TAR

The IPSI BgD Transactions on Advanced Research

Multi-, Inter-, and Trans-disciplinary Issues in Computer Science and Engineering

A publication of IPSI Bgd Internet Research Society New York, Frankfurt, Tokyo, Belgrade
January 2019 Volume 15 Number 1 (ISSN 1820-4511)

Special issue: „ICT Research at the University of Belgrade and at its Foreign Guests“

Guest Editors: Nenad Mitić, Faculty of Mathematics;
Zorica Bogdanovic, Faculty of Organizational Sciences;
Drazen Draskovic, School of Electrical Engineering;
University of Belgrade

Table of Contents:

The IPSI BgD Internet Research Society

The Internet Research Society is an association of people with professional interest in the field of the Internet. All members will receive these TRANSACTIONS upon payment of the annual Society membership fee of €500 (air mail printed matters delivery).

Member copies of Transactions are for personal use only

IPSI BGD TRANSACTIONS ON ADVANCED RESEARCH

www.internetjournals.net

Conditional Random Fields-based Approach to Classification: Application to Life Sciences

Grbić, Milana

Abstract: ***Data prediction is one of the most challenging problems in information sciences and has a lot of applications in various fields of science. In the field of biochemistry, predicting the role of metabolite can be of great importance for obtaining additional information about processes within the organisms, as well as for a better understanding of the metabolic pathways. Due to numerous studies based on energy consumption or production, it is useful to determine whether metabolites participate in production or usage of energy in these reactions. i.e. to classify them as energy producers or consumers.***

In this paper Conditional Random Fields (CRF) method is used for structural prediction of roles the metabolites play in metabolic reactions. For prediction of metabolite roles, three different sets of feature functions, which involve information about elements in the nearest neighborhood and information about their labels, are proposed. By using the CRF++ software package, the proposed CRF method is tested on a set of metabolite reactions from the yeast Saccharomyces cerevisiae. In addition, the results of the classification are verified by another CRF implementation (CRFsuite) from the literature. Obtained results indicate a high level of accuracy of the proposed CRF approach.

Index Terms: ***conditional random fields, data classification, prediction, metabolite reactions***

1. *INTRODUCTION*

Task of data prediction is one of the most important problems in information sciences. A lot of applications can be found in various fields of science, like prediction whether a received email is spam or not, prediction if a cancer cell is benign or malignant, prediction of protein disorder. Prediction encompasses various types of classification problems - as specific type of prediction problems - such as text classification, classification of the validity of credit cards etc.

The goal of this research is to predict the role of metabolites in metabolic reaction is considered. Based on this prediction, each metabolite is classified in a particular class, which corresponds to its role in a reaction. Prediction of roles of metabolites is important for further understanding of metabolic pathways. Each metabolic pathway can be considered as a series of chemical reactions which involve reactants, products and various intermediates. There are two basic types of biochemical reactions that are characterized by their presence in either anabolic or catabolic pathways, i.e. pathways with the ability to synthesize molecules with the utilization of energy or to degrade metabolites with releasing energy. Adenosine triphosphate (ATP) appears in both types of pathways either as a reactant (anabolic pathway) or as a product (catabolic pathway). In anabolic pathways, ATP can be degraded to adenosine diphosphate (ADP) or to adenosine monophosphate (AMP), releasing one or two phosphate groups, which further may stay free or can be bound with a metabolite in a phosphorylation process. In catabolic pathways, reverse processes of synthesis ATP from its lower forms ADP and AMP occur. Identifying the role of metabolites in reaction could be useful for finding out more about connections between metabolites based on their involvement in the same reaction. Also, obtained results can be used for a deeper analysis of metabolism of the particular organism.

Conditional Random Fields (CRF) is an approach to prediction, which includes the dependency between variables in the prediction process. In this research a linear CRF is adapted and applied to a list of chemical reactions. Adaptation of existing methods is a common methodological approach to innovation in computer science [1]. The list of chemical reactions is considered as a sequence of sentences. The chosen set of reactions belongs to the particular pathways with the ability to synthesize molecules with the utilization of energy or to degrade metabolites with releasing energy. In the CRF method, the context is taken into account. Since role of a metabolite depends on the neighboring metabolites in the reaction as well as on their labels, this method seems to be convenient for such prediction. We believe that the idea of combination of information about neighboring elements and labels which are input for CRF method, can give the more precise prediction of the role of metabolites. To our knowledge, the problem of predicting the metabolite roles has not been considered in such a way. Still, in the literature one can find several existing applications of CRF method and some similar predictions of biological elements.

The paper is organized as follows: the next section reviews work related to applications of the CRF method and work related to prediction of roles biological elements play in different processes. In Section 3 a description of the proposed CRF model for classification of metabolite roles is provided. Experimental results are shown and discussed in Section 4. The last section is the conclusion.

2. *Related Work*

CRF method is widely used in the field of name entity recognition. The task is identifying the classes of interest to which particular word or phrase belongs. Several recent results are reported in papers [2-5]. WebListing technique based on CRF method, presented in [2], builds seeds for lexicons based on labeled data. These seeds are further significantly augmented by using HTML data on the Web. Different approaches of using CRF based methods for recognizing specific biological terms, such as PROTEIN, DNA, RNA, CELL-LINE and CELL-TYPE in biomedical abstracts are described in [3]. A CRF based open source machine-learning system called BANNER is designed to maximize domain independence, also achieving significantly better performances than other baseline systems [4]. Hybrid model LSTM-CRF, which is combination of Long Short-term Memory Networks (LSTM) and CRF, is also used for named entity recognition [5].

A lot of applications in text preprocessing and analysis make use of CRF method. For example, it is used for shallow parsing [6], for analysis of sentence which first recognizes nouns, verbs, adjectives, etc. and after that groups them to higher order terms such as phrases. Problems such as Part Of Speech (POS) and Chunking can also be solved by methods which are based on CRF [7]. Tree Conditional Random Fields is used for semantic role labeling [8]. In [9], CRF for extraction information from tables is presented. CRF method also can be used for word alignment [10], document summarization [11] and interactive question answering [12].

CRF is also a base for some methods for image segmentation. Discriminative Random Fields (DRF) model, which is based on concept of CRF, has an application in modeling spatial dependency [13]. Shape and texture can be jointly modeled into textons, which are novel features incorporated in CRF method used for image segmentation [14]. Foreground and shadow segmentation can be done using a special class of CRF, named dynamic conditional random fields (DCRF), introduced in [15].

In [16] CRF is used as a replacement of heuristic approaches for stereo vision algorithms. A large number of stereo datasets with ground-truth disparities was constructed, and a subset of these datasets was used to learn the parameters of CRF.

CRF has been intensively used in various fields of bioinformatics. In [17], it is used for tagging gene and protein mentions in text. As it is already mentioned, CRF is a suitable method for name entity recognition in biomedical texts [3]. The first comparative gene predictor based on semi-Markov conditional random fields (SMCRFs) named Conard is presented in [18]. One of the most important problems in bioinformatics is prediction of protein folding. In [19], an effective solution based on segmentation conditional random fields (SCRFs) is proposed. Estimation of parameters used for RNA structural alignment and structural alignment search based on CRF are shown better and more accurate than existing methods [20]. Markov Random Fields, where variables are jointly Gaussian, is called Gaussian CRF (GCRFs). It has an application in computer vision [21]. Extension of the Gaussian CRF, called Directed Gaussian conditional random fields (DirGCRF), is introduced to allow modeling asymmetric relationships [22].

Predicting roles of biological elements in different processes was analyzed in several papers. In [23], the authors considered the role of metabolites in predicting drug-drug interactions. Focus of that research was on irreversible inhibition of cytochrome P450 enzymes and on the metabolites which are fundamental for that inhibition. By using the Bayesian approach and Labelled Multilevel Neighborhoods of Atoms (LMNA) descriptors, the prediction of reacting atom(s) in molecules was carried out in [24]. The method for prediction to which metabolic pathway a particular compound belongs was described in [25]. That method is based on the highest interaction confidence scores. The Random Forest based approach for prediction of metabolic enzymes and gut bacteria was presented in [26].

3. *CRF Method for Predicting Metabolite Roles*

3.1. Problem Definition

Let a list of metabolic reactions be given. Each reaction contains several metabolites, which appear at the left-hand and the right-hand sides of the arrow (see two examples of metabolite reactions in Figure 1). Metabolites from the left-hand side are usually called reactants and the metabolites from the right-hand side of the arrow are products of the reaction. The classification problem can be formulated as follows: given a reaction, assign a label to every metabolite, representing the role the metabolite plays in the reaction. The set of labels are given in advance and in general, it can be based on a particular need in a considered context.

One possible labeling can be based on metabolite's involvement in the energy transfer or in the phosphorylation process. In the approach presented in this paper, the following eight classes are identified:

- Label 1: "*donor of one or two phosphate group(s)"*
- Label 2: *"acceptor of one or two phosphate group(s)"*
- Label 3: *"free phosphate group(s)"*
- Label 4: *"phosphate (compound built by binding one or two phosphate groups(s) with metabolite)"*
- Label 5: *"lower forms of ATP in anabolic pathways"*
- Label 6: *"ATP synthetized in catabolic pathways"*
- Label 7: *"phosphate group which binds in catabolic pathways"*
- Label 8: *"the class containing the rest of metabolites"*

A reaction is considered as a sequence of several elements which are either on its left-hand or on its right-hand side. Therefore, it is important to recognize the arrow as a mark which separates these two sides of the reaction. So, an additional label, Label 9: *"arrow"*, is introduced, only for this purpose.

It should be noted that if some other lists of reactions are considered, the list of labels can be extended, reduced or modified.

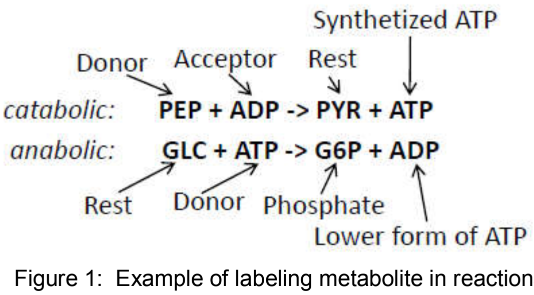


Figure 1: Example of labeling metabolite in reactions

Example 1:

In Figure 1 two reactions are shown. Let us consider the first reaction. This reaction is a part of catabolic pathway. PEP or phosphoenol-pyruvate is an important compound, which contains high energy phosphate bonding [27]. In this reaction PEP is "a donor of one phosphate group". The other reactant is adenosine diphosphate (ADP) which consists of three components: adenine, sugar backbone and two phosphate groups [28]. In this reaction, it is "an acceptor of phosphate group" which is released by PEP. ADP binds that phosphate group and forms adenosine triphosphate (ATP), so the product ATP is labeled as Label 6, i.e. "ATP synthetized in this catabolic pathway". The second reaction from Figure 1 is a part of anabolic pathway. So, ATP is "a donor of one phosphate group" and ADP is "a lower form of ATP". G6P is "a phosphate (compound built by binding one or two phosphate groups(s) with metabolite)" and GLC belongs to "the class of the rest of metabolites".

3.2. *CRF Method for Metabolite Classification*

CRF is a discriminative probabilistic model of machine learning for a structured prediction. Structured prediction refers to supervised machine learning technique that involves predicting structured objects, such as sequence, graph, tree. CRF is introduced in [7] and its main principle can be explained on the problem of labeling a sequence of data. Let T be the length of the sequence. Let $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_T}\}$ represent characteristics of the elements of the sequence, where every $\mathbf{x}_i$ is a vector of characteristics of the element at the position i - x_i. For every element x_i the task is to find the appropriate label y_i based on vector $\mathbf{x}_i$ and neighboring labels. For solving this problem the training set of data $(\mathbf{x}_i, y_i)$ with correct information about labels is provided. The problem of prediction is actually the problem of finding the probability $p(y|\mathbf{x})$, where $y = \{y_1, y_2, \dots, y_T\}$. This probability can be determined in two ways:

i. from the joint probability $p(y, \mathbf{x})$ and the probability $p(\mathbf{x})$ – *generative approach;*
ii. directly, by modeling this conditional probability – *discriminative approach.*

For every generative method there exist a discriminative analogue and vice versa [29]. As it has been already mentioned, the CRF belongs to the class of discriminative methods, i.e. the probability $p(y|\mathbf{x})$ is directly calculated. It is well known that Hidden Markov Models (HMM) is the generative analogue of CRF.

CRF usually imposes the following calculations:

$$p(y|x) = \frac{1}{Z(x)} exp\left\{\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k \, f_k(y_{t-1}\,, y_t\,, \mathbf{x}_t)\right\},$$

where f_k represents feature function and θ_k are the components of parameter vector, for $1 \leq k \leq$

K. The normalization factor is defined by

$$Z(x) = \sum_{y \in Y^T} exp\left\{\sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, \mathbf{x}_t)\right\}.$$

Let us notice that the feature functions $f_k(y_{t-1}, y_t, \mathbf{x}_t)$ use the vector $\mathbf{x}_t$ as an argument, which indicates that all the components of global observations $\mathbf{x}$, that are needed for computing features at the position t, are available. For example, if the next element x_{t+1} is used as a feature for CRF, it is assumed that the information about the identity of that element is included in the vector $\mathbf{x}_t$ [30].

Chemical reactions can be considered as sentences. Although, there are no strict rules for writing reactions, some common conventions still exist. For example, a donor is usually written at the first position or before an acceptor, lower forms of ATP are usually written at the last position or at the position next to the last in the reaction. Therefore, for determining the role of a particular metabolite, it is justified to consider its neighbors and their labels. To examine this hypothesis, three different feature models (named A, B and C), using different sets of feature functions, are constructed.

Model A

In the first model, feature functions are based on the information about the nearest metabolites in the reaction and the information about the label of current element. More precisely, for determining the label of a metabolite at the position t in a reaction, the information about the nearest metabolite as well as the information about nearest consecutive (left and right) pairwise metabolites are taken into account. To formalize, the information about the metabolites at positions from the set $\{t-2,\ t-1,\ t,\ t+1, t+2\ \}$ and information about the label of the current element at the position t are taken into consideration. The following are the feature functions used:

$$f_{x,y}^{-2}\,(y_{t-1}, y_t, \mathbf{x}_t) = I(x_{t-2} = x, y_t = y),$$

$$f_{x,y}^{-1}\,(y_{t-1}, y_t, \mathbf{x}_t) = \ I(x_{t-1} = x, y_t = y\ ),$$

$$f_{x,y}^{0}\,(y_{t-1}, y_t, \mathbf{x}_t) = \ I(x_t = x, y_t = y),$$

$$f_{x,y}^{+1}\,(y_{t-1}, y_t, \mathbf{x}_t) = \ I(x_{t+1} = x, y_t = y),$$

$$f_{x,y}^{+2}\,(y_{t-1}, y_t, \mathbf{x}_t) = I(x_{t+2} = x, y_t = y).$$

$$f_{x,x',y}^{-1,0}\,(y_{t-1}, y_t, \mathbf{x}_t) = \ I(x_{t-1} = x,\ x_t = x', y_t = y),$$

$$f_{x,x',y}^{0,+1}\,(y_{t-1}, y_t, \mathbf{x}_t) = I(x_t = x,\ x_{t+1} = x', y_t = y).$$

Let us explain the notation of these functions. In all formulas, I is the indicator function, i.e. it is equal to 1 if the condition is valid, otherwise 0. In the superscript of a function f, the mark $-i$, (respectively $+i$) $i \in \{1,2\}$, means that the information about the metabolite, which is on the distance i to the left (respectively to the right) from the considered one, is taken into account. The mark 0 means that the information about the current element is considered.

