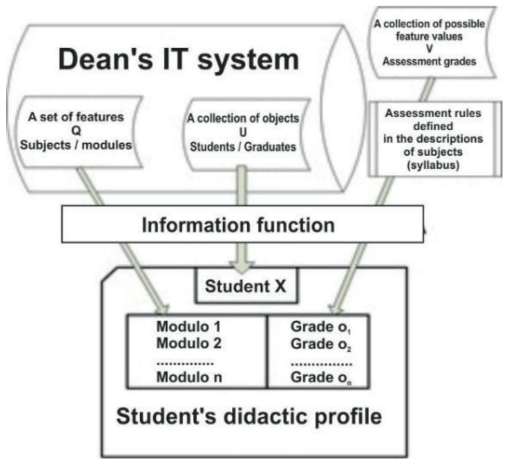

Figure 1. Graphic illustration of the definition of the education system as an information system for a set of values in the form of assessments

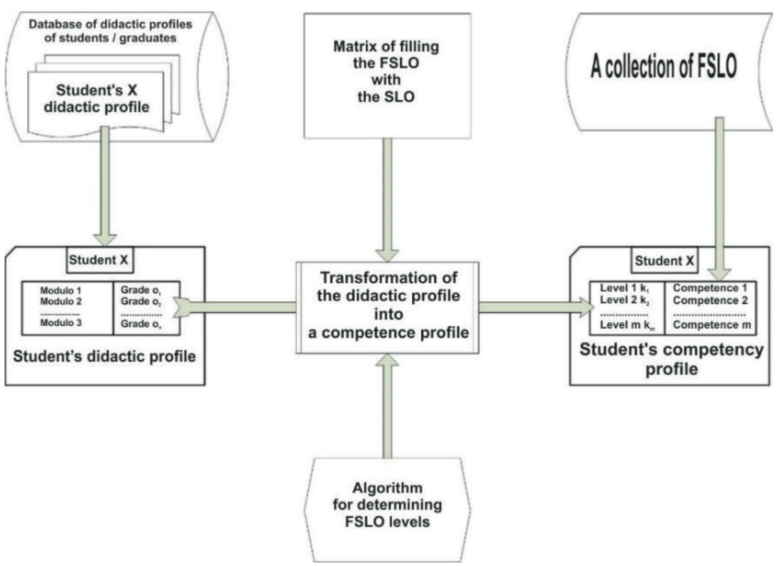

Figure 2. Scheme for determining the values that define the FSLO levels achieved

In the same way, taking into account: grades in subjects (modules), the matrix of filling the field of study learning outcomes (FSLO) by modules and the matrix of filling the Domain Learning Outcomes (DLO) with the Field of Study Learning Outcomes, it is possible to determine the levels of filling each of the Domain Learning Outcomes (dkm). The scheme of determining the values that define the

levels of DLO achieved by the student is shown in Figure 3.

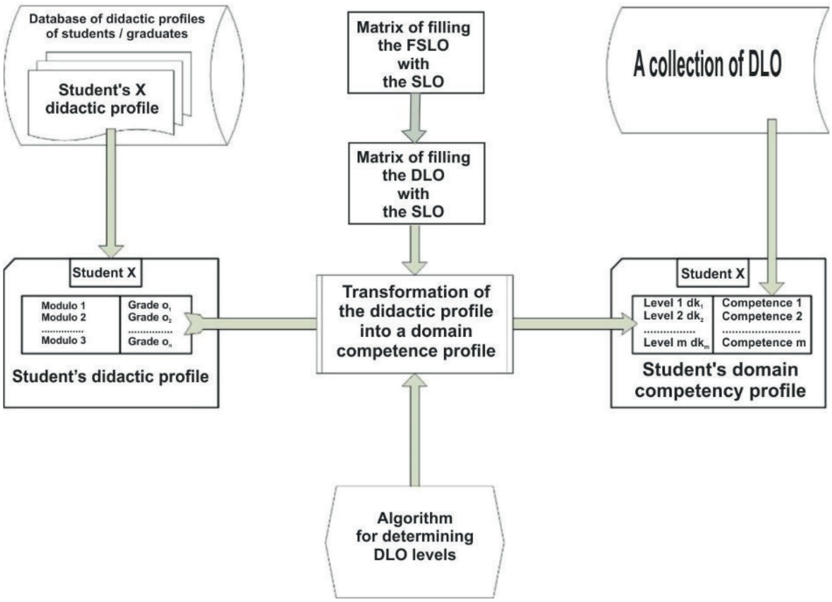

Figure 3. Scheme for determining the values that define the achieved levels of a domain learning outcomes

In the case of the old fields of study standards in force before the implementation of the NQF, the set of competences defining the qualifications (didactic profile) of a student / graduate was synonymous with the set of subjects implemented in a given field and level of education.

It should be noted that for each competence, a minimum value characterizing the level of its fulfilment must be defined, which determines the possibility of including the competence as a component of the didactic profile. Obtaining a certificate confirming a graduate's qualifications is possible only after passing all the competences that are components of a given qualification.

It should also be noted that the student's didactic profile constructed in this way is dynamic, which means that each competency is assigned successive values that change during the studies. Only the graduate has a full set of competences with the values characterizing the levels of their completion, and the diploma of graduation with the attached supplement is just a confirmation of the acquired qualifications.

In the case of the didactic profile, it is desirable to combine the IT system supporting the quality of education management with the dean's office system. This allows for a quick import of large amounts of data and speeds up the process of counting the values of levels and the process of optimizing the entire system itself.

3 Elements supplementing the student's (graduate's) didactic profile

In the case of the education quality management system, there is a need to expand the student's (graduate's) didactic profile with additional elements. These elements (or rather the values placed in their structure) are required to verify the correctness of the structure of study programs, because it is difficult to make such verification based on single values.

The basic element supplementing the basic didactic profile is an alternative didactic profile constructed on the basis of the results of student / graduate self-assessment questionnaires. In our opinion, the questionnaires should be constructed in such a way that their results allow for the construction of an additional set of values assigned to the features (competences) from the basic profile. For example, the questionnaires should include questions about the subjective assessment of competences acquired during the course of studies in particular subjects. Gathering such a set of information allows the construction of an alternative, subjective didactic profile with the same feature specification. A diagram of the construction of an alternative didactic profile is shown in Figure 4.

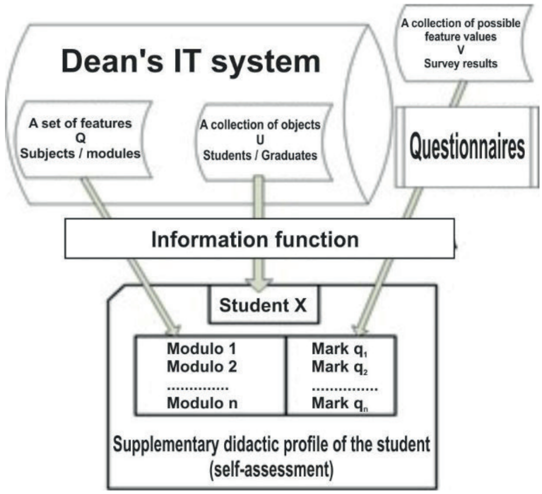

Figure 4. Diagram of the construction of an alternative didactic profile

The described alternative didactic profile has another very important advantage: with properly defined access rights for students, it enables them to independently control their own learning path and adapt it to their own needs and demand on the labor market (subject to the availability of job offers analyzes).

Another complementary element should be a set of results of surveys conducted among employers employing graduates or students. The results of these surveys should enable the construction of a second alternative learning profile. As in the case of the first alternative, it is advisable that also in this case the profile should be based on the same set of features (competences) as the main profile. A diagram of the construction of an alternative didactic profile based on the results of surveys conducted among employers is shown in Figure 5.

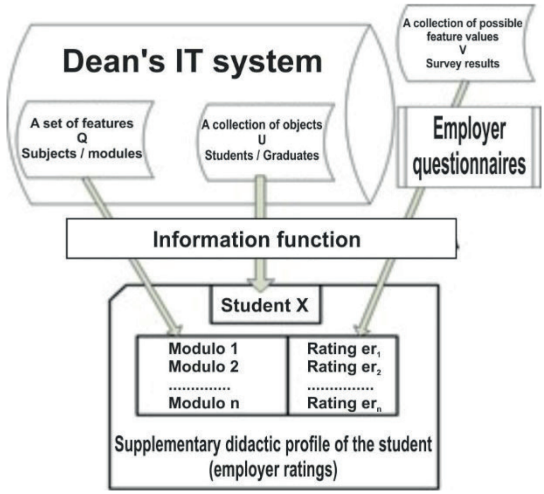

Figure 5. A diagram of the construction of an alternative didactic profile based on the results of surveys conducted among employers

It is also possible to construct a supplementary didactic profile of virtual students by surveying potential candidates for studies and courses. In this case, they should assess not their own skills (they do not have any yet), but the degree of suitability of a given competence to achieve their own goal, which they would like to achieve at the end of the training cycle.

4 Competency profile of the profession (job positions)

In the case of potential job positions (professions) for which students / graduates can apply, the set of system objects is a set of positions, the set of features is a set of competences, and the set of possible values characterizing the features is a set of values that define the level of competences expressed: in a percentage or numerical scale , a two-state scale (with / without), or a descriptive scale (verbal descriptors). The information function is the rule of assigning the expected values that define the levels of competences. A graphic illustration of the definition of a system of potential professions as an information system is presented in Figure 6.

Examples of descriptions of professions consistent with the ones adopted by us can be found in the National Standards of Vocational Qualifications (NSVQ), on the website of the Ministry of Labor and Social Policy (ML&SP) [3]. When defining the assumptions of the IT system model supporting education quality management, in terms of the structure of data on vocational qualifications, we took into account the principles of creating descriptions of professions provided in studies related to the NSVQ [5, 7].