Thus the set of feature functions used in Model A is

$$F_A = \left\{f_{x,y}^{-2}, f_{x,y}^{-1}, f_{x,y}^{0,}, f_{x,y}^{+1}, f_{x,y}^{+2}, f_{x,x',y}^{-1,0}, f_{x,x',y}^{0,+1} | x, x' \in X, y \in Y\right\},$$

where X is the set of all elements which are present in given reactions and Y is the set of all possible labels.

Model B

The first five functions from the model A are also used in the model B. This subset of feature functions is extended by the following functions, which use information about the label of the previous metabolite in the reaction and information about the nearest metabolites:

$$g_{x\prime,x,y',y}^{-1,0}\,(y_{t-1}, y_t, \mathbf{x}_t) = I(x_{t-1} = x', x_t = x,\ y_{t-1} = y', y_t = y\ ),$$

$$g_{x\prime,x,y',y}^{0,+1}\,(y_{t-1}, y_t, \mathbf{x}_t) = I(x_t = x', x_{t+1} = x, y_{t-1} = y',\ y_t = y),$$

$$g_{x'',x',x,y\prime,y}^{-2,-1,0}(y_{t-1}, y_t, \mathbf{x}_t) = I(x_{t-2} = x'', x_{t-1} = x', x_t = x, y_{t-1} = y', y_t = y),$$

$$g_{x'',x',x,y',y}^{-1,0,+1}(y_{t-1}, y_t, \mathbf{x}_t) = I(x_{t-1} = x'', x_t = x', x_{t+1} = x, y_{t-1} = y', y_t = y),$$

$$g_{x'',x',x,y',y}^{0,+1,+2}(y_{t-1,}y_{t,}\mathbf{x}_t) = I(x_t = x'', x_{t+1} = x', x_{t+2} = x, y_{t-1} = y', y_t = y).$$

So, the set of feature functions in Model B is

$$F_B = \left\{f_{x,y}^{-2}, f_{x,y}^{-1}, f_{x,y}^{0,}, f_{x,y}^{+1}, f_{x,y}^{+2}, g_{x',x,y',y}^{-1,0},\ g_{x',x,y',y}^{0,+1}, g_{x'',x',x,y',y}^{-2,-1,0},\ g_{x'',x',x,y',y}^{-1,0,+1},\ g_{x'',x',x,y',y}^{0,+1,+2} | x, x' \in X, y, y' \in Y\right\}$$

The function $g_{x'',x',x,y}^{-2,-1,0}$ takes into account information about the metabolites at positions $t-2$, $t-1$ and t. Similarly, the function $g_{x'',x',x,y}^{0,+1,+2}$ records information about metabolites at positions t, $t+1$ and $t+2$, while the function $g_{x'',x',x,y}^{-1,0,+1}$ records information about metabolites at positions $t-1$, t and $t+1$. All functions take into account the information about the labels of the previous element and the current one.

Model C

The set of feature function in the model C contains only functions which take into account information about the label of the previous element. This set can be derived from the set F_B

by omitting the functions from the model A, so

$$F_C = F_B \setminus \{ f_{x,y}^{-2}, f_{x,y}^{-1}, f_{x,y}^{0}, f_{x,y}^{+1}, f_{x,y}^{+2} | x \in X, y \in Y \}.$$

Example 2:

In order to illustrate how the proposed feature functions are calculated, let us consider the reaction:

ATP + AC + COA → AMP + PPI + ACCOA.

The reactants and products are labeled as follows:

1. ATP – Label 1: "*donor of one or two phosphate group(s)"*
2. AC – Label 8: *"the class containing the rest of metabolites"*
3. COA – Label 8: *"the class containing the rest of metabolites"*
4. → - Label 9: *"arrow"*
5. AMP – Label 5: *"lower forms of ATP in anabolic pathways"*
6. PPI – Label 3: *"free phosphate group(s)"*
7. ACCOA – Label 8: *"the class containing the rest of metabolites"*

Let each metabolite be assigned its position in the reaction. Let us, for example, take $t = 3$, (x_3 = COA) and then the values of considered functions are

$$f_{x,y}^{-2} = \begin{cases} 1, x = \text{ATP and } \ y = \text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$f_{x,y}^{-1} = \begin{cases} 1, x = AC \text{ and } y = Label\ 8 \\ 0\ , \text{otherwise} \end{cases}$$

$$f_{x,y}^{0} = \begin{cases} 1, x = \text{COA and } y = \text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$f_{x,y}^{+1} = \begin{cases} 1, x = \rightarrow \text{ and } y = \text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$f_{x,y}^{+2} = \begin{cases} 1, x = \text{AMP and } y = \text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$f_{x,x',y}^{-1,0} = \begin{cases} 1, x = \text{AC}, x' = \text{COA and } y = \text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$f_{x,x',y}^{0,1} = \begin{cases} 1, x = \text{COA}, x' = \rightarrow \text{ and } y = \text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$g_{x',x,y',y}^{-1,0} = \begin{cases} 1, x'{=}\text{AC}, x{=}\text{COA}, \text{y}'{=}\text{Label 8 and } y{=}\text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$g_{x',x,y',y}^{0,+1} = \begin{cases} 1, x'{=}\text{COA}, x{=} \rightarrow, \ \text{y}'{=}\text{Label 8 and } y{=}\text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$g_{x'',x',x,,y',y}^{-2,-1,0} = \begin{cases} 1, x''{=}\text{ATP}, x'{=}\text{AC}, x{=}\text{COA}, \text{y}'{=}\text{Label 8 and } y{=}\text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$g_{x'',x',x,y',y}^{-1,0,1} = \begin{cases} 1, x''{=}\text{AC}, x'{=}\text{COA}, x{=} \rightarrow, \text{y}'{=}\text{Label 8 and } y{=}\text{Label 8} \\ 0, \text{otherwise} \end{cases}$$

$$g_{x'',x',x,y',y}^{0,1,2} = \begin{cases} 1, x''{=}\text{COA}, x'{=} \rightarrow, x{=}\text{AMP}, \text{y}'{=}\text{Label 8 and } y{=}\text{Label 8} \\ 0, \text{otherwise} \end{cases}.$$

The feature functions for other metabolites can be analyzed in a similar way.

```
# Template for the model A

U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

# Template for the model B

U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]

B01:%x[-1,0]/%x[0,0]
B02:%x[0,0]/%x[1,0]

B10:%x[-2,0]/%x[-1,0]/%x[0,0]
B11:%x[-1,0]/%x[0,0]/%x[1,0]
B12:%x[0,0]/%x[1,0]/%x[2,0]


# Template for the model C

B01:%x[-1,0]/%x[0,0]
B02:%x[0,0]/%x[1,0]

B10:%x[-2,0]/%x[-1,0]/%x[0,0]
B11:%x[-1,0]/%x[0,0]/%x[1,0]
B12:%x[0,0]/%x[1,0]/%x[2,0]
```

Figure 2 : An excerpt from the template file

4. Experimental Results and Discussion

This section presents experimental results obtained by the proposed CRF models. All tests are carried out on the Intel i5 @2.5 GHz with 8 GB RAM. In order to implement all proposed models (A, B and C), the software package CRF++ [31] is used. CRF++ allows user-defined templates, so it is suitable for an approach which includes custom feature functions. In this package, calculation of the CRF model is implemented as a forward-backward algorithm and a logarithmic computation is used [32].

In addition, CRF++ allows customization of two other parameters:

- parameter *c*, which is used for obtaining balance between overfitting and underfitting. The default value is set to 1. In our tests. two additional values, 1.5 and 2, were also used.
- parameter *f*, which is an integer number that sets the cut-off threshold for the features. Only features whose number of occurrences is equal or greater than *f* are taken into account. The default value is set to 1. Two more values, 2 and 3, were used in the tests.

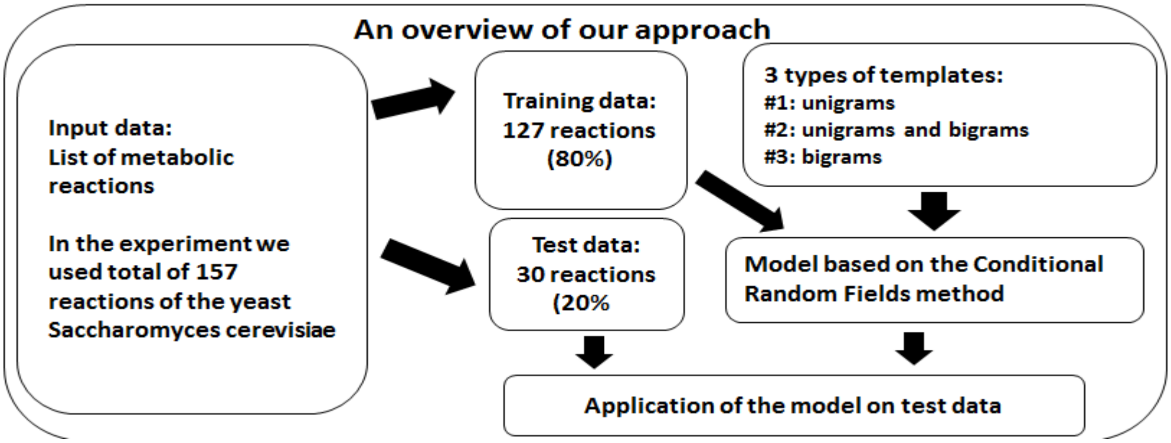

An overview of our approach
Input data:
List of metabolic reactions
In the experiment we used total of 157 reactions of the yeast Saccharomyces cerevisiae
Training data:
127 reactions
(80%)
Test data:
30 reactions
(20%
3 types of templates:
#1: unigrams
#2: unigrams and bigrams
#3: bigrams
Model based on the Conditional Random Fields method
Application of the model on test data

Figure 3: Overall approach

Dataset used for the tests contains biological information about the metabolism of the yeast Saccharomyces cerevisiae. This dataset consists of a list of 157 metabolic reactions, taken from [33]. These reactions are chosen from the list of all metabolic reactions of this organism - as the reactions which are the part of the energy transfer or the part of the phosphorylation process.

The tests were performed in four phases. In the first phase, the set of the chosen reactions was divided into training data and test data, in ratio 80:20. Templates were formed in the second phase, taking into account the information about the current metabolite in the reaction and information about the other metabolites from the same reaction. As it is described in Subsection 3.2, three different templates were used, corresponding to the models A, B and C. The templates whose name starts with ‘U’ are called unigrams and they take into account information about label of the current element. The templates with the mark ‘B’ (bigrams) use the information about labels of the previous and of the current element. An excerpt of the template file is shown in Figure 2. After the templates were formed, the algorithm entered into the third phase, i.e. the construction of the CRF model, based on the training data and the template file. And finally, in the fourth phase, the model was applied on the test data. Graphical representation of the overall approach is shown in Figure 3.

For a deeper analysis of the proposed CRF models, in a series of tests, the following combinations of control parameters were used:

- default combination, in which *f=1* and *c=1*,
- *f=3* and *c=1.5*,
- *f=2* and *c=1.5*,
- *f=3* and *c=2*,
- *f=2* and *c=2*.

Obtained results are shown in Table 1. In the first column, the name of the model is given. In the rest of the table, for each combination of control parameters and for each model, the accuracy obtained on the test data is presented. The accuracy is calculated in the standard way as ratio between number of correctly classified elements and total number of elements.

From Table 1, it can be seen that models A and B are more accurate than the model C and results of the model A are slightly better than results of the model B. For the model A, the difference between accuracy of obtained results for different parameter combinations is very small, which indicates that the model A is the most stable one. If we consider all three models together, the default parameter combination (last column) gives the best results on average.

Table 1: Experimental results obtained by CRF++

Parameters	*f=3, c=1.5*	*f=2, c=1.5*	*f=3, c=2*	*f=2, c=2*	*Default* *f=1, c=1*
Model					
Model A	93.60465%	94.18605 %	94.18605%	94.18605%	93.02326%
Model B	93.60465%	93.60465%	93.60465%	93.60465%	91.86047%
Model C	66.86047%	68.02326%	66.86047%	68.02326%	77.32558%

Table 2: Performances of the model A with control parameters *f=3* and *c=2*

Class	#match	#model	#ref	precision	recall	F1
Label 1	30	31	30	0.967742	1	0.9836066
Label 2	2	2	2	1	1	1
Label 3	11	11	12	1	0.917	0.9565217
Label 4	11	12	17	0.916667	0.647	0.7586207
Label 5	27	28	27	0.964286	1	0.9818182
Label 6	2	2	2	1	1	1
Label 7	1	1	1	1	1	1
Label 8	47	54	50	0.87037	0.94	0.9038462
Label 9	31	31	31	1	1	1
			average	0.968785	0.945	0.9538237

To validate the quality of the proposed models, the same dataset was adapted and tested by another CRF software - CRFsuite. This software reads training data and automatically generates all the necessary state and transition features based on the data [34]. The accuracy obtained by CRFsuite is 94.76% which is very similar to the best results obtained by the proposed model A.

In order to further investigate performance of the proposed models, a comparative analysis of the results obtained for each of the nine label classes, by both CRF packages, was performed. In the case of CRF++, the most successful model and parameter combination, i.e. the model A with control parameters *f=3, c=2* is chosen. For each of 9 classes, three different measures were calculated:

- precision, calculated as

$$\text{precision} = \frac{\#\text{match}}{\#\text{model}}$$

- recall, calculated as

$$\text{recall} = \frac{\#\text{match}}{\#\text{ref}}$$

- F1 measure, calculated as

$$\text{F1} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

where #match, #model and #ref are the total number of matched labels, the total number of labels in the model and the total number of labels in the test set, respectively.

Results for each label, obtained by the chosen model A tested by CRF++ and the CRFsuite model, are shown in details in Tables 2 and 3, respectively. In both tables data are organized as follows. The first column contains the label of the class. In the next three columns #match, #model and #ref are shown. Last three columns contain information about three measures: precision, recall and F1 measure. In the last row average values for these three measures are shown.

From Tables 2 and 3 it can be seen that both models achieve accurate results for all labels. For each measure, on average, the model A is slightly better than the CRFsuite model. Still, the results shown in these tables indicate that both models perform similarly on the considered data set.

5. Conclusion

Classification of roles of metabolites can be of great importance for further understanding of metabolic process in different organisms. The proposed approach to classification is based on the conditional random field method. Metabolic reactions were considered as sequences of elements, enabling the construction of different feature functions, based on unigrams and/or bigrams. In the paper, three different models were proposed for constructing templates, which are further used for CRF model construction.

Performances of the proposed models were tested on a real set of biological data. The obtained results indicate high accuracy of the models. The detailed analysis in previous section show that the proposed method found appropriate labels for the most of the metabolites, so it indicates that it can be used for solving the considered problem.

Future work may include further application of the proposed models, especially to larger data sets. Another promising extension of this research may involve development of other feature functions containing additional information about characteristics of metabolites.

Table 3: Performances of the CRFsuite software model

	#match	#model	#ref	precision	recall	F1
Label 1	29	30	30	0.966667	0.966667	0.966667
Label 2	2	2	2	1	1	1
Label 3	10	11	12	0.909091	0.833333	0.869565
Label 4	16	20	17	0.8	0.941176	0.864865
Label 5	27	29	27	0.931034	1	0.964286
Label 6	1	1	2	1	0.5	0.666667
Label 7	1	1	1	1	1	1
Label 8	46	47	50	0.978723	0.92	0.948454
Label 9	31	31	31	1	1	1
			average	0.953946	0.906797	0.920056

REFERENCES

[1] Blagojević, V., Bojić, D., Bojović, M., Cvetanović, M., Đorđević, J., Đurđević, Đ., ... & Milutinović, V. (2017). A systematic approach to generation of new ideas for PhD research in computing. In Advances in computers (Vol. 104, pp. 1-31). Elsevier.

[2] McCallum, A., & Li, W. (2003, May). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 188-191). Association for Computational Linguistics.

[3] Settles, B. (2004, August). Biomedical named entity recognition using conditional random fields and rich feature sets. In Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (pp. 104-107). Association for Computational Linguistics

[4] Leaman, R., & Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. In Biocomputing 2008 (pp. 652-663).

[5] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

[6] Sha, F., & Pereira, F. (2003, May). Shallow parsing with conditional random fields. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 134-141). Association for Computational Linguistics.

[7] Lafferty, J., McCallum, A. and Pereira, F. CN "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

[8] Cohn, T., & Blunsom, P. (2005, June). Semantic role labelling with tree conditional random fields. In Proceedings of the Ninth Conference on Computational Natural Language Learning (pp. 169-172). Association for Computational Linguistics

[9] Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003, July). Table extraction using conditional random fields. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 235-242). ACM.

[10] Blunsom, P., & Cohn, T. (2006, July). Discriminative word alignment with conditional random fields. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (pp. 65-72). Association for Computational Linguistics.

[11] Shen, D., Sun, J. T., Li, H., Yang, Q., & Chen, Z. (2007, January). Document Summarization Using Conditional Random Fields. In IJCAI (Vol. 7, pp. 2862-2867).

[12] Hickl, A., & Harabagiu, S. (2006, June). Enhanced interactive question-answering with conditional random fields. In Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006 (pp. 25-32). Association for Computational Linguistics.

[13] Kumar, S., & Hebert, M. (2006). Discriminative random fields. International Journal of Computer Vision, 68(2), 179-201.

[14] Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006, May). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In European conference on computer vision (pp. 1-15). Springer, Berlin, Heidelberg.

[15] Wang, Y., Loe, K. F., & Wu, J. K. (2006). A dynamic conditional random field model for foreground and shadow segmentation. IEEE transactions on pattern analysis and machine intelligence, 28(2), 279-289.

[16] Scharstein, D., & Pal, C. (2007, June). Learning conditional random fields for stereo. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (pp. 1-8). IEEE.

[17] McDonald, R., & Pereira, F. (2005). Identifying gene and protein mentions in text using conditional random fields. BMC bioinformatics, 6(1), S6.

[18] DeCaprio, D., Vinson, J. P., Pearson, M. D., Montgomery, P., Doherty, M., & Galagan, J. E. (2007). Conrad: gene prediction using conditional random fields. Genome research, 17(9), 000-000.

[19] Liu, Y., Carbonell, J., Weigele, P., & Gopalakrishnan, V. (2006). Protein fold recognition using segmentation

conditional random fields (SCRFs). Journal of Computational Biology, 13(2), 394-406.

[20] Sato, K., & Sakakibara, Y. (2005). RNA secondary structural alignment with conditional random fields. Bioinformatics, 21(suppl_2), ii237-ii242.

[21] Tappen, M. F., Liu, C., Adelson, E. H., & Freeman, W. T. (2007, June). Learning gaussian conditional random fields for low-level vision. In 2007 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[22] Vujicic, T., Glass, J., Zhou, F., & Obradovic, Z. (2017). Gaussian conditional random fields extended for directed graphs. Machine Learning, 106(9-10), 1271-1288.

[23] VandenBrink, B. M., & Isoherranen, N. (2010). The role of metabolites in predicting drug-drug interactions: Focus on irreversible P450 inhibition. Current opinion in drug discovery & development, 13(1), 66.

[24] Rudik, A. V., Dmitriev, A. V., Lagunin, A. A., Filimonov, D. A., & Poroikov, V. V. (2016). Prediction of reacting atoms for the major biotransformation reactions of organic xenobiotics. Journal of cheminformatics, 8(1), 68.

[25] Hu, L. L., Chen, C., Huang, T., Cai, Y. D., & Chou, K. C. (2011). Predicting biological functions of compounds based on chemical-chemical interactions. PloS one, 6(12), e29491.

[26] Sharma, A. K., Jaiswal, S. K., Chaudhary, N., & Sharma, V. K. (2017). A novel approach for the prediction of species-specific biotransformation of xenobiotic/drug molecules by the human gut microbiota. Scientific reports, 7(1), 9751.

‘

[27] Berg, J. M., Tymoczko, J. L., & Stryer, L. (2002). Biochemistry, ; W. H.

[28] Nelson, D. L., Lehninger, A. L., & Cox, M. M. (2008). Lehninger principles of biochemistry. Macmillan.

[29] Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., & West, M. (2007). Generative or discriminative? getting the best of both worlds. Bayesian statistics, 8(3), 3-24.

[30] Sutton, C., & McCallum, A. (2012). An introduction to conditional random fields. Foundations and Trends® in Machine Learning, 4(4), 267-373.

[31] CRF++: Yet another CRF toolkit (2005)

[32] Sokolovska, N., Lavergne, T., Cappé, O., & Yvon, F. (2010). Efficient learning of sparse conditional random fields for supervised sequence labeling. IEEE Journal of Selected Topics in Signal Processing, 4(6), 953-964.

[33] Förster, J., Famili, I., Fu, P., Palsson, B. Ø., & Nielsen, J. (2003). Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. Genome research, 13(2), 244-253.

[34] N. Okazaki. CRFsuite: A fast implementation of conditional random fields (CRFs), 2007.

Milana Grbić received her MSc degree (2016) in Mathematics at the Faculty of Mathematics at the University of Belgrade. She is a PhD student in the field of data mining in bioinformatics. Her research interests include data classification, data mining and bioinformatics.

Maxeler's MaxWare on AWS with example applications

Milankovic, Ivan and Trifunovic, Nemanja

Abstract: ***In a last few years the use of cloud services constantly continues to rise. One of the leading cloud services providers is Amazon Web Services (AWS) that offers a variety of services. Recently Amazon announced Elastic Compute Cloud (EC2) F1 instance that is equipped with Xilinx Virtex UltraScale+ Field Programmable Gate Array (FPGA). Users can configure FPGA on F1 instance in order to create custom hardware accelerations for their application. The F1 instance is intended for solving complex science, engineering, and business problems that require high bandwidth, enhanced networking, and very high compute capabilities. It is particularly beneficial for applications that are time sensitive such as clinical genomics, financial analytics, video processing, big data, security, and machine learning. During last 15 years Maxeler has been developing environment for their Dataflow Engines (DFEs). With announcement of Amazon EC2 F1 instance Maxeler has adapted their MaxWare so that users can use MaxCompiler to design their applications and MaxelerOS to run their applications on Amazon EC2 F1 instance. In this paper we are going to give brief introduction to Amazon EC2 F1 instance and explain how users could use Maxeler's MaxWare on AWS to easily develop applications for F1 instances. We are also going to present two Maxeler's applications that can be found as AWS SaaS contracts on Amazon Marketplace - Real Time Risk (RTR) Dashboard and Quantum Chromodynamics (QCD) and we are going to explain how they could be used and how big performance boost they could achieve.***

Index Terms: Amazon Web Services, Dataflow Engines, Field Programmable Gate Array, F1 Instance, MaxCompiler, MaxelerOS, MaxJ, Xilinx Virtex UltraScale+

1. *Introduction*

THE use of cloud computing in a recent several years constantly continues to rise. Cloud computing provides a simple way to access servers, storage, databases and a broad set of application services over the Internet in order to offer faster innovation, flexible resources and economies of scale. One of the most popular cloud services platforms is Amazon Web Services (AWS).

The AWS began offering IT infrastructure services in 2006. AWS was one of the first companies to introduce a pay-as-you-go cloud computing model. That model was able to provide users with compute, storage or throughput as needed. Today, AWS offers more than 90 services spanning a wide range including computing, storage, networking, database, analytics, application services, deployment, management and that number is growing rapidly as well.

One of the most popular AWS services is Amazon Elastic Compute Cloud (EC2). The EC2 represents the web service which provides secure and resizable compute capacity in the cloud. The virtual machines on AWS EC2 are called instances. Those EC2 instances are highly scalable. There are many different types of EC2 instances with more or less RAM, CPU and the user can easily choose which one to launch, start or stop. The EC2 instance can run against different types of operating systems depending which Amazon Machine Image (AMI) user chose. The AMI represents special type of virtual appliance which is used to launch a virtual machine within the AWS EC2. User can use AWS Console to configure, start, stop instances from web browser. AWS also provides different interfaces so that user can control instances form several programing languages by using appropriate APIs. User can also create his own AMI and sell it on Amazon Marketplace. Beside AMI user can sell Software as a Service (SaaS) on Amazon Marketplace.

Recently AWS introduced new EC2 instance type called F1. The F1 instance represents compute instance with Field Programmable Gate Arrays (FPGA). It is offered in three different instance sizes that include up to eight FPGAs per instance. It is equipped with 16 nm Xilinx Virtex UltraScale+ VU9P FPGA. Each FPGA contains approximately 2.5 million logic elements and approximately 6,800 Digital Signal Processing

Manuscript received November 30, 2018. This work was supported in part by the Maxeler Technologies.

Ivan Milankovic is with the Maxeler Technologies (e-mail: imilankovic@maxeler.com).

Nemanja Trifunovic (contact person) is with the Maxeler Technologies and School of Electrical Engineering University of Belgrade (e-mail: nemanja@maxeler.com).

(DSP) engines.

Users can configure FPGA on F1 instance in order to create custom hardware accelerations for their application. The F1 instance is intended for solving complex science, engineering, and business problems that require high bandwidth, enhanced networking, and very high compute capabilities. It is particularly beneficial for applications that are time sensitive such as clinical genomics, financial analytics, video processing, big data, security, and machine learning.

Even though EC2 F1 instance has recently introduced there are already several research papers which are using it. Marco Rabozzi et al. [1] proposed an implementation to solve the 5-point relative pose problem accelerated on FPGA. They proposed architecture which implements the classical Nister's algorithm as a deep pipeline deployed on a AWS F1 instance. The results showed that they outperformed the software implementation by a factor ranging from 7.2X to 233X and they achieved a speedup of 64.2X compared to the Nister's software implementation with comparable accuracy. Another research [2] used AWS F1 instance to formalize approach to understand the scalability of the existing hardware architecture with cost models and neural network performance prediction as a function of the target device size. Ioannis Stamelos et al. [3] presented a novel framework that allows the seamless integration of FPGAs from high-level programming languages, like Java and Scala. The proposed approach provided all required APIs for the utilization of FPGAs from these languages. The proposed scheme has been mapped on AWS EC2 F1 instance and a performance evaluation is presented for two widely used machine learning algorithms.

In order to create bitstream for EC2 F1 instance FPGA users have to use FPGA Developer AMI and to write their code in some Hardware Description Language (HDL) which requires from users to have significant expertise in hardware design. Over more than last 15 years Maxeler has pushed their effort to develop environment for their Dataflow Engines (DFEs). Users can now use Maxeler's MaxJ language to design their applications. MaxJ language is a Java based language which allows users without previous MaxJ experience and without significant expertise in hardware design to write high-performance applications for FPGA with a higher level of abstraction from hardware than a HDL. Voss et al. [4] showed on the gzip design example that using MaxJ takes only one person and a period of one month to develop an application and achieve better performance than the related work created in Verilog and OpenCL. Once users write their MaxJ program they can use MaxCompiler to build .max file which is used to reconfigure DFEs. Also, there is MaxelerOS software interface which provides the low-level interface between the application software and the DFE at run-time. Maxeler has also provided simulation and debugging tools which allow designs to be tested before building for a real DFE which provides much faster development than with using low level hardware description languages such as Verilog and VHDL.

Currently, the only way to write design for EC2 F1 instance is to use FPGA Developer AMI and to write code in some HDL. The main problem there is that users need to have good hardware background. In this paper we propose solution to that problem by adapting Maxeler’s MaxWare to EC2 F1 instance. In that way the users will be able to use MaxCompiler to design their applications and MaxelerOS to run their applications on Amazon EC2 F1 instance. In this paper we are going to explain how users could use Maxeler's MaxWare on AWS to easily develop applications for EC2 F1 instance. We are also going to present two Maxeler's applications which are executing on EC2 F1 instance and which could be found as AWS SaaS contracts on Amazon Marketplace - Real Time Risk (RTR) Dashboard [5] and Quantum Chromodynamics (QCD) Dashboard [6]. By searching through the literature and Amazon Marketplace we didn’t find any similar applications.

2. *Background*

Nowadays, most of the computers are based on von Neumann architecture which is also known as control flow architecture. The control flow architecture works by transforming the program source into a list of instructions and loads those instructions into the memory. The Central Processing Unit (CPU) reads those instructions from the memory, runs operations specified by those instructions and writes the results back to the memory. As it can be noticed, execution of each instruction requires cyclic access for memory. This results in a large number of transfers between CPU and memory which represents the main drawback of control flow architecture. The dataflow computing represents a computing paradigm which doesn’t require cyclic access to the memory. The central part of each dataflow engine is FPGA.

2.1 Field Programmable Gate Arrays

Field programmable gate arrays represent integrated circuits designed so that they can be configured by the designer to implement some computations. They represent an array of Configurable Logic Blocks (CLBs) connected via programmable interconnects. The ability to reconfigure FPGAs is the feature which distinguishes them from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks. The main goal of FPGA is to accelerate certain calculation tasks. They also have low processing power consumption. FPGAs achieve the best results with algorithms with low data dependencies and high ability for parallel execution.

In order to configure the FPGA designers need to generate the bitstream which describes the hardware. For that purpose designers usually use some of the Hardware Description Languages (HDLs) and the two most common are VHDL and Verilog. At the first look the code written in HDL and in a high-level software programming language can be similar but those two are fundamentally different. Software code specifies a sequence of operations. On the other hand HDL code can be understood like a schematic that uses text to introduce components and describe their interconnections.

Modern FPGAs have a large number of logic blocks, I/O pads, and routing channels. Since the FPGAs are limited with the amount of resources in the process of designing an application it is required to pay special attention on the available amount of resources. It is not that difficult to determine the number of used logic blocks and I/O pads from the design, but the number of routing channels may vary significantly. In theory, it should be possible to map any algorithm into FPGA, but, in practice, the main constraints are available resources, clock rates, and available I/O pads.

2.2 Dataflow Computing

The dataflow computing paradigm is fundamentally different from the control flow computing. In a Dataflow application as shown on Fig. 1, the program source is transformed into a Dataflow engine configuration file, which describes the operations, layout and connections of a Dataflow engine. Data can be streamed from memory into the chip where operations are performed and data is forwarded directly from one computational unit (“dataflow core”) to another, as the results are needed, without being written to the off-chip memory until the chain of processing is complete.

The main advantage of dataflow computing is that the instructions are executed in a natural sequence as data flows through the algorithm. It also reduces the effect of memory access latency since all of the data flows through the graph. Dataflow computing not only accelerates calculations, but also makes them more energy efficient compared to control flow computing. This is achieved with the low DFE's frequency which can go up to a few hundreds of MHz, while frequency of nowadays processors goes up to a few GHz and it is a well-known fact that power consumption is directly proportional to frequency.

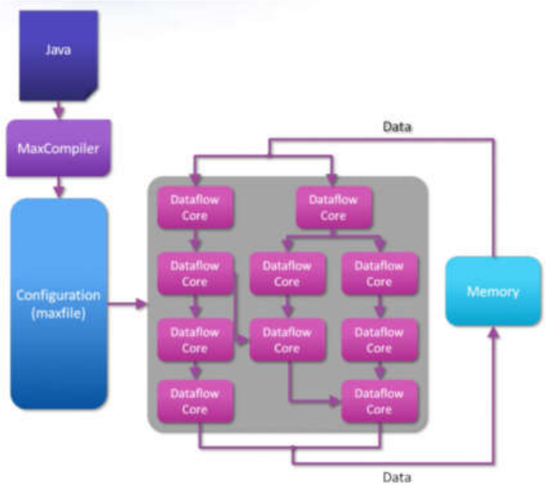


Fig. 1. Dataflow computing

Dataflow computing is nowadays used in a large number of applications. They are used in automation applications [7], digital signal processing [8, 9], artificial neural networks [10]. They are also used a lot in mathematics for solving many problems such as systems of equations [11], floating-point matrix multiplication [12], and much more.