The database of professions for our system should be built on the basis of the above-mentioned professional qualification standards, but should also take into account job offers submitted by employers. In this case, most often they have a structure consistent with the one adopted by us, but a two-state scale is used - employers most often mention the desired competences, without specifying the value characterizing the degree of their achievement. An example of such a job offer for an IT specialist as a software tester is shown in Figure 7.

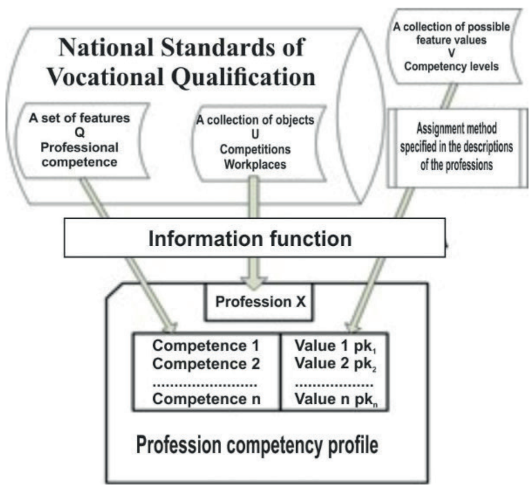

Figure 6. Graphic illustration of the definition of the system of potential professions as an information system for a set of values in numerical form


ERICPOL i-EVOLUTION

UNIVERSUM TOP 30 IDEAL EMPLOYER

Ericpol projektuje, koduje, testuje, wdraża i integruje. Od 23 lat tworzymy kompleksowe i niezawodne rozwiązania telekomunikacyjne, rozwiązania dedykowane oraz świadczymy usługi doradcze dla sektorów telekomunikacji, M2M, służby zdrowia oraz dla przedsiębiorstw opierających swoją działalność na nowoczesnych technologiach. Posiadamy główne lokalizacje w Łodzi, Krakowie i Warszawie, a także spółki zależne na Ukrainie, Białorusi i w Szwecji. Jako Grupa Ericpol zatrudniamy obecnie ponad 1800 pracowników. Ze względu na dynamiczny rozwój, poszukujemy kandydatów na stanowisko:

Software Tester/Developer

Miejsce pracy: Łódź

Osoba zatrudniona na tym stanowisku będzie odpowiedzialna za:

- tworzenie i rozwój oprogramowania,
- tworzenie, weryfikację poprawności i rozwój narzędzi do produkcji oprogramowania,
- weryfikację poprawności dostarczanego kodu,
- tworzenie dokumentacji technicznej,
- tworzenie/edycję scenariuszy testowych w oparciu o dokumentację techniczną,
- realizację scenariuszy testowych w środowisku żywym, jak i symulowanym,
- raportowanie błędów oraz współpraca z programistami w celu ich rozwiązania.

Poszukujemy osób:

- posiadających umiejętności analitycznego myślenia, szybkiego uczenia się oraz pracy w zespole,
- otwartych na pracę w metodykach Agile,
- ze znajomością minimum jednego języka programowania: Java, Perl, Python, C/C++,
- zainteresowanych rozwojem w zakresie telekomunikacji, programowania i testowania oprogramowania,
- posiadających doświadczenie w programowaniu/testowaniu,
- ze znajomością języka angielskiego na poziomie umożliwiającym swobodną komunikację w mowie i piśmie.

Podziel się opinią

Figure 7. Job offers for an IT specialist as a software tester (source: http://www.pracuj.pl/praca/software-tester-developer-lodz,oferta,3375970)

It would be advisable for the vocational competency profile to be based on the same competence specification as the basic didactic profile (or vice versa: the competence specification of the didactic profile should be based on the occupational competence specification). Unfortunately, in many cases such a similarity does not take place due to different principles of the structure of these profiles (for example, lack of communication between the Ministries), which results in differences in the database dictionaries. This is one of the problems that was mentioned in the introduction.

5 Indirect competence profile

As already mentioned, the characteristics of the characteristics for the learning profiles and the competency profile of the profession (workplace) are usually different, which is natural and is the result of adapting them to different fields of application: the didactic profile is used in education in relation to graduates, students, pupils, etc., and the professional profile is used in units employing employees. We propose, for the purposes of the IT system supporting the quality management of education at the University, the construction of an additional, intermediate competency profile (a common competence dictionary) into which both the characteristics of the didactic profile and the competency profile of the profession will be transformed.

The indirect competency profile is not an information system, but only used both in learning and professional profiles (workplaces). An example of such a set of competences may be a set of competences characteristic of the general profession: computer scientist (or: surveyor, mechanic, psychologist, historian, art historian, etc.). For this set of competences, it is necessary to construct a transformation matrix both from / to the didactic profile and from / to the competency profile of each profession. Most often, in the case of a professional profile (workplace), not the entire set of competences will be exhausted, the features of this profile will only constitute a subset of the set. A diagram of the construction of an intermediate competency profile is shown in Figure 8.

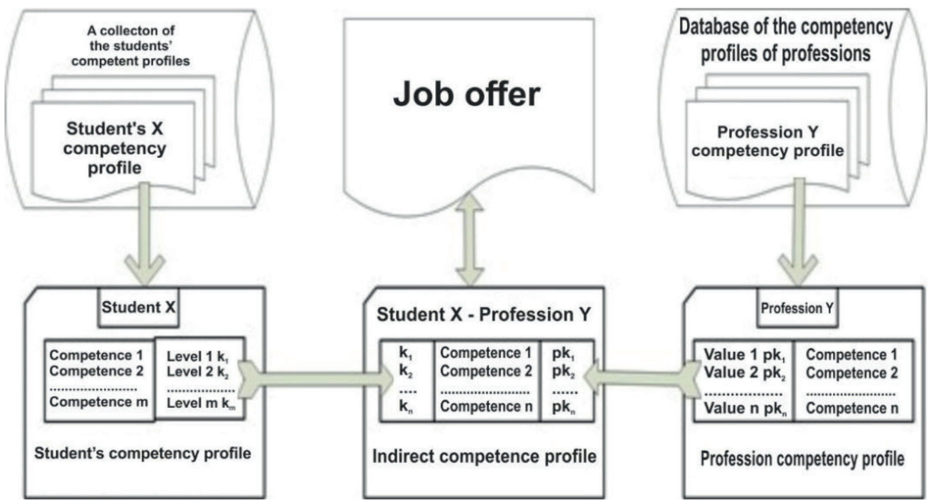

Figure 8. Diagram of the construction of an intermediate competency profile

In the case of the didactic profile, the entire set is most often used, but a single competence may correspond to several features of the didactic profile. It is then necessary to construct an appropriate transformation matrix.

The construction of an intermediate competency profile (a common competency dictionary) enables the unification of the description of profiles, and thus the determination of the similarity coefficients of information systems objects, i.e. matching the didactic profile of a graduate (or student) to the competency profile of the profession (job position), i.e. the analysis of employment opportunities based on acquired competences .

6 Educational programs (curricula)

The main elements of the education system that all activities related to the management of education quality relate to are the curricula. They are constructed for each field of study and level of study and for each course conducted as part of the University's activities and concern the entire period of study for a given recruitment (i.e. the program adopted in the recruitment year is assigned to all students of a given field of study in this recruitment).

The learning programs themselves are lower-level systems that are the objects of the super-system. The set of features for these objects are sets of subjects (modules), and the assigned values are the values calculated on the basis of the grades of all students following the program. These values can be defined, for example, as the average of the grades or in a different manner adopted at a given university. It seems to us that taking the average of the marks is the best way.

The programs have to be transformed into their didactic profiles, in which the characteristics are the field of study learning outcomes (FSLO), and the values are the averaged values determining the level of their achievement. The form of the transformation is the same as the one described earlier and shown schematically in Figure 2, the transformation of the student's results into his / her didactic profile.

It is also necessary to construct complementary curriculum profiles, based on students 'self-assessments and employers' assessments. The values for the achieved levels are the average of the respective values for all students following the program. In the same way, it is possible to construct a complementary didactic profile of the program based on the grades of the candidates for the studies.

7 The essential rules of the system knowledge base

The process of managing the quality of education at the University should consist in constantly adapting the offer to the requirements of the labor market and social needs. The sales offer of each university are primarily education programs (studies, postgraduate studies, courses). The quality of education will therefore be determined primarily by the quality of education programs considered in the context of the entire social environment. In the processes of creating and modernizing programs, the opinions of external social groups interested in the education process and graduates should be of great importance, i.e. primarily potential candidates for studies and employers. The use of an IT system ensures the influence of these groups, inter alia, by participating in acquiring value for the core and complementary didactic profiles.

Taking into account the opinions of external groups when shaping education programs is not a characteristic feature only of an IT system supporting the management of the quality of education. They are also taken into account in the absence of such a system (Student Career Office, University Business Council, etc.), but the system should significantly facilitate the collection of opinions and the preparation of applications on the basis of the already collected applications and the process of implementing applications. It can be assumed that the implementation of the system should increase the effectiveness of the University's cooperation with the environment.

The quality of education depends not only on the quality of the programs, but also on the quality of their implementation, i.e. the teaching staff. As the methods of staff evaluation have been developed and implemented for many years in each University by the Education Quality Departments and have even received the ordinance of the Minister of Science and Higher Education, we did not pay attention to them in our study. It can only be mentioned that the education quality management support system should contain a module for this purpose and that the implementation of the system will certainly facilitate and accelerate the assessment of the staff.