2.3 Maxeler's Dataflow Engines

During the last 15 years Maxeler was working on developing their DFEs which use FPGA chip as a computational unit. The general architecture of these DEFs is shown in Fig. 2. Each DFE consists of one or more kernels which perform computation as data flows from CPU through the DFE. There are two types of memory which DFE has: FMem (Fast Memory) and LMem (Large Memory). FMem has a smaller capacity than the LMem. FMem can store several megabytes of data, while LMem can store many gigabytes of data. On the other hand FMem with terabytes/second of access bandwidth is much faster than the LMem. One of the main reasons why DFEs can achieve such high performances are the bandwidth and flexibility of FMem.

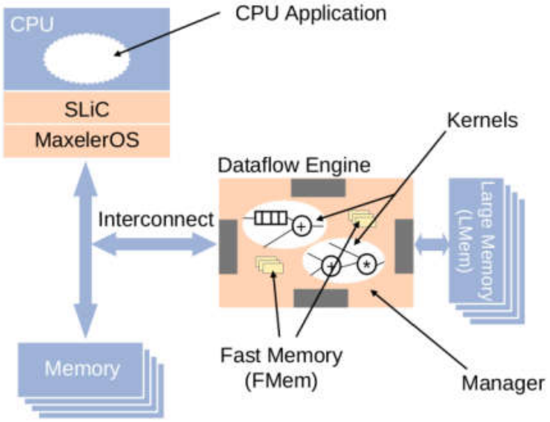


Fig. 2. DFE architecture

The dataflow engine is programmed with one or more Kernels and a Manager. Kernels represent the part of the DFE which is responsible for computation while the Manager has a task to define data movement within the DFE. Once the user builds his design, the MaxCompiler generates MAX file which is used to configure DFE and which allows that actions can be called from the CPU via the SLiC interface. The SLiC (Simple Live CPU) interface is automatically generated. The overall system is managed by MaxelerOS which is responsible for data transfer and dynamic optimization at runtime.

Maxeler's DFEs are subject of many researches and are used in many fields. Veljovic [13] used example of moving from bi adders to three adders to described discrepancy reduction between the topology of dataflow graph and the topology of FPGA structure. Voros et al. [14] managed to accelerate the key search algorithm DFEs up to 205 fold. Pell et al. [15] used DFEs for finite difference wave propagation modeling and achieved up to 30 times more energy efficient solution than with conventional CPUs. Oriato et al. [16] used DFEs for the meteorological limited area model and increased execution speed up to 74 fold compared to x86 CPU. Weston et al. [17] accelerated derivatives computations over 270 times compared to a Intel Core for a multi-asset Monte Carlo model. Some studies [18, 19] showed the ability of using DFEs in biomedical images processing and showed really good results. Gan et al. [20] used Maxeler's DFEs to find the solution of global shallow water equations (SWEs) and managed to achieve speedup of 20 over a fully optimized design on a CPU rack with two eight-core processors and speedup of 8 over the fully optimized Kepler GPU design. Another study [21] proposed novel benchmarking methodology which showed that dataflow systems outperform control flow systems. Maxeler has also developed AppGallery website [22] where we can already find more than 50 applications and where researches can publish their own applications.

3. *Materials and Methods*

With announcement of AWS EC2 F1 instance, Maxeler, who is AWS partner in AWS EC2 F1 Instance project, started to adapt MaxWare so that it gets fully functional for AWS EC2 F1 instance. Currently, as presented in Fig. 3, MaxWare on AWS is composed of three AMIs: MaxSim, MaxCompiler and MaxelerOS.

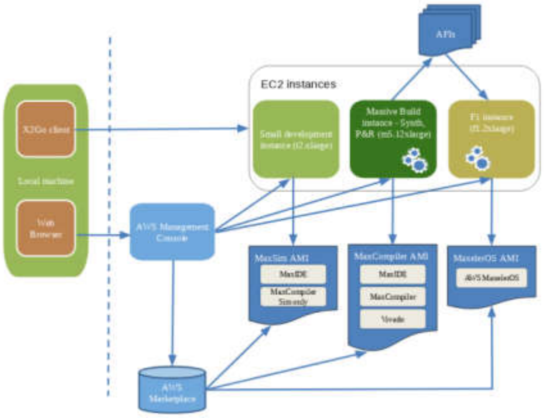


Fig. 3. MaxWare on AWS

MaxSim AMI contains MaxIDE and MaxCompiler Simulator. Recommended AWS EC2 instance for this AMI is t2.xlarge. This AMI should be used for developing application and testing the correctness of the output results. For testing purposes input data set should be reduced to a minimal input data set. The main advantage of this AMI is that the build time for simulation is much smaller than the build time for real EC2 F1 instance and MaxCompiler Simulator is 100 times faster than the HDL simulation.

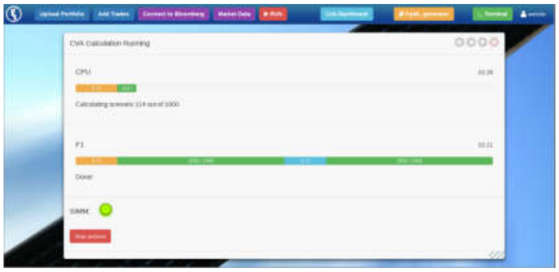
Once we made our design and it passes all tests against MaxSim AMI we should try to build application for EC2 F1 instance. For that purpose we need to use MaxCompiler AMI. The MaxCompiler AMI contains MaxIDE, MaxCompiler and Vivado and the result of its build is an Amazon FPGA Image (AFI). We can understand AFI as bitstream which we use to reconfigure FPGA attached to the EC2 F1 instance. The AFI lives somewhere in AWS cloud and by default it is private, but we can share it with another AWS account, make it public or connect it to some AMI and sell it on Amazon Marketplace.

Once we have generated an AFI and built the binary from our CPU code we can try to run it on EC2 F1 instance. For that purpose we need to use MaxlerOS AMI. The MaxelerOS AMI has installed all dependencies required for successfully running application against EC2 F1 instance. It automatically fetches the AFI, reconfigures FPGA attached to the EC2 F1 instance and streams data between CPU and FPGA.

Most of the applications from Maxeler's AppGallery should be able to run on EC2 F1 instance. Two example applications which are offered on Amazon Marketplace as SaaS contracts and which are running on EC2 F1 instance are Real Time Risk (RTR) Dashboard and Quantum Chromodynamics (QCD).

3.1 *Real Time Risk (RTR) Dashboard*

Maxeler Real Time Risk (Maxeler RTR) is a suite of Finance Risk tools and components, including Credit Value Adjustment (CVA), Margin Requirements (ISDA SIMM and CME Clearing) but also a full derivatives pricing library, driven by Bloomberg market data and the customer trades in FPML format. Maxeler RTR can run on CPU cloud instances or, for ultrafast real time purposes, on AWS EC2 F1 Instances.



CVA Calculation Running

Calculating scenario 114 out of 1000

Done!

Fig. 4. RTR Dashboard on AWS EC2 F1

On Fig. 4 execution of RTR on EC2 F1 instance is presented. We can see how much faster execution on F1 is compared to the execution on CPU. In case where we select portfolio with 10000 trades and we run it against 1000 scenarios the execution time on F1 is 2 minutes and 21 seconds while execution on CPU is 32 minutes and 53 seconds.

3.2 *Quantum Chromodynamics (QCD) Dashboard*

Quantum Chromodynamics is the theory for the Strong force that binds together the fundamental particles, called quarks, to form protons and neutrons, as well as other hadrons. The actual size of quarks is not known, but measurements indicate that they are more than 1,000 times smaller than the proton. One of the challenges in computational physics is calculating the binding of quarks by applying Monte Carlo methods to QCD.

On Fig. 5 execution of QCD on EC2 F1 instance is presented. We can see how much faster execution on F1 is compared to the execution on CPU. While execution on F1 is done the execution on CPU has finished about 40% of its job.

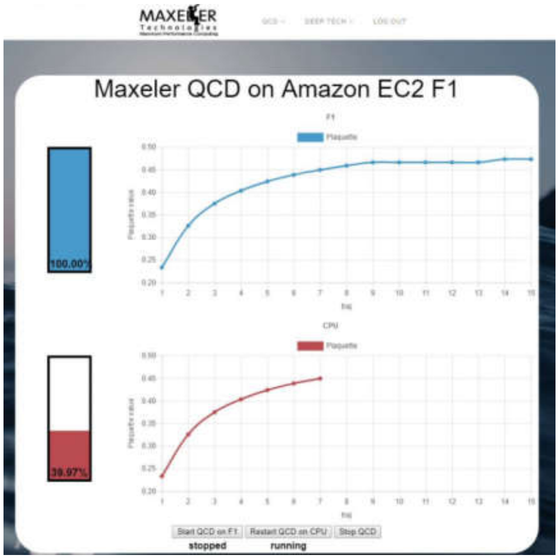


Fig. 5. QCD Dashboard on AWS EC2 F1

4. CONCLUSION

In this paper we have tried to resolve the problem of designing applications for EC2 F1 instance for user without significant expertise in hardware design. Currently, the only way to develop application for EC2 F1 instance is by using FPGA Developer AMI which requires good hardware background. We have proposed the usage of Maxeler's MaxWare which allows users without hardware background to develop applications for EC2 F1 instance. The full stack of application development for EC2 F1 instance using Maxeler's MaxWare is explained in details. Also, the description of two applications developed with Maxeler's MaxWare is presented. The results showed that RTR Dashboard executes 14 times faster on EC2 F1 instance than on CPU, while QCD Dashboard executes more than 2 times faster on EC2 F1 instance than on CPU. Those applications can be found on Amazon Marketplace as SaaS contracts and can be used by anyone. Further work on this research would be to put all Maxeler's AppGallery apps on AWS so that users could, with only few clicks, run them on EC2 F1 instance.

ACKNOWLEDGMENT

This paper was supported by the Maxeler Technologies.

REFERENCES

[1] Marco Rabozzi, Emanuele Del Sozzo, Lorenzo Di Tucci, Marco D. Santambrogio, "Five-point algorithm: An efficient cloud-based FPGA implementation," 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP), 2018

[2] Michaela Blott, Thomas B. Preußer, Nicholas Fraser, Giulio Gambardella, Kenneth O'Brien, Yaman Umuroglu, Miriam Leeser, "Scaling Neural Network Performance through Customized Hardware Architectures on

Reconfigurable Logic," 2017 IEEE International Conference on Computer Design (ICCD), 2017

[3] Ioannis Stamelos, Elias Koromilas, Christoforos Kachris, Dimitrios Soudris, "A Novel Framework for the Seamless Integration of FPGA Accelerators with Big Data Analytics Frameworks in Heterogeneous Data Centers," 2018 International Conference on High Performance Computing & Simulation (HPCS), 2018

[4] Voss N., Becker T., Mencer O., Gaydadjiev G. "Rapid Development of Gzip with MaxJ," Proceedings of the 13th International Symposium, on Applied Reconfigurable Computing (ARC '17); April 2017; Delft, The Netherlands. Springer International Publishing

[5] Maxeler's Real Time Risk (RTR) Dashboard AWS SaaS contract (accessed 30 November 2018), https://aws.amazon.com/marketplace/pp/B076JG3TYB?qid=1543580347698&sr=0-2&ref_=srh_res_product_title

[6] Maxeler's Quantum Chromodynamics (QCD) Dashboard AWS SaaS contract (accessed 30 November 2018), https://aws.amazon.com/marketplace/pp/B07FPVMR7R?qid=1543580347698&sr=0-1&ref_=srh_res_product_title

[7] Panfilov P., Salibekyan S. "Dataflow computing and its impact on automation applications," Proceedings of the 24th DAAAM International Symposium on Intelligent Manufacturing and Automation; October 2013; pp. 1286–1295.

[8] Bhattacharya B., Bhattacharyya S. S. "Parameterized dataflow modeling for DSP systems," IEEE Transactions on Signal Processing, vol. 49, issue 10, 2001, pp. 2408 – 2421

[9] Voigt S., Baesler M., Teufel T., "Dynamically reconfigurable dataflow architecture for high-performance digital signal processing,". Journal of Systems Architecture, vol. 56, issue 11, 2010, pp. 561–576.

[10] Li W. X. Y., Chaudhary S., Cheung R. C. C., Matsumoto T., Fujita M., "Fast simulation of Digital Spiking Silicon Neuron model employing reconfigurable dataflow computing," Proceedings of the 12th International Conference on Field-Programmable Technology, FPT 2013; December 2013; pp. 478–479

[11] Morris G. R., Abed K. H., "Mapping a jacobi iterative solver onto a high-performance heterogeneous computer," IEEE Transactions on Parallel and Distributed Systems, vol. 24, issue 1, 2013, pp. 85 – 91

[12] Jovanović Ž., Milutinović V. "FPGA accelerator for floating-point matrix multiplication," IET Computers and Digital Techniques, vol. 6, issue 4, 2012, pp. 249 – 256

[13] D. Veljovic, "Discrepancy Reduction between the Topology of Dataflow Graph and Topology of FPGA Structure," IPSI BgD Transactions on Advanced Research, July 2017, Volume 13, Number 1

[14] N. S. Voros, A. Rosti, M. Hübner, "Dynamic System Reconfiguration in Heterogeneous Platforms – The MORPHEUS Approach," Springer Verlag, 40, 2009.

[15] O. Pell, J. Bower, R. Dimond, O. Mencer, M.J.Flynn, "Finite-Difference Wave Propagation Modeling on Special-Purpose Dataflow Machines," IEEE Transactions on Parallel and Distributed Systems, vol. 24 , issue 5, 2013, pp. 906 – 915.

[16] D. Oriato, S. Tilbury, M. Marrocu, G. Pusceddu, "Acceleration of a Meteorological Limited Area Model with Dataflow Engines," Symposium on Application Accelerators in High Performance Computing (SAAHPC), Chicago IL, 2012, pp. 129 – 132

[17] Weston S., Spooner J., Racanière S., Mencer O. "Rapid computation of value and risk for derivatives portfolios," Concurrency Computation: Practice and Experience, vol. 24, issue 8, 2012, pp. 880 – 894

[18] Ivan L. Milankovic, Nikola V. Mijailovic, Nenad D. Filipovic, and Aleksandar S. Peulic, "Acceleration of Image Segmentation Algorithm for (Breast) Mammogram Images Using High-Performance Reconfigurable Dataflow Computers," Volume 2017 (2017), Article ID 7909282, 11 pages

[19] Milanković Ivan, Mijailović Nikola, Peulić Aleksandar, Filipović Nenad, "Application of Data Flow Engines in Biomedical Images Processing", IPSI BgD Transactions on Advanced Research, January 2018, Volume 14, Number 1

[20] Gan L., Fu H., Luk W., Yang, C., Xue, W., Huang, X., Zhang Y., Yang, G., "Solving the global atmospheric equations through heterogeneous reconfigurable platforms," ACM Transactions on Reconfigurable Technology and Systems, vol. 8, issue 2, 2015

[21] Kos Anton, Tomazic Saso, Salom Jakob, Trifunovic Nemanja, Valero Mateo, Milutinovic Veljko, "New Benchmarking Methodology and Programming Model for Big Data Processing," International Journal of Distributed Sensor Networks, 2015

[22] Maxeler AppGallery (accessed 30 November 2018), http://appgallery.maxeler.com

Ivan Milankovic was born in Gornji Milanovac, Serbia in 1988. He received his BSc and MSc degrees from the Technical Faculty Cacak (now Faculty of Technical Sciences) University of Kragujevac in 2011 and 2012 respectively. Currently he is PhD student at Faculty of Engineering University of Kragujevac. From 2012 to 2015 he has been engaged at the Bioengineering Research and Development Center (BioIRC). From 2015 he is engaged at Maxeler Technologies. His research interests include Computer Science.

Nemanja Trifunovic was born in Belgrade, Serbia in 1991. He received his BSc and MSc degrees from the School of Electrical Engineering University of Belgrade in 2013 and 2014 respectively. Currently he is PhD student at School of Electrical Engineering University of Belgrade. From 2014 he is engaged at Maxeler Technologies. His research interests include Computer Science.

A system for Crowdsensing Vibration Comfort in Smart Traffic

Baljak, Luka; Filipović, Filip; Jezdović, Ivan; and Labus, Aleksandra

Abstract: ***The purpose of this research is developing a system for crowdsensing vibration comfort in smart traffic. The aim of the research is to develop a system which enables monitoring of vibration in city traffic using mobile devices. The developed system for measuring vibration includes a mobile application for collecting data, a set of web services and a non relational database. The system was tested with an experiment carried out in a city traffic. The results show that the developed system can be used for measuring vibration in traffic using crowdsensing and it allows the community to contribute to comfort improvement in city traffic.***

Index Terms: ***smart cities*, *crowdsensing, smart traffic***

1. INTRODUCTION

THE main problem of vibration in traffic is their negative impact on human health, because the human body is not created to perceive vibration [1]. Due to the influx of the rural population into the urban environment, an increasing number of people spend a lot of time in public transport on a daily basis [2].

By using Internet of things it is possible to precisely collect vibration data. One way to collect data about vibration in traffic is through crowdsensing technique, which means that a large group of individuals who own mobile devices collectively share data and extract information from them in order to achieve a specific common goal, which in this case is to achieve a higher level comforts of public transport [3].

The aim of this paper is to develop a system for measuring vibration comfort of public transport using the crowdsensing technique. The system is based on the use of a mobile application, which can be used by a larger number of people, which enables fast and efficient collection of data that can be used for more efficient decision making.

2. RELATED WORK

2.1 Smart Cities

A smart city is an urban space that accelerates economic growth, offers high quality of life and facilitates citizen participation in health, education, utilities, business, transport and public security services [4]. In this work [5], a smart city is defined as a precisely defined geographical area in which ICT, logistics, and energy production are closely linked in order to create benefits for citizens in terms of well-being, improving the quality of the environment, and intelligent development. With the development of smartphones and advanced technologies (GPS, microphone, camera, etc.), citizens can collect data from urban areas. With these technologies, people can be part of smart cities and their services [6].

Areas of application of IoT solutions in smart cities can be categorized into several areas of application [7]: administration, participation program, and public security; buildings and houses; health protection; education, transport, and energy. Infrastructure of the Internet of intelligent devices and services in cities can contribute to the optimization of traffic, energy consumption, administrative and other processes [8]. From the aspect of communication, management and data processing, the multilayer architecture of the Internet of intelligent devices in smart cities consists of the following layers [9]:

- Measurement layer and sensors
- Network-Centric layer,
- Cloud-Centric layer,
- Data-Centric layer.

The lowest Internet infrastructure of intelligent devices is the sensor layer. It consists of intelligent (smart) devices that collect and process information from the environment. Such a device must have the following physical components [10]: power, memory, processor, and communication interface. The Network-Centric layer in the IoT infrastructure is responsible for providing a communication channel from sensors to the Internet, including the use of various technologies and network devices such as routers, base stations, and more. The Cloud-Centric layer is responsible for delivering available data and services to users. The role of cloud technology is to create an environment in which the management and use of sensors can be offered as a service to end users.

The application layer consists of applications that use data collected in the sensor layer to control various devices and smart buildings in the city.

Most of the existing solutions for smart cities are based on the integration of wireless communication technology, with the goal of creating a flexible and scalable infrastructure. A special segment in IoT infrastructure is a smart city solution that provides customers with the mobility and continuity of the network connection. The basic requirement is to enable use of different access technologies and to enable communication with smart devices and objects from different locations.

Technologies that enable development of a smart city include: internet intelligent devices, mobile technology, crowdsensing techniques, cloud computing, and big data.

2.2 Smart Traffic

The goal of smart traffic is to improve traffic infrastructure and traffic safety. It includes intelligent transport systems, automated traffic signaling and smart parking. Traffic control is of strategic importance in large cities [11]. Internet intelligent devices should provide interactive management of a central system for monitoring and regulating traffic. On the basis of the data obtained, traffic flows can be analyzed and improved in real time. The problem of traffic congestion is becoming serious due to increase in the number of inhabitants, the process of urbanization, and motorization. The use of ICT and intelligent transport systems (ITS) to monitor urban traffic can increase safety, make transport more efficient, reduce delays, and reduce environmental pollution.

2.3 Crowdsensing Techniques

Cities that have a predisposition to become intelligent can use IoT technologies and applications to collect and share real-time data [12]. Crowdsensing is a new paradigm that uses mobile devices to efficiently collect data, enabling the work of numerous large applications [13]. It is an ICT tool that focuses on different areas such as environment, citizen co-operation, urban traffic, health, and social networking [6]. The advantage of crowdsensing is use of sensory-based services [14], which is a cheaper way of applying smart cities in cities because they do not require expensive infrastructure [12].

Involvement of people is one of the most important features, and human mobility offers unprecedented opportunities for sensual coverage and data transmission [15]. There are two types of crowdsensing:

- Participatory sensing - requires participants to knowingly decide to meet application requirements by deciding when, where, how, and what observation to take.
- Opportunistic sensing - data from the environment is collected through applications without actively activating users (for example, continuous monitoring of Wi-Fi signals only requires that Wi-Fi be opened).

Some examples of using crowdsensing techniques include measuring pollution levels in the city, water levels, and monitoring of wildlife habitats. This way of collecting data enables mapping of various ecological phenomena by involving ordinary people. An example of a prototype for pollution monitoring is Common Sense [16]. Common Sense uses specialized handheld devices to measure air quality that communicate with mobile phones (using Bluetooth) to measure different air enthusiasts (e.g. CO2, NOx). These devices, when used on a large population, collectively measure air quality at a local or wider level. Similarly, microphones on mobile phones can be used to monitor the level of vibration in communities [17].

2.4 Theoretical Frame of Vibration Measurement

Vibration (vibratio - trembling, translated from Latin) represent periodic oscillations of the body around the equilibrium position. The equilibrium position is the position in which the body is not exposed to external forces [18].

Vibration is measured using an

accelerometer. It is a hardware component that is embedded in every smart phone. It reacts to the movement of the phone and updates each vibration for each axis. In case of hibernation of the phone, the X and Y axis vibration values will be approximately equal to zero, while the Z axis vibration will be equal to the gravity force on the phone. In order to avoid gravity from the observation itself, it is necessary to know in what position the phone is located during the measurement so that the value of the gravitational acceleration can be deducted from the values obtained from that axis. When the phone is on the move, the measured values represent a summarized motion of motion and gravity [19].

Accelerometer can be used to test soil vibration, to determine the vibration of workplaces or vehicles, and it finds great application in the mobile game industry. Manufacturers of same real games use movements that are controlled by the tilting of the phone, and the data on tilt and speed are precisely obtained from the accelerometer [19].

Determining the vibrations of the X, Y and Z axes is not enough to consider the effect on humans. Namely, it is necessary to perform a certain sizing on the basis of which it is possible to determine the influence. According to the ISO 2631-1 standard, it is first necessary to determine the mean square deviation [20]. After calculating the mean square deviation, it is necessary to determine the sum of the vectors of all measured vibrations at a specific location, which is designated as the total point vibration total value (PVTV) [20]. By summing up all total values of one-point vibration (PVTV), the total vibration total value (OVTV) is obtained [20]. Finally, in order to determine the comfort of the environment, the total vibration value (OVTV) obtained is compared to scale determining discomfort.

Discomfort is a subjective feeling that varies between individuals, but there is a certain consistency [21]. Human perception of vibration depends on its characteristics, such as strength, duration, orientation and characteristics of the people themselves, such as height, weight, gender, and age. A person in a sitting position (usually the target of a survey) senses vibration from 3 locations - seat, floor and backrest. Vibration at each location is viewed through three axes, X, Y and Z, as well as through rotation around them, as already explained. The rotational impact of the vibration in the backrest and the floor can be ignored. [20]

Human body has no single sensor or organ that senses vibration, but it has certain sensing systems that are sensitive to vibrations [22]:

1) Type - it is possible to detect the change in the position of the observed objects that occur as a result of shifting the head under the influence of vibration [22].
2) Vestibular apparatus - a static organ, which is placed in an internal ear which serves to maintain the balance and to provide information on the position of the body in the space [23]. If vibration cause head movements, this apparatus detects these changes and informs the brain about it [22].
3) Somatic nervous system - a part of the peripheral nervous system that is related to skeletal muscles under the voluntary control of body movements. It consists of afferent nerves that transmit impulses to the central nervous system [24],
4) Audit system - a sensory system that serves for the sense of hearing. It has the ability to detect the change in air pressure by more than 20Hz due to vibration [22].

When a man is exposed to vibration, his body negatively reacts to them [25]. Reactions can be controlled by muscles, but human body is not intended to deal with vibration, which can lead to health problems. Factors that can affect health deterioration due to exposure to vibration are cigarette smoking, obesity, duration of exposure, etc. [22].

One of the most common problems with vibration is lower back pain [26]. In the study [27], it was concluded, based on the large amount of observations, that vibration throughout the body has an impact on the occurrence of back pain. If, with the exposure to vibration, the lifting of the load is added, the effects are even greater. Although clinical back pain and lower back pain have never been correlated, there is statistically significant interdependence between these two phenomena.

In a study [28], conducted on a sample of 600 citizens exposed to body vibration, it was concluded that there is a significant correlation between vibration and lower back pain. The surveys used for data collection consisted of questions related to the private and business life of the respondents.

In addition to back pain, vibration has been studies as a cause of digestion, hearing, neck and shoulder pain, etc. [29]. Pain in the neck and shoulders was the subject of certain studies, but none succeed in finding the correct correlation. In the Ishitake study in 2002, vibration has been shown to affect the digestive tract, but not to the extent that it can endanger the health of a healthy person [20].

2.5 The Problem of Vibration in Traffic

In modern society, it is impossible to avoid encountering vibrations. Whether you are at work, or at home, or in the center of the city, you are exposed to the influence of vibrations. Whole body vibration by definition is transmitted through the accompanying surfaces and differs from the vibration transmitted from a certain part of the body, for example through hand. It occurs in a sitting, standing and lying position. In ISO 2631-1, most research is related to the seating position [20].

Passengers are irritated by vibration and consequently concentration may be reduced, depending on the psychological and physical condition of the person. For example, reading is difficult due to vibration [30]. A large number of people use public transport on a daily basis to carry out day-to-day activities. Although there is no official research that confirms that discomfort affects health, the assumption is that improving comfort has an impact on reducing health risks [20].

Traffic-induced vibration is one of the ways to cause damage to the environment. One of these ways is to create cracks in the facades of buildings, influence their depreciation, or have other impacts on households, etc. [31].

First of all, there are certain factors that influence the vibration itself [31]:

- quality and structure of sidewalk
- structure of vibration conductors to buildings
- structure and damage of roads
- speed and weight of the vehicle
- duration of exposure to vibration
- structure of the building (type, number of floors)
- distance from the road

It is very difficult to determine the effect of vibration on buildings experimentally, because there are many factors that affect the damage to the building. Precipitation, wind, man, aging, quality of the material, soil are just some of them. There are international and national standards that determine the allowed level of vibration in buildings, and one of the most famous is the ISO 4866 from the International Standardization Organization [31].

One way to collect data on vibration in traffic is through the crowdsensing technique, which implies that a large group of individuals owning mobile devices collectively share data and extract information from them in order to achieve a particular common goal, which in this case implies improving the vibration comfort of the environment [32]. According to classification of this technique referred to in [32], this form of crowdsensing is classified as an example of ecological crowdsensing in terms of the purpose of data collection.

3. Designing an IoT Based System for Measuring Vibration Comfort

3.1 Modeling the Architecture of the System for Measuring Vibration in Traffic

It is necessary to develop the infrastructure of the system for measuring vibrations in traffic using the Internet of intelligent devices. The proposed system includes the following components:

1. Mobile cellular architecture
    a. Power supply
    b. Communication modules
    c. Web services
    d. Accelerometer
    e. GPS
2. Mobile application architecture
    a. Local database
    b. Location services
    c. Accelerometer
3. Backend
    a. Database
    b. API for client applications
    c. Data processing and analysis services

The architecture of the system for measuring vibration in traffic using the Internet of intelligent devices is shown in Figure 2 and developed as a project of the Department of

Electronic Commerce at the Faculty of Organizational Sciences, University of Belgrade.

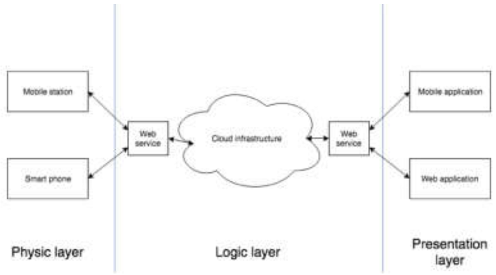


Figure 1 - Architecture of vibration measurement system in traffic in smart cities

In the physical layer, data from different sources is collected. The data is stored at the Cloud infrastructure through a web service. Communication is done through a web service. The presentation layer includes web and mobile applications. Data is displayed as tables, maps and charts.

Mobile stations with GPS and accelerometer are connected to the Raspberry Pi device. A web server written on the Raspberry Pi device allows wireless control of the system. The device can be installed in urban transport vehicles allowing data collection and transmission in real time.

Because of the large amount of data that needs to be stored, it is necessary to use an unrelated base. The mobile application of the mobile station communicates with the database through a web service. The web application uses data from the database for additional analysis and displays the results of vibration measurements to users.

3.2 Designing a Mobile Application

The mobile application enables manual and automatic vibration measurements that store time and location data for each measurement. Also, the application should provide the user with appropriate visualization of the measured results and, depending on the level of vibration, give the user an assessment of the level of vibration comfort. It is necessary to provide insight into the situation on certain lines of public transport, as well as on certain streets, residential and business buildings.

Android applications serve to communicate between systems and users. The software architecture of this part of the system is shown in Figure 2.

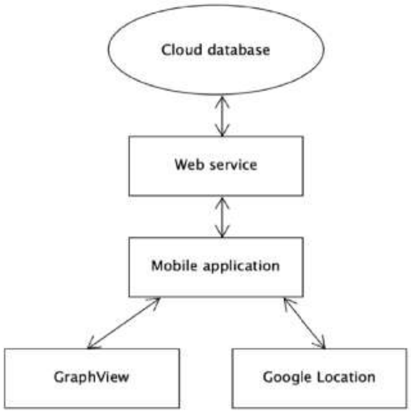


Figure 2 - Software architecture of the system

At the beginning of the process, recording of vibration data is performed. Upon completion of the recording, the data is stored in the local memory of the device and the process of analyzing the recorded data is automatically initiated. The analysis of the recorded data is done using the Fast Fourier Transformation used to record the recorded data in a separate representation of the separate frequencies. Upon completion of the analysis, the results, data, and metadata are sent via the API to the cloud.

From the user point of view the application has two ways to use it - manual and automatic, with more detailed specifications below.