As mentioned earlier, the process of managing the quality of education at the University should be a continuous process. Analyzes and introducing changes to the curricula should be possible after each sub-stage of education - after each semester. Of course, not all such analyzes and conclusions drawn from them need to be implemented immediately - often you can wait with them until the end of the entire cycle. However, they can be used to modernize the programs that follow. For example: applications obtained after the end of the 1st semester of the 2015/2016 academic year may be included in the program, the implementation of which will begin in the 2016/2017 academic year.

In principle, it can be stated that the discussed system fulfills the features of an Total Quality Management System (TQMS), and its operation can be described in accordance with the Deming methodology [e.g. 2].

On the basis of the presented assumptions, it can be concluded that the proposed IT system has the features of an expert system. Therefore, the rules grouped in the knowledge base of the system play a key role in the system. They can be divided into several classes:

1. Rules for determining (assigning) the values of the levels of achieved competences in profiles.

The proposals of these rules have already been described in the earlier parts of the study, along with the definitions of individual profiles. They must include the fill matrices.

2. Rules for the transformation of features and values between profiles.

They mainly concern the transformation of education programs into the corresponding didactic profiles of programs (described in chapter 6) and the transformation of didactic profiles and competency profiles of professions (jobs) into a general competence profile. The last of the transformations mentioned should also enable the reverse transformation: from the general competency profile to didactic profiles or the competency profile of professions.

The rules of this class should be individually determined by the appropriate bodies for each field of study at each University, because the programs and learning outcomes for various fields of study at different universities are different.

3. Rules for optimizing education programs.

The rules of this class should make it possible to compare: didactic profiles of the education program (basic and complementary) and the relevant competency profiles of professions (jobs). Conclusions from this comparison should make it possible to distinguish groups of subjects (modules) within each curriculum, which: do not have to be modified, should be modified, should be removed or replaced, and should be introduced.

In the case of the rules for comparing didactic profiles, it seems most reasonable to compare the values that define the levels of competences achieved. During the comparison, any chosen method of testing the hypothesis about the equality of the mean values of the parameters of different trials can be used. It seems that due to the nature of the data, it would be advisable to use inference methods under conditions of high uncertainty (the use of artificial intelligence methods).

However, in the case of inference taking into account competency profiles of professions, it seems reasonable to analyse the frequency of occurrence of a given competency (transformed into the general competency profile and then into the didactic profile of the education program).

In addition to the main function of optimizing education programs, the system should enable the implementation of other, quite important and aforementioned tasks. Among other things, it should:

1. Enable each student to follow and even compose their own learning path.
2. Make it possible to examine the integrity and coherence of educational programs.
3. Facilitate the University's contacts with the environment and ensure the participation of interested groups in shaping educational programs.
4. Facilitate and speed up the evaluation of teaching staff.

The system should have a module ensuring reporting (descriptive and in the form of graphic visualizations) of all analysis results and defined change structures.

A very important feature of the system should be the possibility of archiving historical data - that is, it should be based not on a database but on a data warehouse.

8 Examples of the use of the structure of the IT system supporting the management of the quality of education and the application of knowledge base rules

The presented structure was used in the MinervaEDU system developed by eDialog (http://www.edialog.pl/) and tested at the University of Social Sciences (formerly the Social University of Entrepreneurship and Management) based in Lodz in 2011-2014 at the Faculty of Studies International and Computer Science, majoring in Computer Science; and at the Faculty of Social Sciences and Humanities, majoring in Administration. A more detailed description of the system functionality can be found on the eDialog website. The authors of this work and employees of the USS Promotion Department participated in cooperation with the company.

The knowledge base system and rules were also presented to the representatives of the Polish Accreditation Committee during a visit in March 2014, and the presentation was met with great interest.

According to information obtained from eDialog, the system was implemented at the University of Lodz, at the Faculty of Management and at the Faculty of Physics and Applied Computer Science.

The system uses the rules and algorithms presented in the publication by Przybyszewski et al. [14]. The rules mainly use an algorithm based on the classic method of calculating the degree of achievement of learning outcomes, using students' grades in individual subjects and the corresponding ECTS points taken from the dean's office system.

The verification of the assumed learning outcomes is carried out according to the following algorithm:

1. The procedure of verification of the assumed learning outcomes is based on the following assumptions:

   - „*i*” – designation of individual subjects included in the plan of study,

   - „*j*” – designation of individual learning outcomes (separately in terms of knowledge, skills and social competences),

- k_{ji} – coefficient determining the degree of influence of the subject "i" on the educational effect "j"; the value of the k_{ji} factor is given by the following formula:

$$\mathbf{k_{ji}} = \frac{\mathbf{p_i}}{\sum_{\mathbf{l \in \{impact\ on\ j\}}} \mathbf{p_l}}$$

where p_i is the number of ECTS points for the "i - th" subject and the denominator is the sum of points for all subjects contributing to the "j - th" directional learning outcome.

- w_i – weighting factor for the subject "i", specifying the degree to which the given learning effect "j" assumed for this subject has been achieved; the values of the weighting factors are determined on the basis of the grades obtained by the student according to the following formula:

$$\mathbf{w_i} = \frac{\mathbf{o_i}}{\mathbf{5}}$$

where: o_i is the grade value. The formula is used for calculations based on averages, the values of which do not have to be integer values. In the case of the assessment values adopted in the USS, the scale takes the following values:

Grade	w_i
failing to pass	0,0
satisfactory	0,6
better then satsfactory	0,7
good	0,8
better than good	0,9
very good	1,0

2. The level of implementation of a specific didactic outcomes effect "j" in a given semester is defined as the sum of the k_i coefficients calculated for individual subjects carried out in this semester and influencing a given educational effect:

$$\mathbf{E}_j = \sum_{\mathbf{i \in \{impact\ on\ j\ in\ a\ given\ semester\}}} \mathbf{k_{ji}}$$

Łączny stopień realizacji wszystkich kierunkowych efektów kształcenia w danym semestrze (oddzielnie w zakresie wiedzy, umiejętności i kompetencji społecznych) wyznacza się następująco:

$$\frac{\sum_{\mathbf{j \in \{learning\ outcomes\ in\ the\ semester\}}} \mathbf{E_j}}{\mathbf{n}}$$

where "n" is the number of learning outcomes for a given field of study - (separately in terms of knowledge, skills and social competences).

Similarly, it is possible to determine the degree of implementation of a specific field effect "j" throughout the study cycle and the total degree of implementation of all field learning outcomes, assuming that the k_{ji} factors from all semesters (the entire course of study) are taken into account.

3. The degree of achievement of a specific learning effect "j" in a given semester by a particular student is the sum of the products of the weighting factors "w_i" corresponding to the grades obtained by the student in the individual subject "i" by the coefficients k_{ji}:

$$\mathbf{S_j = \sum_{i \in \{impact\ on\ j\}} w_i \cdot k_{ji}}$$

The total degree of achievement by the student of the specific learning outcomes in a given semester (separately in terms of knowledge, skills and social competences) is determined as follows:

$$\frac{\displaystyle\sum_{j \in \{\text{a given semester, a given type of learning outcomes}\}} \mathbf{S_j}}{\displaystyle\sum_{j \in \{\text{a given semester, a given type of learning outcomes}\}} \mathbf{E_j}}$$

This proportion, expressed as a percentage, is the basis for the verification of the learning outcomes, as it determines the degree to which the student has achieved the expected faculty learning outcomes in accordance with the study plan, which can be achieved in this semester. Similarly, it is possible to determine the degree of achievement of a specific learning effect "j" for a given student in the entire study cycle, assuming that the k_{ji} and w_i coefficients for all semesters (the entire course of study) are taken into account.

EXAMPLE:

Let us assume that the subjects / modules: P1, P2, P3, P4, P5, P6 and P7 (their respective forms) influence (in accordance with the matrix of completing the learning outcomes by modules attached to the Program) on two learning outcomes: EKW1 (in the area of knowledge) and EKU1 (in the area of skills) as given in the table below.

If the values of ECTS points and obtained grades for given subjects / modules are as given in the table, it is possible to calculate all values of predefined parameters.

Subject	Form of didactic classes	Semester	ECTS	Grade	Impact on the learning outcome	
					EKW1	EKU1
P1	Lecture	2	4	4	x	
	Exercises			4,5		x
P2	Lecture	3	2	5	x	

P3	Lecture	3	5	3,5	x	
	Lab			5		x
	Design			5		x
P4	Lecture	4	4	3	x	
	Lab			4		x
P5	Lecture	5	2	5	x	x
P6	Lecture	6	5	3	x	
	Lab			3		x
P7	Seminar	7	3	5		x

Calculations:
For EKW1:

$$\sum_{l\epsilon\{impact\ on\ EKW1\}} p_l = 4 + 2 + 5 + 4 + 2 + 5 = 22$$

Semester	ECTS	$p_{semesetr}$	The level of achieving the educational effect
1	0	0,00	0,000
2	4	0,18	0,144
3 P2	2	0,23	0,230
3 P3	5	0,09	0,063
4	4	0,18	0,108
5	2	0,09	0,090
6	5	0,23	0,138
7	0	0,00	0,000

The degree of implementation of EKW1 in individual semesters - E_i (and total in all preceding ∑Ei) is as follows:

Semester	$E_{semesetr}$	$\sum E_{semester}$	The level of achieving the educational effect
1	0,00	0.00	0,000
2	0,18	0,18	0,144
3 P2	0,09	0,50	0,374
3 P3	0,23		0,437
4	0,18	0,68	0,545
5	0,09	0,77	0,635
6	0,23	1,00	0,773
7	0,00	1,00	0,773

That is: after completing the entire education cycle, the student achieved the level of 0.773 of the expected learning outcomes (which is equivalent to 77.3% or the grade: 3.87 or 4.0 after rounding).