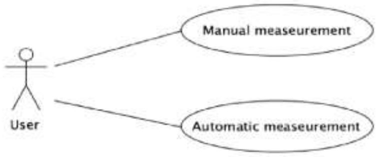


Figure 3 - Use Cases

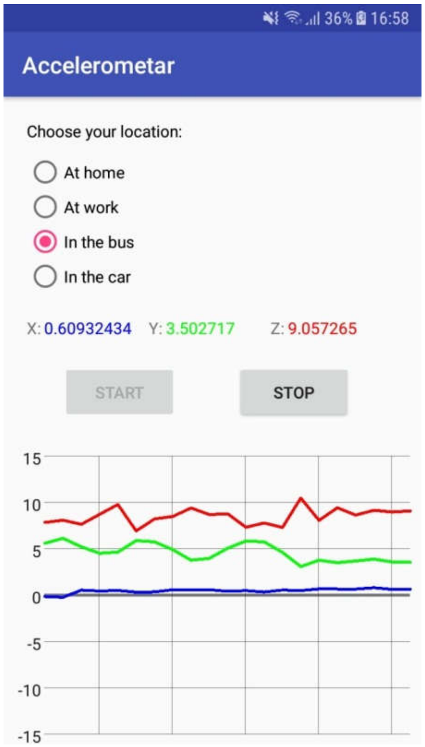
3.3 Implementation of the Mobile Application

The application is developed in Android Studio in the Java programming language and is intended for mobile devices that have an Android operating system. For simplicity, the application is in English.

Projected use cases are implemented as follows:

1) Use Case: Manual measurement. The user can choose between 4 locations - at home, at work, on the bus and in the

car. After selecting his location, the user starts recording by pressing the Start button. Storing the location of the mobile phone is implemented using Google Location API, provided by Google for free. The API works by returning the last known device location to each location request, which is not available only when the location is disabled on the phone. The graph displays real-time vibration changes and is realized with the GraphView plug-in. Since there are 3 axes, the vibration for each of the axes is presented with a unique color. By pressing the Stop button, the recording stops and all collected data is sent to the database via the web service.



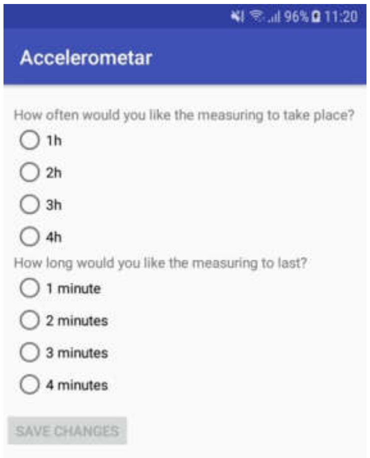
Choose your location:

○ At home
○ At work
● In the bus
○ In the car

X: 0.60932434 Y: 3.502717 Z: 9.057265

Figure 4 – Manual measurement

2) Use Case: Automatic measurement. The user initially selects the time interval between the two measurements and the duration of the measurement. Background measurements are made through the Background Service which is executed at a predetermined interval and lasts the selected time period. After each measurement, all data collected is sent to the database via a web service.



How often would you like the measuring to take place?

○ 1h
○ 2h
○ 3h
○ 4h

How long would you like the measuring to last?

○ 1 minute
○ 2 minutes
○ 3 minutes
○ 4 minutes

Figure 5 - Automatic measurement

Due to the incompatibility of the data for a relational database, a Mongo database was chosen, which was created on the site mlab, which provides service of storing Mongo databases on the cloud. The database is set up to have its own exposed public API (web service) through which it is possible to implement all CRUD operations from the application itself. As a rule, the documents in this database have their primary key - ID, which is used for accessing a document within the link to the API.

Each user in the database has own document in which all recorded vibration is stored. Due to the Mongo database structure, within the document of each user, there are five arrays, for the values measured in the bus, car, at home and at work, and one for the values measured automatically in the background, in which the measurement results are stored. Phone location is stored using two decimal values - latitude and longitude. Vibration is stored as oscillation values for X, Y and Z axes in a decimal format. The date and time are stored as a text value formatted as "dd.MM.yyyy. hh:mm:ss". Within the arrays, each element represents one measurement result, in which the recorded vibration, the date and time, location, and the width and length are stored.

In order to access data from an accelerometer embedded in a mobile device, it is necessary as the activity or service to implement the SensorEventListener interface and to select an accelerometer as a type of a sensor. After registering the sensors, activity/ service implements methods onAccuracy-Changed (Sensor sensor, int precision) and onSensorChanged (SensorEvent event). The second method is the most important one for this application because it contains implementation of the logic that is executed when a value of vibration on some axis changes, which happens very often. For this reason, storing the results of the measurement in the database is done every second, in order to avoid unnecessary overflow of data.

4. Evaluation

Testing of the system was performed from August 28th until September 3rd on the public transport line number 17 on the route between the stations of François DePere and the Marshal Tolbulhin Boulevard at the time interval from 12am to 22pm. The values obtained were measured in the seated position. Summarized results and derived parameters are given in the table 1.

Table 1: Summarized results

Type of bus	Value of vibration
Old	0.589
New	0.394
Total	0.417

Due to inability to find data on the age of the vehicle, or any categorization of vehicles in public transport, old vehicles are considered those which do not possess air conditioning, and new ones are the buses that have air conditioning. The total value of the vibration determines that the bus ride on the number 17 was overall a bit uncomfortable. From the obtained results, it can be concluded that the subjective perception of vehicle discomfort would be for new vehicles a little bit uncomfortable, while for old vehicles it could be said to be either a bit uncomfortable or completely uncomfortable, depending on the individual.

5. Conclusion

The use of an accelerometer, a hardware component embedded in many modern smartphones, for discovering impact of the environment on human health will surely experience a real bloom in the future.

There is no application that uses the measurements of vibration comfort of the environment mentioned in this paper. For example, after manually measuring vibration, it is possible to calculate the coefficient and to determine the level of comfort of the environment according to the scale. It is also possible to create reports based on automatic measurements and to welcome the user with them when they access the application the next time, or by notifying them that the report is ready. In addition, it is possible to allow user to supplement the list of locations to their will (adding cottages, trucks, sports centers, clubs or cafes in which a user spends his time) or in agreement with the public city transport the application can allow users to choose the exact line of public transport and garage number of a certain bus.

In order to increase the level of comfort of roads, buildings, and vehicles, it is possible to conclude an agreement with competent institutions that will present the measured vibration and will take appropriate measures of improvement in accordance with them. Based on the results of the measurements, it is possible to determine which streets are the biggest problems, which buildings need to be better isolated from the road, and which vehicles need to be repaired or removed from the traffic.

There are many opportunities for further exploitation of vibration for reducing health risks due to the increasing presence and ease of use of mobile devices. It is safe to expect that the mobile application industry will be one of the main exploiters of this functionality.

6. References

[1] Pope, M., Magnusson, A., Lundstrom, R., Hulshof, C., Verbeek, J. and Bovenzi, M. 2002, "Guidelines for whole-body vibration health surveillance", Journal of Sound and Vibration, vol. 253, no. 1, pp. 131-167.

[2] Kourtit, K., Nijkamp, P. and Arribas, D. (2012), Smart cities in perspective–a comparative European study by means of self-organizing maps. Innovation: The European journal of social science research, 25(2), pp.229-246.

[3] Ganti F., Ye F., Lei H.,2011, „Mobile crowdsensing: current state and future challenges“. IEEE Communications Magazine. 49(11), pp 32-39.

[4] L. Sanchez, I. Elicegui, J. Cuesta, and L. Munoz, “On the energy savings achieved through an internet of things enabled smart city trial,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 3836–3841.

[5] R. Dameri and R. P. Dameri, “Searching for Smart City definition: a comprehensive proposal Searching for Smart City definition: a comprehensive proposal TYPE (METHOD/APPROACH) Council for Innovative Research,” *Peer Rev. Res. Publ. Syst. J. Int. J. Comput. Technol.*, vol. 11, no. 5, 2013.

[6] K. Farkas and I. Lendak, “Simulation environment for investigating crowd-sensing based urban parking,” in *2015 International Conference on Models and*

Technologies for Intelligent Transportation Systems (MT-ITS), 2015, pp. 320–327.

[7] B. Radenkovic, M. Despotović-Zrakić, Z. Bogdanović, D. Barać, and A. Labus, *Internet inteligentnih uređaja*. Beograd: Fakultet organizacionih nauka, 2017.

[8] P. Wang, A. Ali, and W. Kelly, “Data security and threat modeling for smart city infrastructure,” in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, 2015, pp. 1–6.

[9] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, “An Information Framework for Creating a Smart City Through Internet of Things,” *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.

[10] L. Sanchez *et al.*, “SmartSantander: IoT experimentation over a smart city testbed,” *Comput. Networks*, vol. 61, pp. 217–238, Mar. 2014.

[11] B. Radenkovic, M. Despotović Zrakić, Z. Bogdanović, D. Barać, and A. Labus, *Elektronsko poslovanje*. Beograd: Fakultet organizacionih nauka, 2015.

[12] G. Cardone *et al.*, “Fostering participaction in smart cities: a geo-social crowdsensing platform,” *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 112–119, Jun. 2013.

[13] H. Ma, D. Zhao, and P. Yuan, “Opportunities in mobile crowd sensing,” *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 29–35, Aug. 2014.

[14] Á. Petkovics, V. Simon, I. Gódor, and B. Böröcz, “Crowdsensing Solutions in Smart Cities,” in *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia - MoMM 2015*, 2015, pp. 33–37.

[15] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, “Urban sensing systems,” in *Proceedings of the 9th workshop on Mobile computing systems and applications - HotMobile ’08*, 2008, p. 11.

[16] P. Dutta *et al.*, “Common Sense,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems - SenSys ’09*, 2009, p. 349.

[17] R. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[18] Jovančić P., „Skripta iz predmeta Tehnička dijagnostika, Univerzitet u Beogradu – Rudarsko-geološki faklutet“.

[19] Sharma S., „What is the accelerometer used for in mobile devices“, Source: www.credencys.com/blog/accelerometer/, [accessed 15.11.2018.]

[20] Marjanen Yka, „Validation and improvement of the ISO 2631-1 (1997) standard method for evaluating discomfort from whole-body vibration in a multi-axis environment“,Source:https://dspace.lboro.ac.uk/2134/6250, [accessed 15.11.2018.]

[21] Griffin, M.J., Parsons, K.C. Whitham, E.M. 1982, "Vibration and Comfort - 4. Application and Experimental Results", Ergonomics, vol. 25, no. 8, pp. 721-739.

[22] Mansfield, N.J. 2005, Human response to vibration, CRC Press, London.

[23] Nedeljko Bajić, Phisical Dimensions of Aging, Motor Control, Coordination and Skill, LECTURE 6, Faculty of Education in SOmbor.

[24] Berberović LJ., Hadžisemilović R., Dizdarević I., 2986, Medicinska antrolpologija, Svejtlost, Sarajevo.

[25] Pope, M., Magnusson, A., Lundstrom, R., Hulshof, C., Verbeek, J. and Bovenzi, M. 2002, "Guidelines for whole-body vibration health surveillance", Journal of Sound and Vibration, vol. 253, no. 1, pp. 131-167.

[26] Palmer, K.T., Harris, E.C., Griffin, M.J., Bennett, J., Reading, I., Sampson, M., Coggon, D. 2008, "Case-control study of low-back pain referred for magnetic resonance imaging, with special focus on whole-body vibration", Scandinavian Journal of Work, Environment and Health, vol. 34, no. 5, pp. 364-373.

[27] Seidel, H. Heide, R. 1986, "Long-term effects of whole-body vibration: a critical survey of the literature", International Archives of Occupational and Environmental Health (Historical Archive), vol. 58, no. 1, pp. 1-26.

[28] Bovenzi, M., Rui, F., Negro, C., D'Agostin, F., Angotzi, G., Bianchi, S., Bramanti, L., Festa, G., Gatti, S., Pinto, I., Rondina, L., Stacchini, N. 2006, "An epidemiological study of low back pain in professional drivers", Journal of Sound and Vibration, vol. 298, no. 3, pp. 514-539.

[29] Griffin, M.J. 1990, Handbook of human vibration, Academic Press, London.

[30] Sundström, J. 2006, Difficulties to read and write under lateral vibration exposure, PhD. thesis, KTH, Sweden.

[31] Jakubczyk-Galczynska A., Jankowski R., 2014. „Traffic-induced vibrations. The impact on buildings and people“, ISBN 978-609-457-640-9.

[32] Ganti F., Ye F., Lei H.,2011, „Mobile crowdsensing: current state and future challenges“. IEEE Communications Magazine. 49(11), pp 32-39.

Luka Baljak is a student of master studies at Faculty of organizational sciences, University of Belgrade. Currently, he is an associate at the Department of e-business. His research interests are: e-business, Internet technologies and m-business.

Filip Filipović is a student of master studies at Faculty of organizational sciences, University of Belgrade. Currently, he is an associate at the Department of e-business. His research interests are: e-business, Internet of things and Internet technologies.

Ivan Jezdović is a PhD student at Faculty of organizational sciences, University of Belgrade. Currently, he is an associate at the Department of e-business. His research interests include: e-business, Internet of things and web development.

Aleksandra Labus is an associate professor at Faculty of organizational sciences, University of Belgrade. She is a treasuress of IEEE Computer chapter C-16. Her research interests include: e-business, Internet of things and m-business.

An Application of Agent Based Simulation in E-education

Živojinović, Lazar; Naumović, Tamara; Barać, Dušan; and Despotović-Zrakić, Marijana

Abstract: ***The subject of this paper is agent-based simulation and how it is applied in e-education. The goal is to analyze the possibilities of using agent-based simulation models for solving*** **various** ***problems in the context of e-education. Based on the results of the analysis, two simulation models have been developed: one regarding grouping students into teams and the other one regarding the organization of lectures in e-learning. NetLogo software was used for implementation. The data was collected from the e-learning platform Moodle at the Department for e-Business, Faculty of Organizational Sciences. The results of the research show that agent-based simulation can contribute to the improvement of the teaching process.***

Index Terms: ***agent-based simulation models, e-education, NetLogo***

1. *Introduction*

THE subject of this paper is agent-based simulation and how it can be applied in e-education.

The continual development in the field of e-education necessitates using various tools for support in decision-making. An essential feature of such tools is their ability to process big amounts of data in real time, as well as the possibility that the results of their analyses will later become decisions that help improve the teaching process.

Agent-based simulation is used in many fields of science, such as Systems Theory, System Dynamics, Information Technologies, Management, Social Sciences, and Modeling and Simulation. However, there are not yet any satisfactory solutions in the field of e-education.

Agent-based simulation may greatly improve e-education. The purpose of the models developed through this research is to improve the tools which are used in e-education systems.

The goal of this paper is to overcome such a research gap by providing examples of how agent-based simulation is used for solving problems in e-education.

2. *Literature Review*

2.1 Simulation and modeling based on agents

One approach to computer simulation is based on agents (Agent-Based Modeling and Simulation, ABMS). Agents are characterized by their properties: the rules based on which they make decisions and their ability to interact with other agents. It is important to note that they can modify and adapt their behavior [1][2]. Simple behavior protocols and an agent's relations to other agents determine how it behaves [3].

Every simulation model has a big number of inputs and outputs, and often the systems they model are interdependent. The input variables of one system can be the output variables of another. Agent-based modeling and simulation provide more realistic models and new opportunities for simulation and modeling, and the output of such models can be better organized and later analyzed using specialized instruments.

The agents used in the simulation models presented in this paper come from the field of Robotics and Artificial Intelligence.. The main application of agents lies in modeling people's social behavior, social events, and individual decision-making [4].

Various scientific areas, such as Systems Theory, System Dynamics, Information Technologies, Management, Social Sciences, and Modeling and Simulation, all have connections to agent-based modeling and simulation.