For EKU1:

$$\sum_{l\epsilon\{impact\ on\ EKU1\}} p_l = 4+5+5+4+2+5+3 = 28$$

Semester	ECTS	$p_{semester}$	The level of achieving the educational effect
1	0	0,00	0,000
2	4	0,14	0,126
3 P3 Lab	5	0,18	0,180
3 P3 Project	5	0,18	0,180
4	4	0,14	0,112
5	2	0,18	0,108
6	5	0,11	0,066
7	3	0,07	0,070

The degree of implementation of EKU1 in individual semesters - E_i (and total in all preceding $\sum Ei$) is as follows:

Semester	$E_{semester}$	$\sum E_{semester}$	The level of achieving the educational effect
1	0,00	0,00	0,000
2	0,14	0,14	0,126
3 P3 lab	0,18	0,32	0,306
3 P3 Projekt	0,18	0,50	0,486
4	0,14	0,64	0,598
5	0,18	0,82	0,706
6	0,11	0,93	0,772
7	0,07	1,00	0,842

That is: after completing the entire education cycle, the student achieved the level of 0.842 of the expected learning outcomes (which is equivalent to 84.2% or the grade: 4.21 or 4.0 after rounding).

In the same way, it is possible to calculate the level of achievement of the assumed learning outcomes for subjects / modules by inserting, instead of the student's grades, the average of the grades of students pursuing a given subject in the same semester.

In the same way, it is possible to calculate the level of achievement of area learning outcomes for each student by substituting for subjects / modules, directional learning outcomes, instead of student's grades, their levels of achieving the assumed learning outcomes, and for directional learning outcomes, area learning outcomes and using the matrix of filling area effects education through directional learning outcomes.

9 Conclusion

Before starting the design of an IT system supporting the quality management of education at the University, participants of the executive team often asked the question: is such a system even needed? After all, all education quality management procedures are codified in each University and this process is always carried out by the Education Quality Department in practice, also with the participation of external stakeholders. The experience gained during the implementation of the National Qualifications Framework at the University of Social Sciences confirmed the positive answer to this question. It turned out that it would be difficult to analyze such a huge amount of data in a short time and draw reasonable conclusions from it without the support of an appropriate IT system. Therefore, we started testing such a system.

The first step was to formulate assumptions regarding the structure (specification of basic elements) and functionality of the designed system.

The core of the system is a database with an appropriate structure ensuring appropriate grouping of these data. Due to the nature of the data, it must be linked with the University's dean's office database (and actually of each Faculty). The data has been grouped into structures that enable dynamic supplementation of the features and values of the didactic and competency profiles described in the study. Using the rules grouped in an appropriate database, it is possible to compare these profiles and on this basis infer the effectiveness of a specific education program and propose changes to the curriculum.

By defining appropriate access rights to the system, it is possible for external stakeholders to participate in the process of optimizing education programs, without the need for direct discussion (meeting) with them, which significantly increases their involvement in the process of improving the quality of education at the University. The participation of employers was guaranteed thanks to the job offers they submitted (in many cases, job offers are obtained from the media by employees of the Career Office) and the evaluation of own employees who are graduates or students of the University. The participation of candidates for studies is gained by completing an appropriate evaluation questionnaire for the education program they are interested in.

It should be noted that the most important element of the IT system is the rule base used in the system. Without defining them, it is impossible to define quantitative indicators of adjusting the adopted education system to the requirements of the environment (labor market). Due to the fuzzy nature of both the data taken into account and the fuzzing of the inference rules themselves, it would be advisable to use various artificial intelligence technologies and soft computing techniques to determine the value of estimators determining the quality of education. The amount of data taken into account in the calculations and their nature also indicate that it would be convenient to use neural networks, especially machine learning and deep learning methods. This is the direction of the activities of the team represented by the authors of this study.

References

1. Bednarczyk H., Woźniak I., Kwiatkowski S.M., (red.), *Krajowe standardy kwalifikacji zawodowych. Rozwój i współpraca,* Ministerstwo Gospodarki, Pracy i Polityki Socjalnej, Warszawa 2004.
2. Hamrol A., Mantura W., *Zarządzanie jakością. Teoria i praktyka*, PWN, Poznań 1999.
3. Krajowe Standardy Kwalifikacji Zawodowych, MPiPS, [Online]. Dostępne: http://www.kwalifikacje.praca.gov.pl/
4. Kraśniewski A., *Jak przygotowywać programy kształcenia zgodnie z wymaganiami wynikającymi z Krajowych Ram Kwalifikacji dla Szkolnictwa Wyższego*?, MNiSW, 2011, [Online]. Dostępne: http://www.nauka.gov.pl/g2/oryginal/2013_05/66ac97d0be9ed5cec5e62b50fbd41d2.pdf
5. Kwiatkowski S.M., Woźniak I. (red.), *Krajowe standardy kwalifikacji zawodowych. Projektowanie i stosowanie,* Ministerstwo Gospodarki, Pracy i Polityki Socjalnej, Warszawa 2003.
6. Kwiatkowski S.M., Woźniak I. (red.), *Krajowe standardy kwalifikacji zawodowych. Kontekst europejski,* Ministerstwo Gospodarki, Pracy i Polityki Socjalnej, Warszawa 2004.
7. Kwiatkowski S.M., Woźniak I., *Elementy metodologii projektowania standardów kwalifikacji zawodowych*, [w:] Bednarczyk H., Woźniak I., Kwiatkowski S.M., (red.), *Krajowe standardy kwalifikacji zawodowych. Rozwój i współpraca,* Ministerstwo Gospodarki, Pracy i Polityki Socjalnej, Warszawa 2004, s. 33-45.
8. Majcher-Teleon A., *Krajowe Ramy Kwalifikacji a budowa systemu uczenia się przez całe życie,* [w:] Bednarczyk H., Woźniak I., Kwiatkowski S.M., (red.), *Krajowe standardy kwalifikacji zawodowych. Rozwój i współpraca,* Ministerstwo Gospodarki, Pracy i Polityki Socjalnej, Warszawa 2004.
9. Moos J., Sienna M., *Rola i zadania standardów kwalifikacji zawodowych w programowaniu zawodowego kształcenia ustawicznego*, [w:] Bednarczyk H., Woźniak I., Kwiatkowski S.M., (red.), *Krajowe standardy kwalifikacji zawodowych. Rozwój i współpraca,* Ministerstwo Gospodarki, Pracy i Polityki Socjalnej, Warszawa 2004.
10. Podstawa programowa wychowania przedszkolnego oraz kształcenia ogólnego w szkołach podstawowych, gimnazjach i liceach [Online]. Dostępne: http://www.men.gov.pl/index.php/2013-08-03-12-10-01/podstawa-programowa/197-podstawa-programowa-wychowania-przedszkolnego-oraz-ksztalcenia-ogolnego-w-szkolach-podstawowych-gimnazjach-i-liceach

11. Przybyszewski K., Cader A., Filutowicz Z., *Automatyzacja i obiektywizacja ewaluacji w procesach kształcenia*, [w:] Cader A. *et al.* (red.), *Wybrane zagadnienia inżynierii wiedzy*, Wydawnictwo SWSPiZ, Łódź, 2008, ss.: 36-76.
12. Przybyszewski K., Cader A., Marchlewska A. *O możliwościach zastosowania wybranych metod wyznaczania efektywności klasycznych i e-learningowych systemów kształcenia*, Automatyka, 13, 3, Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH Kraków, 2009.
13. Przybyszewski K., Filutowicz Z., Cader A., *Standaryzacja wyników ewaluacji studenta/ucznia dla potrzeb prognozowania efektywności klasycznych i e-learningowych systemów kształcenia*, [w:] Niedźwiedziński M., Lange-Sadzińska K. (red.), *Wybrane problemy zastosowania platformy elektronicznej*, Consulting, Łódź, 2009.
14. Przybyszewski K., Filutowicz Z., Gawryś J., Piasecki P., *Propozycja transformacji wyników nauczania studentów i absolwentów uczelni wyższych do opisu kompetencyjnego kwalifikacji zawodowych*, Studia i Materiały Polskiego Stowarzyszenia Zarządzania Wiedzą nr 53, Bydgoszcz 2011, s. 141-153.
15. Przybyszewski K., *Zastosowanie zbiorów rozmytych do ewaluacji różnych aspektów systemów kształcenia*, Automatyka, 3 (12), Uczelniane Wydawnictwa Naukowo - Dydaktyczne AGH, Kraków, 2008, s.: 1033-1045.
16. Przybyszewski K., Filutowicz Z., Założenia modelu systemu informatycznego wspomagającego zarządzanie jakością kształcenia w uczelni, Studia i Mate-riały Polskiego Stowarzyszenia Zarządzania Wiedzą nr 71, Bydgoszcz, 2014, ss. 203-216
17. Przybyszewski K., Filutowicz Z., Właściwości informatycznych systemów wspomagających zarządzanie jakością kształcenia, [w:] Cader A., Przybyszewski K., Goetzen P., Marchlewska A., Grzanek K. (red.) Zagadnienia współczesnej in-formatyki. Wybrane zastosowania technologii informatycznych, Wydawnictwo SAN, 2014, ss. 141-158
18. Rozporządzenie Ministra Edukacji Narodowej z dnia 23 grudnia 2008 r. w sprawie podstawy programowej wychowania przedszkolnego oraz kształcenia ogólnego w poszczególnych typach szkół
19. Rozporządzenie Ministra Edukacji Narodowej z dnia 23 grudnia 2008 r. w sprawie podstawy programowej wychowania przedszkolnego oraz kształcenia ogólnego w poszczególnych typach szkół *(rozporządzenie zostało opublikowane w Dzienniku Ustaw z dnia 15 stycznia 2009 r. Nr 4, poz. 17)* [Online]. Dostępne: http://bip.men.gov.pl/men_bip/akty_prawne/rozporzadzenie_20081223_v2.pdf
20. Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 2 listopada 2011 r. w sprawie Krajowych Ram Kwalifikacji dla Szkolnictwa Wyższego [Online]. Dostępne: http://www.nauka.gov.pl/g2/oryginal/2013_05/478e9241dffed3a0bcd4fb28792392a8.pdf
21. Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 30 września 2016 r. w sprawie charakterystyk drugiego stopnia PRK typowych dla kwalifikacji uzyskiwanych w ramach szkolnictwa wyższego po uzyskaniu

kwalifikacji na poziomie 4 – poziomy 6-8, [Online]. Dostępne: http://www.nauka.gov.pl/g2