In the first stages of their development, agent-based simulation models were used for modeling biological systems. Nowadays their usage is far

Manuscript received November 30, 2018. This work was supported in part by the Ministry of Education, Science and Technological development, grant no 174031.

Lazar Živojinović, Tamara Naumović, Dušan Barać and Marijana Despotović-Zrakić are with the Department of E-business, Faculty of organizational Sciences, University of Belgrade, Serbia (e-mail: {lazar, tamara, dusan, maja}@elab.rs).

wider, and they are mostly applied in modeling social processes.

The applications of agent-based models can be seen in stock market models, supply chains, epidemic models, and many other models. On one hand, agent-based models can be simple models used for academic purposes, and on the other hand, they can represent systems that support large-scale decision-making.

The most important characteristic of an agent is its ability to exist independently and provide an answer to any event that occurs. Agents have behaviors through which they independently make decisions, perform actions, and react to the inputs from other agents or from the environment, all with the aim of realizing required goals.

Researching students, their behaviors and habits constitutes the basis for analyzing an e-education system. This is significant both for the professor and the student. The analysis of students provides information about their needs.

The requirement is to build a model of a part of an e-education system. The questions that need to be answered are the following ones:

- How do the students behave?
- What are the results of applying certain education strategies?
- What are the results of certain pedagogical approaches?

2.2 Rules for ABMS Models

Constructing models based on agents is similar to constructing any simulation model. At the beginning, it is necessary to determine the purpose of the model, then the questions for which the model should provide the answers, and finally who the users of the model are.

The process of modeling has multiple iterations. The following figure demonstrates the cycle of modeling [5].

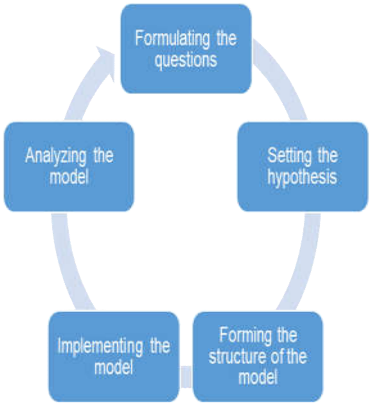


Figure 1: *T*he cycle of modeling agent-based models

2.3 Environments for Developing ABMS Models

ABMS models may be created using the standard programming languages and tools, or in environments that are suited specifically for such kind of modeling. All these software environments have multitude of functionalities and there is a plethora of scientific literature on the subject.

There are multiple approaches and tools when it comes to developing ABMS models [6]:

- modeling using spreadsheet software – mostly MS Excel, while also using VBA (*Visual Basic for Applications*) macros,
- usage-specific programming languages and environments, such as NetLogo, Repast, Repast Symphony, StarLogo etc.,
- systems for mathematical modeling, such as Mathematica, MATLAB and others and
- standard software languages and tools, such as .NET, Python, Java and others.

3. *An Application of Agent Based Simulation For Grouping Students into Teams*

Within the Department for Electronic Business, there is a course named Electronic Business into which six hundred students enroll each year. Within the course there are assignments and a project which constitutes the final part of the exam. Students can work on everything individually or in teams. However, the course's focus is on the implementation of the project by a group of at least two students and at most three students. It is necessary to determine a way for these teams to form.

For solving this problem, a simulation model made in NetLogo (version 6.0.3), is used [7][8]. The following figures show the user interfaces for situations when a team of two students is formed, as well as when a team of three students is formed. The figures consist of the model's inputs, its and the simulator's parameters, the agents' interactions with one another, and finally the outputs. The outputs can be seen on the graph on the right.

This model is made up of inputs, which are represented using the following variables:

- the size of the team (*team-size*),
- the probability that a new addition to the team will be a student experienced in teamwork (*p slider*),
- the probability that a former partner will be chosen (*q slider*) and
- the number of steps (*max-downtime*).

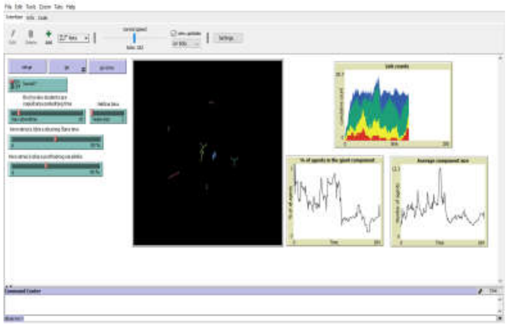


Figure 2: ***The results of the simulation for a two-member***

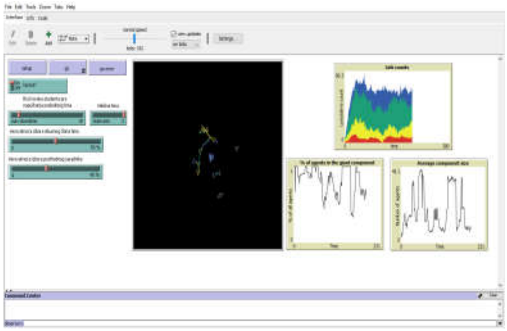


Figure 3: ***The results of the simulation for a three-member team***

This model contains two groups of students, or more precisely team members: newcomers and incumbents. The first type represents students who had not previously collaborated with anyone, whereas the second type represents the students who already have experience in teamwork (this is not the first time they are collaborating with someone on a project), and who are more knowledgeable (or more educated).

In the left part of the user interface there are controls (sliders) that reference the input variables, and their values are determined at the beginning of the simulation. The control for team-size represents the number of team members and possible values are 2 and 3. The next control is the p slider, which represents the probability that a new addition to the team will be an experienced student (incumbent), whereas q slider represents the probability that a former partner will be chosen. Once the formation of the teams is finished, all the team's members become connected. If students do not participate in their team's activities for a certain period of time, they are removed from the model, as are all of their connections. That period is defined though a number of steps, which is determined by the control max-downtime.

The students who are newcomers are in green and the ones who are incumbents are in yellow. The color of the relationship that belongs to a team member signifies which of these two types the member is. If the relationship is in blue, the student is of type newcomer. The colors yellow and green represent incumbent-incumbent and newcomer-incumbent relationships, respectively. The color red represents the relationship between students who have partnered on a project before.

On the right part of the screen there are graphs that contain the outputs of the model, and on it there are the results of the simulation.

The figure below shows the graphical output of a model for a two-member team. Here the model is centered on the type of the relationship between the team members.

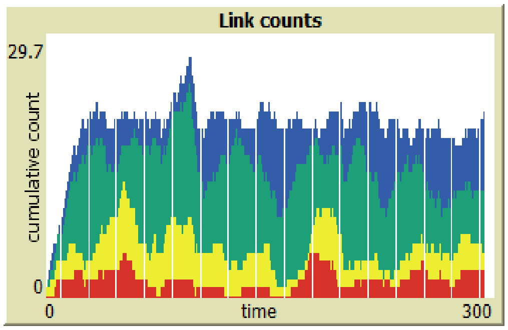


Figure 4: ***Graphical representation of the relationships in a two-member team***

The following graph shows that most relationships are of type newcomers (in blue), where the team is made up of two inexperienced students; there are fewer relationships of type newcomer-incumbent (in green), where the team is made up of one inexperienced and one experienced student. Finally, there are the fewest relationships of type incumbent-incumbent.

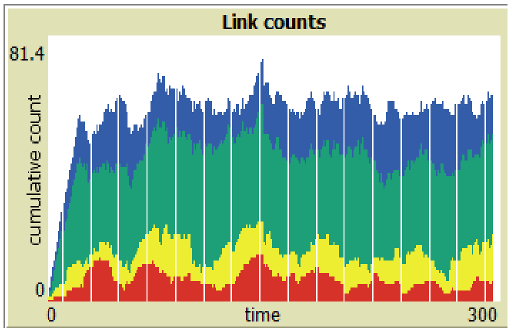


Figure 5: ***Graphical representation of the relationships in a three-member team***

The previous graph shows that most relationships are of type newcomers (in blue), where the team is made up of three inexperienced students; there are fewer relationships of type newcomer-incumbent (in green), where the team is made up of one inexperienced and two experienced students (or two inexperienced and one experienced student). Finally, there are the fewest relationships of type incumbent-incumbent.

If the relationships of type newcomers (in blue) were dominant in number over the other two types, we could conclude that the experienced students - the ones who have experience in teamwork and are more educated - are underperforming. If there were more relationships in red and the relationships of type incumbent-incumbent (in yellow), then we could conclude that there is a lack of creativity and experience among the students.

Professors need to have an insight into how the teams are formed, or more precisely what the team members are like, and based on that, they can distribute the assignments.

4. An Application of Agent Based Simulation for Organizing Lectures in E-education

Within the aforementioned course on Electronic Business, practical lectures are delivered in classrooms that are equipped with computers. There are classrooms with 20, 30 or 60 seats. The course's practical lectures are delivered by a professor or a teaching assistant, with the help of a few learning assistants, whose number depends on the capacity of the classroom. There can be one, two, three or four assistants.

The following table contains the data on how the practical lectures in computer-equipped classrooms may be organized.

Capacity (number of seats)	Number of Learning Assistants
20	1, 2
30	1 - 4
60	1 - 4

Table 1: ***The possibilities for organizing practical lectures***

For solving this problem, a simulation model made in NetLogo (version 6.0.3) is used [8][9][10]. The following figure contains the results of the simulation for a 30-seat classroom. The results consist of the model's inputs, its and the simulator's parameters, the agents' interactions, and finally the outputs, which are shown on the graph.

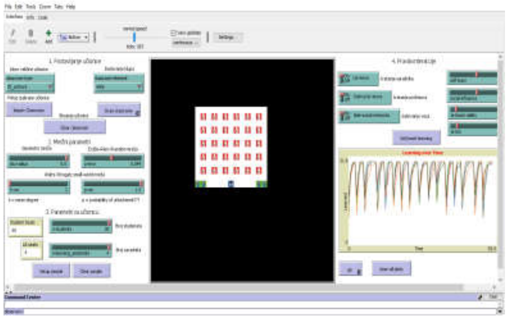


Figure 6: ***The results of the simulation for 30-seat classroom***

Within the model there are three types of agents: professors (Instructor), students (Students), and assistant (Learning Assistant, LA). Red squares represent the students' seats. The blue square represents the place where the professor or the assistant delivers the lecture, and the green squares represent the learning assistants. The white space represents the area in which the learning assistants may move.

This model contains three networks which connect the students: Geometric, Erdős-Rényi Random and Watts-Strogatz small-world networks. Based on these networks, which demonstrate the number of one's neighbors, we may propose the following scenarios of how the students study:

1) lecture only: individual studying, with no interactions with others;
2) ER-RN: individual studying, with help from the neighbors (students) in the random network;
3) WS: individual studying, with help from the neighbors (students) in the small-world network;
4) Geom: individual studying, with help from the neighbors (students) in the geometric network, and
5) Geom-LA: individual studying, with help from the neighbors (students) in the geometric network, and the learning assistants who answer questions.

Students who don't follow any of these scenarios when it comes to studying might still raise their hand in class. The learning assistants then approach them. Once they have answered all of the students' questions, they go back to their initial positions.

The right part of the user interface contains the controls (sliders) for the rules of interaction. The control self-learn represents the probability that a student will study passively. Social-influence represents the probability that a student will study with others, and the control la-teach-ability represents the probability that a student's knowledge is the result of their interactions with learning assistants. La-tick represents the time a learning assistant spends with a student.

The following figures show the graphical output of a model for a 30-seat classroom, which may have one, two, three or four learning assistants.

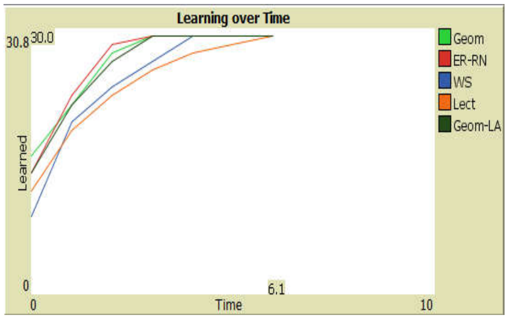


Figure 7: ***Graphical representation of the outputs of a model for a 30-seat classroom with one learning*** **assistant**

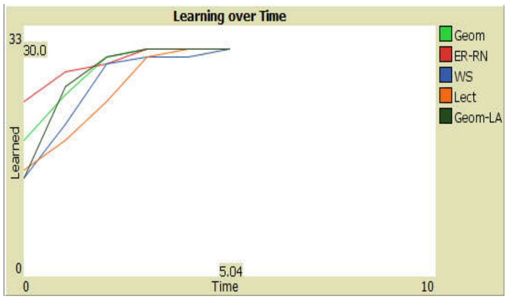


Figure 8: ***Graphical representation of the outputs of a model for a 30-seat classroom with two learning*** **assistants**

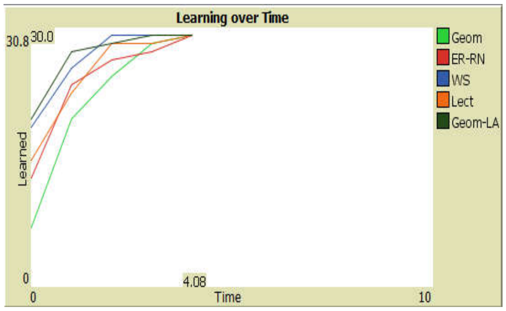


Figure 9: ***Graphical representation of the outputs of a model for a 30-seat classroom with three learning*** **assistants**

These figures show graphs with the number of students who have successfully followed through the lectures (that is, completed their assignments); the scenario they used; and finally the time they took to accomplish all this. We may conclude that in all of these four situations all 30 students successfully followed through the lectures, in a certain period of time, using different scenarios.

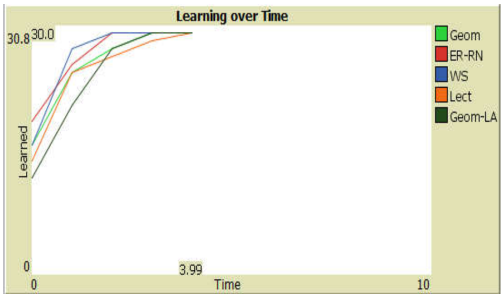


Figure 10: ***Graphical representation of the outputs of a model for a 30-seat classroom with four learning*** **assistants**

The difference between the students is in time they needed to complete the assignments. In the first case – in the case of a classroom with a single learning assistant – they needed 6.1 units of time. In the case of a classroom with two learning assistants, they needed 5.04 units of time. In the case of a classroom with three learning assistants, they needed 4.08 units of time, whereas those in a classroom with four learning assistants needed the least amount of time: 3.99 units. Based on these results, we may conclude that the difference between the last two cases (three and four learning assistants) is not significant, so the decision regarding the optimal number of learning assistants for a 30-seat classroom needs to be made using other parameters.

5. Conclusion

Research into students, their behavior and their habits, constitutes the basis for analyzing an e-education system. This is significant both for the professor and for the student.

More and more educational facilities are using the tools of electronic education, which is a more effiecient way of teaching. It provides better communication between the participants in the process and motivates students to perform better.

Agent-based modeling and simulation provide more realistic model and new possibilities in the field of simulation and modeling.

This paper may help solve multiple problems in the context of e-education, such as those related to grouping students into teams and organizing lectures.

The results of the research show that agent-based simulation can contribute to the improvement of teaching process.

REFERENCES

[1] Prokopenko, M., Boschetti, F., Ryan, A., “An information-theoretic primer on complexity, selforganisation and emergence,” Complexity, Volume 15, 2009, pp. 11–28.

[2] Abbott, R., “Emergence explained: Abstractions: Getting epiphenomena to do real work,” Complexity, Volume 12, Issue 1, 2006, pp. 13–26.