22. Rutkowski L.: *Metody i techniki sztucznej inteligencji*, PWN Warszawa 2005
23. Stachowiak-Kudła M., *Autonomia szkół wyższych a instytucjonalne mechanizmy zapewnienia jakości w Polsce i wybranych państwach europejskich*, Wydawnictwo Difin, Warszawa 2012
24. Ustawa z dnia 18 marca 2011 r. o zmianie ustawy – Prawo o szkolnictwie wyższym, ustawy o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki oraz o zmianie niektórych innych ustaw, [Online]. Dostępne: http://www.nauka.gov.pl/g2/oryginal/2013_05/74cd500b0d3c8043ab5f1a0114884573.pdf
25. Zalecenia Parlamentu Europejskiego i Rady z dnia 23 kwietnia 2008 r, [Online]. Dostępne: http://www.nauka.gov.pl/g2/oryginal/2013_05/913801416e23b7f4a3c171cecc67b96c.pdf

JACSM 2017, Vol. 9, No. 2, pp. 59 - 67
10.1515/jacsm-2017-0009

AUTOMATION OF SOFTWARE INSTALLATION IN SERVICE WORK

Kamil Zakrzewski

Information Technology Institute,
University of Social Sciences, Lodz, Poland
kzakrzewski@san.edu.pl

Abstract

The use of automation to improve certain processes can bring many benefits. One of the potential areas of its use can be automation of service work at a computer service. Routine software installation carried out by an employee may result in many mistakes. The use of automation not only eliminates them, but also speeds up the entire installation process and reduces the employee's duties.

Key words: optimization maintenance, automatic installation, software installation

1 Introduction

By carrying out frequent maintenance work related to the installation of software on computers, it can be easily stated that to many operations are repetitive. Routine activities during this process can lead to many mistakes for e.g. forgetting to install a specific software or its proper configuration. One the possible solution of this problem is automation of whole process which in turn should bring not only benefits of reducing the number of mistakes but also to speed up the whole process of installation software and at the same time make the employee's work easier [5].

This publication describes the division of applications due to the distribution method, typical service work of the service technician during the software installation and the impact of applying automation on the software installation process was presented.

2 Software Division

Each application and service software can be distributed in many ways. We distinguish three possibilities of software distribution: installing, portable or web. It is worth mentioning, that the same application can simultaneously occur in three variants [2].

Installed software is the one which fully or partially integrates with the operating system. The application is most often in the form of a compressed installation file [1] e.g. the installer of Apache Open Office or in the case of computer games, installer file with compressed archives files, which ones the installer decompress. The process of installation usually proceeds in the same way. At the beginning, the files are decompressed, if any of them exist in this form, and then the installer copies necessary files to the target location and creates entries in the system registry, by informing the operating system about new file extension, which ones this application can open [3].

In the contrary to installed software on the target computer, portable software represents its special variety. It doesn't interfere with the operating system, and that's why it has both disadvantages and advantages. From the assumption that all configuration files the application stores in the same directory or subdirectory, where there is its execution file. It makes it possible to transfer of the whole application together with user settings among different computers. In the case of installed software it is impossible, as most often configuration files and settings are saved in the registry of operating system or in the user directory. If the application doesn't create configuration files, and its run is possible without installation, it is treated as portable as well [4].

The different group is represented by web application called Single Page Application. In order to use this sort of application, the Web browser is essential. In contrast to installed software or portable one, the application itself is located on server outside of the user's computer [9]. The advantage such solution is easy access to application from any computer. Biggest disadvantage is the problem with the access to application in case of server failure or problems with the Internet connection. The main feature of web applications is fact that they are presented as one web page. In the case of making changes, the content of the page is refreshed and not its whole page just as in the case of classic websites [7].

3 Routine Procedure of Software Installation

The main problem during the software installation is the possibility of the occurrence of routine. The standard course of this process assumes preparing software for installation on an external carrier e.g. USB Flash Drive and their own installation by service worker [10]. Length of this process can vary considerably depending on hardware performance. In the case of newer computers equipped with more efficient hard drives, the whole process can take a few minutes, and in the case of older hardware even over a dozen. As a result of such dispersion in the length of installing software is inability to estimate how much time must a worker take to complete the task.

One of the ways of eliminating the routine associated with software installation is reduction employee's duties by manual software installation. It can be done by creating automatic installer, which independently of serviced hardware performance make it possible for the worker to take care of other duties, and the application itself will provide faultless of the given software installation.

4 Example of User Interface of Software Installing Application

The choice of the software for installation should be readable, as well as simple to use. The application should allow selection of the required software for installing and the choice of the correct version 32-bit or 64-bit.

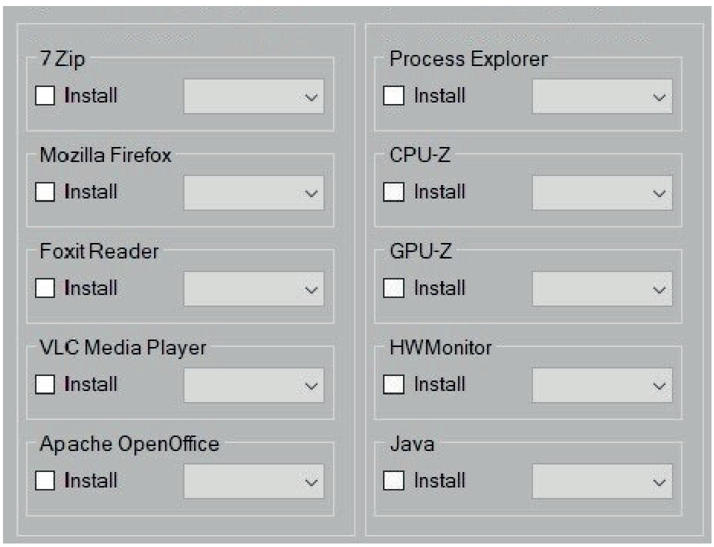

Figure 1. Example of application user interface (own execution).

Application prepared in this way can be additionally expanded in the future for an option to detect current version of the operating system in order to automatically select software as well as the log system.

5 Test Platforms and Selected Software for Installation

Tests carried on computers with different performance allow for better mapping of typical service work conditions. The age of computers presented here ranges from 2007 (Intel Core 2 Duo E4600 release) to 2017 (AMD Ryzen 7 1800X release).

Table 1. Computers and their components part 1 (own execution).

ID	Model	USB	CPU	RAM
S_1	ASRock Taichi X370	3.1	AMD Ryzen 7 1800X	32GB DDR4
S_2	MSI X370 Gaming Pro Carbon	3.1	AMD Ryzen 5 1600	16GB DDR4
S_3	Gigabyte GA-G31M-S2L rev:1.0	2.0	Intel Core 2 Duo E4600	2GB DDR2
S_4	Gigabyte AB350K-DS3H-CF	3.0	AMD Ryzen 5 2600	8GB DDR4
S_5	Gigabyte GA-M52L-S3 rev:1.0	2.0	AMD Athlon x64 6000+	2GB DDR2
S_6	Foxconn A74ML-K	2.0	AMD Athlon x64 6000+	4GB DDR2
S_7	Gigabyte GA-MA770-DS3	2.0	AMD Athlon x64 5000+	4GB DDR2
L_1	Lenovo Y700	3.0	Intel Core i7-6700HQ	16GB DDR4
L_2	Lenovo Z70-80	3.1	Intel Core i7-5500U	8GB DDR3
L_3	Samsung rc510	2.0	Intel Core i3-380M	4GB DDR3
L_4	Lenovo G570	3.0	Intel Core i5-2450M	8GB DDR3

Table 2. Computers and their components part 2 (own execution).

ID	HDD Type	HDD Capacity	RPM	OS
S_1	SSD (M.2 NVMe)	120GB	---	W10 Pro x64
S_2	SSD (SATA)	128GB	---	W10 Pro x64
S_3	HDD (ATA)	640GB	5400	W7 Home x32
S_4	HDD (SATA)	1000GB	7200	W10 Pro x64
S_5	HDD (SATA)	120GB	7200	W7 Home x32
S_6	HDD (ATA)	160GB	7200	W7 Home x64
S_7	HDD (SATA)	250GB	5400	W7 Home x64
L_1	SSD (M.2)	120GB	---	W10 Pro x64
L_2	SSHD (SATA)	1000GB	5400	W7 Pro x64
L_3	HDD (SATA)	500GB	5400	W7 Home x32
L_4	HDD (SATA)	500GB	5400	W7 Pro x64

The analysis applies only to computers with Windows 10 or Windows 7 installed. The reason of such a choice is the fact that according to the statistics of Steam platform (May 2019) these are the most often used operating systems among the users of this platform [12].

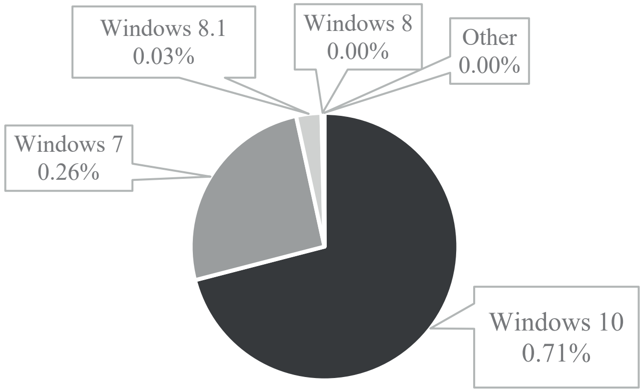

Figure 2. Diagram of used Windows operating systems on the Steam platform (own execution based on data [12]).

The chosen software for carrying out tests of manual and automatic installation has been presented in table 3. If the application doesn't exist in 64-bit version, and it will be installed on 64-bit operating system, its 32-bit equivalent is installed.

Table 3. Selected software for testing (own execution).

ID	Application	32-bit	64-bit
1	7zip	X	X
2	VLC Media Player	X	X
3	Google Chrome	X	X
4	Apache OpenOffice	X	-
5	Foxit Reader	X	-
6	Java (offline)	X	X
7	Process Explorer	X	-
8	HWMonitor	X	X
9	GPU-Z	X	-
10	CPU-Z	X	X

6 Methodology of Installation Time Measurements

All the applications intended for manual installation are run exactly in the same order. In the case of an automatic installation the user chooses only whether to install application on 64-bit or 32-bit operating system.

Time measurement during manual installation was completed by means of a program written in C# language returning only the current time with an accuracy of milliseconds. One of the possibilities it was also using system function "time" on command line, but in the case of Windows 7 operating system, this function returned time without milliseconds [6, 8].

Calling the application returning current time during its start-up, was realized in BAT script, which later is starting 7 zip application installer. At the end of the script it is necessary to use "pause" function in order to prevent closing console windows with displayed time [11].

```
call _TIMER.exe
start 7zx64.exe
pause
```

Exactly the same method was used during registering the time of finishing the installation of the last application. The exception is only starting the installer by means of "call" function instead of "start". The result of this is suspension of the script until the finishing of the called process. Thanks to this after the finishing of installer operation the time returning program will be called.

```
call javax64.exe
call _TIMER.exe
pause
```

From such obtained starting and finishing time of installation the total length of the whole process was calculated. In the case of automatic installation the time was recorded the moment of starting of the installation as well as its finishing.

7 Results

The results in Table 4 represent the average of 10 trials for manual and automatic installations.

Table 4. Average times of manual and automatic installation (own execution).

ID	Manual	Automatic
S_1	154,09	85,84
S_2	175,02	93,10
S_3	309,15	196,08
S_4	187,58	106,30
S_5	327,14	219,23
S_6	323,64	217,84
S_7	467,63	374,71
L_1	220,98	102,86
L_2	197,53	107,43
L_3	276,33	198,37
L_4	228,49	165,37

Presentation of data from table 4 in the chart's figure better shows the difference between the use of automation and manual installation. It is easy to notice on the graph that despite the differences in the performance of computer hardware, there is no significant deviation in the difference in the length of software installation. In fact, the difference in the installation methods used is due to the time the user handles the graphical interface. The use of automation at the same time eliminates the need to manually operate the interface of each of the installed applications, which contributes to saving time.

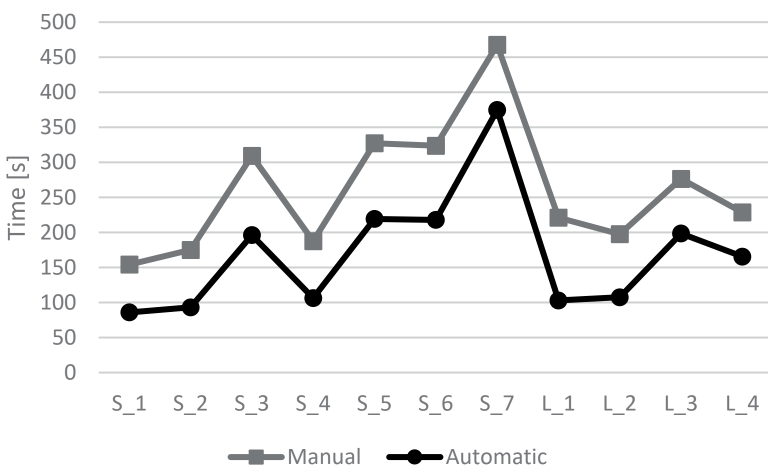

Figure 3. Graph of manual and automatic installation times (own execution).

8 Conclusion

Currently, the number of duties during service work related to computer hardware is constantly increasing, which is why there is a need to automate as many of them as possible. Thanks to this, not only the chance of human error is reduced, but also the efficiency of work performed by employees is increased. Instead of dealing with the monotonous software installation process, employees can use their time for jobs that are not easily automated like hardware diagnostics.

It can be stated that the use of automatic installation during service work greatly accelerates the whole process. Automatic software installation was faster on average by 64.7% than manual installation. Regardless of the performance of the tested hardware, the human handling of user interface lasted on average 90 second. The remaining process of the proper installation, including decompressing archives and copying files by installers, had a large time dispersion depending on the performance of the computers. The most essential advantage of the created application is the reduction of the employee's duties. In the worst observed case, automatic installation lasted 374.71 seconds and was faster only by 19.87% from manual installation which lasted 467.63 seconds.

References

1. Adamczewski P., *Słownik Informatyczny.* Helion, 2005
2. Bucki J., *Software and Office Application Suites.* May 2019, https://www.thebalancesmb.com/what-is-a-software-suite-or-application-suite-2533660
3. Ceruzzi P. E., *A History of Modern Computing Massachusetts.* MIT Press, 2000
4. Glenn W., *What Is a „Portable" App, and Why Does It Matter.* May 2019, https://www.howtogeek.com/290358/what-is-a-portable-app-and-why-does-it-matter/
5. Kanat-Alexander M., *Understanding Software.* Packt Publishing, 2017
6. Liberty J., *C# Programowanie.* Helion, 2005
7. Mikowski M., Powell J., *Single Page Web Applications: JavaScript end-to-end.* Manning Publications, 2013
8. Microsoft, *Windows Commands.* May 2019, https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands
9. Monteiro F., *Learning Single-page Web Application Development*. Packt Publishing, 2014
10. Mueller S., *Rozbudowa i naprawa komputerów PC. Wydanie XVIII*. Helion, 2009
11. Szeląg A. *Windows 10 PL. Optymalizacja i zaawansowane zarządzanie systemem.* Helion, 2015

12. Valve Steam, *Hardvare survey.* May 2019, https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam?platform=pc

DE GRUYTER
OPEN

JACSM 2017, Vol. 9, No. 2, pp. 69 - 78

10.1515/jacsm-2017-0010

Offered traffic vs Load Dispersion Factor dimensioning of Mobile Networks

Mateusz Sztukowski

Research & Development et Nokia Wroclaw, Poland

mateusz.sztukowski@gmail.com

Abstract

Offered traffic is a basic parameter which is essential for carrying out the dimensioning of mobile networks, as well as IT networks. Usually, the source of its estimation would be the measurement of data in the core network (place of aggregation). Another solution would be to identify the average use by the end user, e.g. the amount of data sent in a month. In practise, this means that in the first instance, offered traffic was calculated on the side of the aggregation element, while in the second, on the side of the access network. Load Dispersion Factor (LDF) was defined [1] as the parameter which enables offered traffic to be estimated, regardless of its place of determination. It should be taken into account that the biggest mistake in network dimensioning is made when defining the traffic model [2], particularly offered traffic, and an incorrectly determined LDF can additionally aggravate this significantly, by as much as several hundred percent [1]. This paper is a summary of research and measurements [2] which show how to estimate the LDF value practically, and how this size took shape in the case of several dozen mobile operators.

Keywords: telecommunication networks; telecommunication network dimensioning; Load Dispersion Factor; offered traffic estimation.

1 Introduction

Dimensioning a network at the design stage is one of the key elements which determines the cost of the planned investment. In particular, in large projects such as the modernisation of GSM, UMTS or LTE mobile wireless networks, inaccurate dimensioning has a significant economic impact and can be a factor in the failure of such initiatives. This is a challenge which mostly revolves around the amount of unknowns and the lack of opportunity to estimate them, particularly when taking into account the state of a project in its initial stages, its scale and duration. It is worth emphasizing what these terms mean in the practise of implementing the technology under consideration. The initial stage means that the mobile network operator has to deliver criteria for network building such as the predicted traffic model, the area of implementation, development phases for the network, or business planning, at the stage of the RFP (Request for Proposal) or RFQ (Request for Quotation). The provider, meanwhile, has to present technical solutions, the SoC (Statement of Compliance) and the cost estimate at this stage.

The key element of the traffic model, offered traffic, can be interpreted as the average traffic generated by users who are independent of each other at the point when the network is used the most. Depending on the service under consideration, it may be in various units, e.g. erlang, volume as number of bites, or bandwidth in kbps. It is the basic variable in most dimensioning methods. In simplified terms, a directly proportional relationship could be supposed between the offered traffic value and the demand for minimal network resources. An error in determining it leads directly to overdimensioning or underdimensioning, both of which will ultimately lead to excess project costs. Unfortunately, no method has been developed to this point which would allow this parameter to be estimated at the network design stage. The desire to avoid this error at the initial stages of a project was a direct motivation for undertaking this research.

2 Correlation of LDF with Key Performance Indicators (KPIs)

Previous research [3, 4] has established the indicator of rush hour dispersion between LDF base stations. A linear influence on the offered traffic parameter by the level of rush hour dispersion has been demonstrated. Additionally, by analysing actual performance data from several dozen mobile operators, it has been established that this indicator is characterised by considerable variability. A doctoral thesis [2] proved that it is possible to build an effective prediction model for the level of rush hour packet traffic aggregation in mobile networks which is useful at the network design stage. This was based on the assumption that there is a dependency between the LDF and the performance parameters of the network being designed.

For each cell in mobile networks, several thousand various parameters are collected. Although the measurement data used in the research were gathered from several dozen operators (Table 1), in comparison to the several thousand variables analysed, the number observed was still insufficient. As a result, the data exploration algorithms were not very effective. For this reason, based on observation of indirect results of the experiment, the thesis was expanded by stating that if LDF is dependent only on the combination of load profiles and there is a correlation between LDF and performance parameters, then the load profiles also depend on the network's performance parameters. As a result of these assumptions, the path to a new and effective method of selecting relevant KPIs, to be used in building the LDF prediction model, was paved.

Table 1. Basic information on the database of performance parameters collected.

Region	# RNC	#cell
APAC (Asia and Pacific)	131	60802
Europe	69	39431
MEA (Middle East and Africa)	18	9307
America	7	9551
Sum	225	119091

3 Indirect way to select relevant KPIs

An alternative method of choosing variables was proposed, since the direct method failed. In Table 2 [2], several examples are presented in which variables mistakenly led to a high significance level value. They were frequently a group of specific parameters e.g. those indicating network overload, and their unusual nature related to them being found only in a particular measurement period. Another example of unnecessary parameters were variables related to rarely used services such as video, or packages with particular bandwidths. Since they only appeared sporadically, they had a large input into defining the data entropy, while in practise they were a fraction of a percentage of the traffic as a whole. Specific parameters are not significant in a global approach. An additional interpretation problem for specific parameters is that, on the one hand they have hardly any significance in a global approach, and additionally, they are very difficult to predict at the design stage. The aim of the indirect method of choosing variables was to solve the above mentioned problems. On the one hand, the problem of the faulty relationship between the number of variables and number of observations, and on the other, enabling the elimination of specific or rare parameters.

The indirect algorithm for selection of variables is based on not searching for a direct dependency between LDF and the performance parameters of the network, and instead being limited to correlations with the load profiles. For the learning algorithms, this meant transforming the goal parameter from LDF to a representative indicator of the RNC profile and considering each project independently (Table 3). The algorithm was divided into two key parts. The first is the creation of representative performance profiles at the level of a single network, described in [3, 4], and the second is the final process of categorizing performance counters. The concept of the category is in this case related to the generalized counter, understood at the level of the system as size bringing together all its more detailed appearances.

Table 2. Example of results of variable selection algorithm for original input data with their categorization according to specificity [2]

CounterName	Category	Relevance
Voice_CALL_ATTS	valid (#connectoin attempts)	0.8154533415
UE_SUPP_13_14	valid (#SmartPhones in network)	0.8092345765
NO_CODES_SF128	specific (overload)	0.7569323173
FAIL_PS_DCH_DL_CONN	specific (errors)	0.7495387803
RRC_CONN_FAIL_UE	specific (errors)	0.6690655444
DUR_USERS_13_16	valid (#users)	0.6596905854
RAB_Voice_HLD_TM	valid (connection duration)	0.6446040139

Table 3. Example of learning data for BTS cell profiles based on performance counters [2]

NodeB ID	PM_1	PM_2	...	PM_N	Profile ID
1_1_3	132	43.2	...		1
1_2_1	2423	11.1	...		3
1_2_2	222	22.7	...		1

Variable selection algorithms, also known in the literature as feature selection [5] serve to define the relevance of variable in terms of the goal parameter. However, their main use in practise is essentially reducing the number of variables before using data mining methods, such as predictive modelling or object grouping. To complete this task, the use of the variable selection node from SAS Enterprise Miner [6] was suggested. Parameters were chosen in which the significance level value is greater than 0.6. This value is a heuristic, which indicates between several and several dozen parameters for each project. Next, the results from all projects were combined into one table and categories were given to each variable. This type of categorization enables all the counters which are specific to one project, and associated with overload or network error, to be removed in a simple action. The next step is essentially to choose categories which are globally significant. This can be obtained by adding together the counters from each category. The more frequent the occurrence of a given category, the greater its global significance. This thesis is based on the assumption that the probability of choosing the same unrelated variables in various project is significantly lower than the probability of choosing the same related variables in various projects. The mathematical proof of this thesis was presented in [7] and described further. As a result of these activities, a table containing categories and their size is created. By sorting the table in descending order by size, a list of the most significant categories can be obtained.

The last stage of the indirect variable selection algorithm was to extract the significant parameters as based on the selected categories from the previous step. This step, similarly to the previous one, needs to be carried out using expert knowledge. The aim of this step is the widest possible coverage of relevant categories with counters, and this task should be delegated to an expert in that telecommunications field (radio or access). The widest possible coverage in this case means identifying which counters are accessible in the version of software currently used by the operator at base stations (NodeB) and in control networks (RNC). The result of this operation will be a list of new relevant KPIs in relation to the traffic load profile (Table 4) and so indirectly in relation to the LDF.

Table 4. Extract from the list of relevant parameters in relation to performance profiles and indirectly in relation to the LDF, marked on the basis of significant categories.

Nazwa parametru	Kategoria
PS_DL	Amount of transferred PS traffic
PS_UP	Amount of transferred PS traffic
PS_ATTS_RAB	PS connection attempts
PS_ATTS_RRC	PS connection attempts
PS_CONN_RAB	#parallel PS connection
PS_CONN_RRC	#parallel PS connection
PS_CONN_DCH	#parallel PS connection
PS_CONN_HSDSCH	#parallel PS connection
PS_RAB_TIME_AVG	PS connection duration
PS_RAB_TIME	PS connection duration
PS_DCH_TIME_AVG	PS connection duration
PS_DCH_TIME	PS connection duration
PS_DSCH_TIME	PS connection duration
PS_HSDSCH_TIME_AVG	Czas trwania połączeń PS
PS_HSDPA_USERS_AVG	Ilość użytkowników HSDPA per nadajnik

In the paper [2] the list of significant parameters was widened to include general information about the network. This information does not arise directly from analysis of the counters, but is available at the design stage as, for example, the percentage of service coverage of the area under consideration, or the amount of base stations with very low load (responsible for covering the area, not population). In this way, the list of parameters obtained is the final result of the indirect variable selection algorithm.

4 Predictive model for LDF

On the basis of analysis of historical performance data, the number needed for effectiveness of a model was defined. It was assumed that if the prediction error is not greater than 10% of the mean LDF value, then the model is effective. Knowing the demands of the model being built and being in possession of the result of the relevant parameters selection, the process of building a prediction model was carried out. In particular, the focus was on two methods: linear regression and a decision tree, since both methods enable an expert to interpret the variables used relatively easily. Obtaining a high level of effectiveness in one of these models opened the door to further limiting of variables, which is essential to estimate the level of packet traffic aggregation at the network design stage.

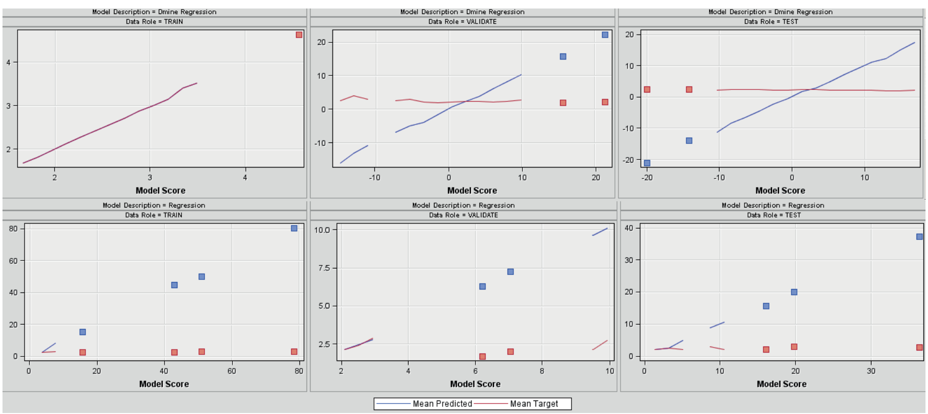

Figure 1. Analysis of the effectiveness of prediction models based on linear regression.

The model based on linear regression did not obtain results which met the quality standards for any data group: training, learning and testing. In each case, absolute error (RASE) exceeded the value one and grew along with the growth of predicted values (Figure 1 – lower graphs) this means that the correlation with LDF is not linear. An attempt was also made to tune a derivative method to DMineRegression linear regression, which uses additional PCA analysis [8] on the input data. However, in this case too, all the results led to the situation presented on Figure 1, which is similar to the classic curse of dimensionality problem [9]. An insufficient number of observations in relation to the amount of variables mean that the model was very adaptive to the learning data, but was ineffective in the new test.

The second direction, the decision tree, brought much better results. From the first attempts, a model with effectiveness that met standards was created, obtaining element values from mean squared error at level 0.2 (Figure 2). This situation allowed an attempt to be made to tune the model to the network dimensioning process. An interactive method of building a decision tree was used, allowing variables to be selected by an expert. When selecting a variable, the decision tree algorithm ranks the most relevant variables, and the tree should be divided into segments according to these. Sometimes the difference in significance between adjacent parameters on the list is very small or non-existent, but the algorithm will always choose the first parameter on the list. With an interactive interface, the expert can choose the variable which will be easiest to estimate from the point of view of dimensioning the network before each division of the tree. As it turned out, in several cases, the parameters ranked second or third, with a small difference in significance, confirmed these suppositions. With expert help, a model was created which turned out to be more effective than the one that was created automatically with the same configuration parameters. The increase in effectiveness had the added value of being noticeable for LDF exceeding the value of 2.5. As mentioned above, it is errors committed at this particular value allocation that are key for the investor or bidder.

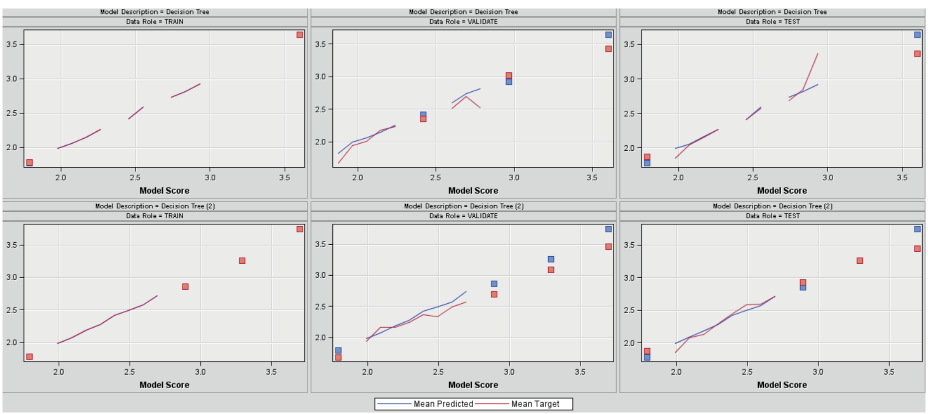

Figure 2. Analysis of the effectiveness of prediction models based on decision trees[2]

Four models which meet the expected quality criteria were built. Bearing in mind that the prediction model can be interpreted, the one based on the interactive decision tree is key. Analysing its individual leaves from the perspective of the possibility of dimensioning, it can be stated that the following KPIs are decisive in terms of potential overestimation of network resources [2]:

- Sum_peak_rnc – signifying offered traffic according to the new traffic model, which defines periodic use of the network by the user, e.g. monthly
- Ps_rab_time – total time of all packet connections.
- PS_TAB_TIME_AVE – average time of packet connection.

- PS_DL – amount of downlink traffic sent in packet services

5 Conclusion and practical aspects for LDF

The research presented was conducted at Nokia, one of the largest providers of mobile networks of all generations: 2G, 3G, 4G and 5G. Most of the research results were and are used in the process of dimensioning networks, particularly in estimating offered traffic. Dozens of projects have been carried out around the world based on these findings. Several dozen more are being carried out.

As a summary of a range of publications [1, 3, 4] and a doctoral thesis [2], it is worth comparing a few research conclusions in the context of the LDF parameter and offered traffic. They are used in practise both among operators and packet service providers, especially in mobile networks.

- It was demonstrated that approximately 1/10 of the traffic generated by users in 24 hours is sent at rush hour – a value which is essential for estimating offered traffic [1, 2].
- Key parameters which determine both the LDF parameter and the shape of the load profile for an element of the network are: offered traffic, understood as monthly usage, total packet call time, average packet call time [2]
- It is possible to build an effective prediction model for LDF based on parameters which can be estimated at the network building stage.
- It was demonstrated that LDF for voice services (Circuit Switched) has a significantly smaller influence on offered traffic. It is estimated to be at the level 1.2 – 14, which means that the risk of overdimensioning or underdimensioning is small if the average value is accepted. [1, 3]
- Based on the performance data gathered, it can be stated that the mean LDF value is about 2.3 (Figure 3) [2].
- Accepting the mean LDF value is an acceptable course of action for 55% of projects with 10% error and for about 80% with acceptable errors at the level of 20% (Table 5). [2]

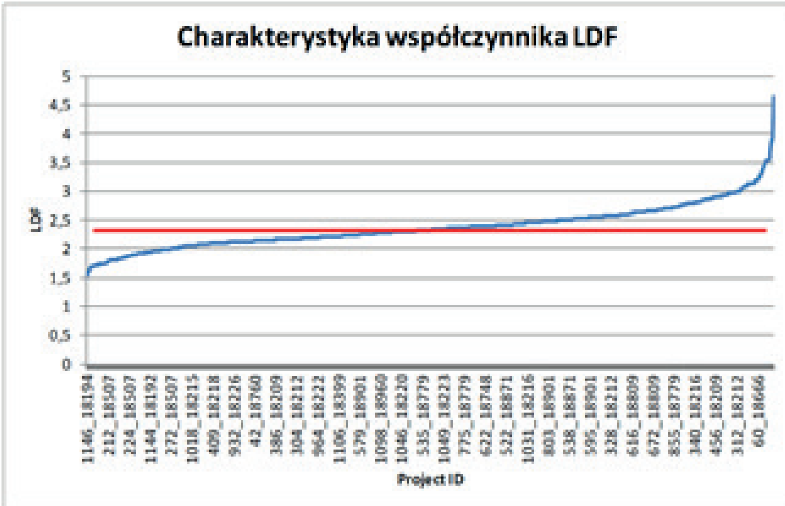

Figure 3. Values for the LDF coefficient from various projects.

Table 5. Characteristics of projects analysed from the point of view of erroneous estimation of LDF with average value accepted.

Inaccuracy	LDF		#networks	%share
	Bottom value	Upper value		
acceptable (+/- 10%)	2,11663	2,586992	517	55%
significant (10%>LDF>20%)	1,881449 2,586992	2,11663 2,822173	242	26%
Critical (>20%)	< 1,881449	> 2,822173	176	19%

References

1. Sztukowski M., Maciejewski H., Koonert M. and Bartłomiej C., *Dimensioning of Packet Networks based on Data-Driven Traffic Profile Modeling,* in *European Teletraffic Seminar*, Poznań, 2011.
2. Sztukowski M., *Model predykcyjny stopnia agregacji ruchu pakietowego w mobilnych sieciach bezprzewodowych*, PhD dissertation, Social Academy of Management, Lodz 2014
3. Maciejewski H., Sztukowski M. and Chowański B., *Traffic proffiling in mobile networks using machine learning techniques,* in Digital information processing and communications : international conference, ICDIPC, Ostrava, 2011.
4. Sztukowski M., Maciejewski H. and Cader A., *Enhanced Approach of Traffic Profiling for Dimensioning of Mobile Wireless Networks.,* in Lecture Notes in Computer ScienceVolume 7268, Zakopane, Springer, 2012, pp. 724-732.

5. Han J. and Micheline K., *Data Mining: Concepts and Techniques, Second Edition*, Elsevier, 2006.
6. SAS Institute Inc., SAS Enterprise Miner 12.3: Reference Help, SAS Institute Inc., 2010.
7. Maciejewski H., P*redictive Modelling in High-Dimensional Data: Prior Domain Knowledge-Based Approaches,* Oficyna Wydawnicza Politechniki Wrocławskiej, 2013.
8. Witten I. H., Frank E. and Hall M. A., *Data Mining: Practical Machine Learning Tools and Techniques,* Third edition, Elsevier, 2011.
9. Clarke B., Fokuue E. and Zhang H. H., *Principle and Theory for Data Mining and Machine Learning,* Springer, 2009.

Information for Authors

Authors are invited to submit original papers, within the scope of the JACSM, with the understanding that their contents are unpublished and are not being actively under consideration for publication elsewhere.

For any previously published and copyrighted material, a special permission from the copyright owner is required. This concerns, for instance, figures or tables for which copyright exists. In such a case, it is necessary to mention by the author(s), in the paper, that this material is reprinted with the permission.

Manuscripts, in English, with an abstract and the key words, are to be submitted to the Editorial Office in electronic version (both: source and pdf formats) via e-mail. A decision to accept/revise/reject the manuscript will be sent to the Author along with the recommendations made by at least two referees. Suitability for publications will be assessed on the basis of the relevance of the paper contents, its originality, technical quality, accuracy and language correctness.

Upon acceptance, Authors will transfer copyright of the paper to the publisher, i.e. the Academy of Management, in Lodz, Poland, by sending the paper to the JACSM Editorial Office.

The papers accepted for publication in the JACSM must be typeset by their Authors, according to the requirements concerning final version of the manuscripts. However, minor corrections may have been carried out by the publisher, if necessary.

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm.

The first page of each paper should contain its title, the name of the author(s) with the affiliation(s) and e-mail address(es), and then the abstract and keywords, as show above. Capital letters must be applied in the title of a paper, and 14 pt. boldface font should be used, as in the example at the first page. Use: 11-point type for the name(s) of the author(s), 10-point type for the address(es) and the title of the abstract, 9-points type for the abstract and the key words.

For the main text, please use 11-point type and single-line spacing. We recommend using Times New Roman (TNR) fonts in its normal format. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided. With these sizes, the interline distance should be set so that some 43 lines occur on a full-text page.

The papers should show no printed page numbers; these are allocated by the Editorial Office.

Authors have the right to post their Academy of Management-copyrighted material from JACSM on their own servers without permission, provided that the server displays a prominent notice alerting readers to their obligations with respect to copyrighted material. Posted work has to be the final version as printed, include the copyright notice and all necessary citation details (vol, pp, no, …).

An example of an acceptable notice is:
"This material is presented to ensure timely dissemination of scholarly work, and its personal or educational use is permitted. Copyright and all rights therein are retained by the Copyright holder. Reprinting or re-using it in any form requires permission of the Copyright holder.

The submission format for the final version of the manuscript
and the MS Word template are available on our web site at:
http://acsm.spoleczna.pl

Journal of Applied Computer Science Methods

Contents

Andrzej Cader, Krzysztof Przybyszewski
Application of Deep Neural Networks to Mining
the Macro-Characteristics of the Teaching Process
from the Current Structure of the Student's Grades **5**

Patrycja Pankiewicz
Using The MVC Pattern in the Design
of Computer Games **23**

Krzysztof Przybyszewski, Andrzej Cader
Rules of Knowledge Bases of it Systems Supporting
Education Quality Management in Universities **35**

Kamil Zakrzewski
Automation of Software
Installation in Service Work **59**

Mateusz Sztukowski
Offered Traffic vs Load Dispersion Factor
Dimensioning of Mobile Networks **69**