[3] Miller, J.H., Page, S.E., “Complex Adaptive Systems: An Introduction to Computational Models of Social Life,” Princeton University Press, 2007, pp. 114-118.

[4] Bonabeau, E., Dorigo, M., Theraulaz, G., “Swarm Intelligence: From Natural to Artificial System,” Oxford University Press, 1999, pp. 19–27.

[5] Grimm, V., Railsback, S., “Individual-based Modeling and Ecology,” Princeton University Press, 2004, pp. 25–34.

[6] Čavoški, S., “Simulacioni modeli zasnovani na agentima kao podrška odlučivanju u elektronskom poslovanju,” FON, 2016, pp. 47-55.

[7] Bakshy, E., Wilensky, U., NetLogo Team Assembly model, http://ccl.northwestern.edu/netlogo/models/TeamAssembly. Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL, 2007.

[8] Bakshy, E., Wilensky, U., NetLogo Team Assembly model, http://ccl.northwestern.edu/netlogo/models/TeamAssembly. Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL, 2007.

[9] Wilensky, U., NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.

[10] McDevitt, A.L., NetLogo Classroom Model with Learning Assistants. Department of Integrative and Systems Biology, University of Colorado Denver, Denver, CO, 2017.

[11] Bonwell, C.C., Eison, J.A., Active learning: Creating excitement in the classroom (ASHE–ERIC Higher Education Rep. No. 1), Washington, DC: The George Washington University, School of Education and Human Development, 1991.

Lazar Živojinović completed his bachelor and master studies at Faculty of Organizational Sciences, University of Belgrade. Currently, he is a PhD student at the Department of E-business. His research interests are computer simulation, e-education, and e-business.

Tamara Naumović is a teaching associate at Faculty of Organizational Sciences, University of Belgrade. She has a PhD student at the Department of e-business. Her research interests include computer simulation, internet of things, and software development.

Dušan Barać is an associate professor at Faculty of Organizational Sciences, University of Belgrade. His research interests include computer simulation, e-business, internet technologies, and web development.

Marijana Despotović-Zrakić is a professor at Faculty of Organizational Sciences, University of Belgrade. She is a Chair of the Laboratory for simulation, and a Vice-chair of IEEE Computer chapter C-16. Her research interests include computer simulation, internet technologies, and e-education.

Innovative IoT-based Business Models in Telecommunications

Kokolj, Srđan; Stojanović, Mirjana; Bogdanović, Zorica; and Radenković, Božidar

Abstract: ***The topic of this article is innovative IoT-based business models in telecommunications. The goal of this article is to analyze and explore the possibilities and opportunities for telecommunication operators to enter IoT market through the selection of a business model that provides support for the development of innovative IoT services. As a practical example, an IoT service for smart objects monitoring using telecom IoT platform as well as the web application for displaying and analysis of sensor data in real time is presented. The results are showing that telecom operators, through the process of digital transformation in the telecommunication industry, can select an appropriate IoT business model that can enable support for innovative IoT-based services that provides additional revenue streams.***

Index Terms: ***business model, digital ecosystem, Internet of Things, IoT, monetization, operators, telecommunications***

1. INTRODUCTION

All types of Communication Service Providers (CSPs), but particularly mobile operators, are facing serious challenges. To follow rapid development of the technology, and to fulfill constantly evolving regulation and increased expectations of the end-users, require significant investments. On the other side, traditional telco services are perceived as the commodity and the revenue from them is declining, as the new entrants, like OTT players, provide more convenient alternatives (example: messaging).

Nowadays, two key trends are present in mobile operators' businesses:

- Focus on reducing cost and increasing automation, productivity, and efficiency
- Search for the new services/ new revenue streams

At the same time, digitalization in other industries means that networks and communications functions are becoming crucial for their business and integrated into their core structures. While in some cases generic telco connectivity offers will be suitable, others will require some more customized variants. The strategic question for the telco operators is where to position themselves in the new value chains (or value networks) and how to redefine their role in the market [1]. One extreme strategy could be to remain pure Infrastructure or Connectivity provider; the other one might be to become Innovative Services Provider, with different combinations in between.

In the changing business landscape, telecom service providers have particularly recognized the huge potential of the IoT market. Internet of Things is forecast to reach over 25 billion connections and the global IoT market is estimated to be worth $1.1 trillion in revenue by 2025 [2],[3]. Telecom operators are among the players aiming to capture part of this revenue, however, the way on how to achieve this and the relevant business models are not clearly set in all the cases [4].

The problem addressed in this research is the systematic approach, from the strategy definition to the practical implementation that operators should take to fully realize the IoT potential. Thus, in the first part, this paper analyzes different possibilities on how the telecommunications operators can become a part of the new IoT ecosystems and how they can monetize it through the different business models. In the second part, it shows a practical example of the IoT-based service that telecom operators could offer to both the enterprise and the residential segments of its subscribers' base and the business model that could be applicable for such a case. While most of the existing works in this

Manuscript received November 30th. This work was supported in part by the Ministry of Education, Science and Technological Development, Republic of Serbia, under Grant 174031.

Srđan Kokolj is with the Faculty of Organizational Sciences, University of Belgrade, Serbia (e-mail: srdjan.kokolj@gmail.com).

Mirjana Stojanović is with the Faculty of Organizational Sciences, University of Belgrade, Serbia (e-mail: stojanovic.p.mirjana@gmail.com).

Zorica Bogdanović is with the Faculty of Organizational Sciences, University of Belgrade, Serbia (e-mail: zorica@elab.rs)

Božidar Radenković is with the Faculty of Organizational Sciences, University of Belgrade, Serbia (e-mail: boza@elab.rs)

area remains in the theoretical domain, this paper illustrates the concrete deployment of an IoT service and how operators can benefit on it.

2. The Role of Telco in the IoT

2.1 IoT Business Models

The development of the Internet of Things, or IoT, enabled by the technology evolution, creates new, significant, business opportunities in different areas and impacts different industries. World Economic Forum, for instance, forecasts only Industrial IoT to add $14 trillion of economic value to the global economy by 2030 [5]. An approach in using IoT could be sustained, which improves existing products and services, or disruptive, which creates completely new ones [6]. In both cases, the way of interactions among the actors relevant for the business is changed and to capture part of IoT commercial potential, new business models are needed.

Researches indicate the value proposition, followed by customer relationships and key partnerships, respectively, as the three most important building blocks of the IoT related business models [7].

2.2 IoT Value Propositions Offered by Telco

Telecom operators have high expectations from the IoT business, despite still open questions related to the technology and the standardization. In already saturated market, IoT is seen by the operators as the main opportunity to increase the number of connections and as the potential for the new revenue streams [2],[8]. The already happening or anticipated IoT scenarios that telecommunications are interested in are often classified as Massive IoT and Critical IoT. Massive IoT use cases are those including a massive number of low-cost devices, typically transmitting a relatively low volume of non-delay-sensitive data (smart metering, smart agriculture, fleet management, tracking, etc.). On the other side, Critical IoT use cases have very strict requirements related to throughput, latency, reliability, and availability (industrial applications, remote manufacturing, traffic safety and control, etc.) [9].

Deployment of those use cases requires collaborations with different partners or even industry verticals. In the new IoT business ecosystem, telecom operators can take different roles and offer different value propositions. Typical value propositions are listed below [10] ,[4], [8]:

(a) Reliable and scalable network to connect IoT devices

This is the basic offering of the operator in the role of the network provider. It can be further enhanced by the services of integrating customer devices, providing the relevant network data or providing managed services for enterprises [10],[4],[8].

(b) Network capabilities together with IoT devices and management of their connections

This value proposition of the operator in the role of connectivity provider can evolve by including device lifecycle management, security updates, analytics capabilities, validation and the storage of the collected data, etc. [10],[4],[8]

(c) Horizontal platform to facilitate onboarding of new ecosystem partners and the introduction of the new IoT services to the market

Key capabilities of such a platform are orchestration, analytics, security, and policy management to enable integration of different ecosystem participants and open APIs to involve application developers. The modularized platform building blocks (e.g. billing, security, analytics), are usually offered "as-a-service". Besides operation of the platform and the IoT marketplace, the operator could offer the applications (either developed internally or in the partnership or bought from the third party) and systems integration and consulting services [10],[4],[8].

(d) End-to-end services directly to end users

Value proposition that gives to the operator the highest role in the IoT value chain is the most complex for implementation. It might require transformation of the operator's own sales, distribution and delivery channels and acquisition of the specific vertical industry's expertise [10],[4],[8].

Which role will be taken is determined by operators' overall strategy and its resources/capabilities, but also by the targeted customer segment. The same operator can take different roles depending on the specific IoT use case and the different value propositions in the B2C, B2B or B2B2X scenarios.

2.3 Implementation Considerations

Most of the telco operators in the world are still exploring how to maximize IoT opportunities. In each of the four basic scenarios outlined above, in addition to the already mentioned customer relationships and the key partnerships, privacy and security policies and monetization strategy could be crucial success factors [3], [11], [12].

Partnering versus acquisition is the decision that should be carefully evaluated, considering the internal resources, capabilities, and strategy but also relevant cost-benefit analysis and time-to-market in both cases. If the partnership is the selected way forward, early involvement of the selected partners is necessary.

Different analysis indicated that security issues are a significant inhibitor to the deployment and adoption of many new IoT services. Additionally, in the case of IoT services for the consumer market, the privacy of personal data is becoming more and more important [11],[13]. Telecom operators aiming to offer new and innovative IoT services, have to take into consideration all the threats their service may face and to address security and privacy challenges in a serious way. The telecommunications industry already has a long history of providing secure products and services to their customers, which is a good starting point for them to become a trusted IoT service provider [11]

The aspect that is becoming particularly important for the innovative IoT services is the monetization. Most service providers are currently adopting one of the three classic pricing models (fixed fees, transaction fees, and revenue sharing). However, pricing policies from traditional telco offering might not be suitable for future IoT use cases, especially with the complex relationships between different ecosystem participants [12].

Examples of innovative models that are currently evaluated by the telco industry are the 'Quality of Experience'-based pricing, which consider several objectives and subjective components of the quality of experience [12] or the Outcome-based pricing [14]. Instead of charging by traffic/volume or number of devices, this pricing model sets the prices for enterprise clients based on achieving jointly determined outcomes.

3. Example of an IoT Service and the Relevant Business Model

This chapter describes in detail an example of IoT based monitoring service and the relevant business model. In this case, a telco operator is in the role of the platform provider. The value proposition is related to the security of different residential and business objects. A relevant application can be developed and offered to the end-users by external partners or by the operator itself. If the operator's role is extended to the end-to-end service provider, it can leverage on the existing customer relationship with both residential and business segment. Additionally, new IoT based service can be bundled with other services from the operator's portfolio.

3.1 Innovative Service for Objects Monitoring

The possibility of development of value-added services with inexpensive IoT devices is very interesting in many ways. It provides location capabilities for IoT services through continuous tracking [15]. The project task is to create web application as a service for objects monitoring with the entities that are IoT devices with sensors, network infrastructure, and IoT platform (Figure 1). IoT platform stores data generated by sensors on IoT devices and exposes it through external interfaces. The web application uses these interfaces to display collected data to the user.

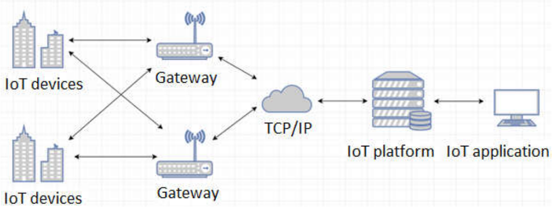


Figure 1. System architecture

Security of residential and business objects is very important. IoT based security system as an innovative service is reducing the cost of the protection. IoT devices in the objects can be easily connected and added to the system. IoT infrastructure is providing support for smart objects service creation through the enablement of data protection, access authorization, and data integrity. Innovative IoT service for the protection of objects is enabling access to the information related to monitored objects in real time and registration of alarms.

The main goal of this project is the improvement of security systems in terms of increased efficiency, lower costs, and improvement of scalability. Devices are sending defined parameters that include battery level, temperature, humidity, as well as the alarms. (Figure 2).

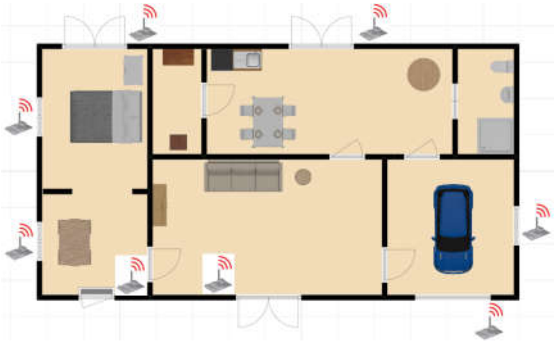


Figure 2. Locations of the sensors

IoT web application provides the functionality of the administration of users as well as the presentation of the data from the sensors (Table 1). In case of unauthorized access, users are informed through generated alarms.

Functionalities of the IoT web application	
Administrator	User
List of all registered sensors Data collected from all sensors Date and time of last sensor activity List of alarms (motion detection) Users administration Access rights	List of registered user sensors Data collected from registered sensors List of defined alarms (motion detection) User details

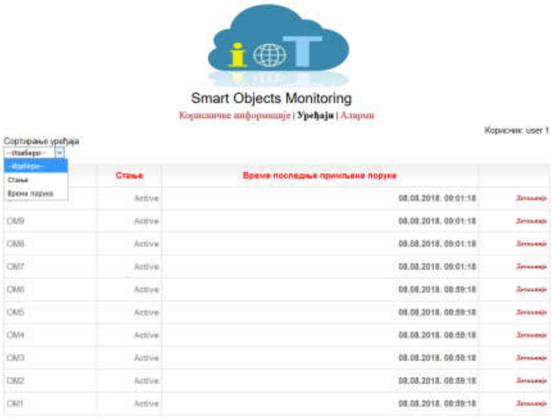
Table 1. Functionalities of the application

Communication of web application with IoT platform is through open API interfaces that are enabling presentation of collected data to the users of the application. There is a set of interfaces that are providing different functionalities based on the user type.

3.2 Implementation of the Application for Buildings Monitoring

List of devices - user

Figures 3 and 4 show a list of generated alarms for registered user service. Users can also see the basic information about each IoT device that includes humidity, temperature, and battery level.



Smart Objects Monitoring

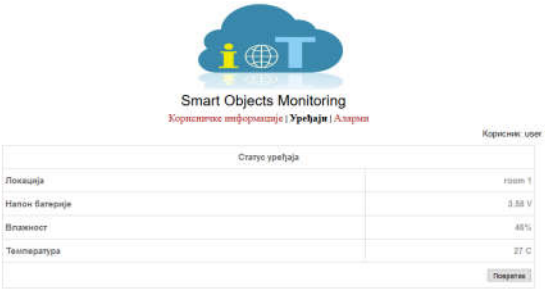
Корисничке информације | Уређаји | Аларми

Корисник: user 1

Сортирање уређаја

	Стање	Време последње примљене поруке	
	Active	08.08.2018. 09:01:18	Детаљније
OM9	Active	08.08.2018. 09:01:18	Детаљније
OM8	Active	08.08.2018. 09:01:18	Детаљније
OM7	Active	08.08.2018. 09:01:18	Детаљније
OM6	Active	08.08.2018. 08:59:18	Детаљније
OM5	Active	08.08.2018. 08:59:18	Детаљније
OM4	Active	08.08.2018. 08:59:18	Детаљније
OM3	Active	08.08.2018. 08:59:18	Детаљније
OM2	Active	08.08.2018. 08:59:18	Детаљније
OM1	Active	08.08.2018. 08:59:18	Детаљније

Figure 3. List of IoT devices



Smart Objects Monitoring

Корисничке информације | Уређаји | Аларми

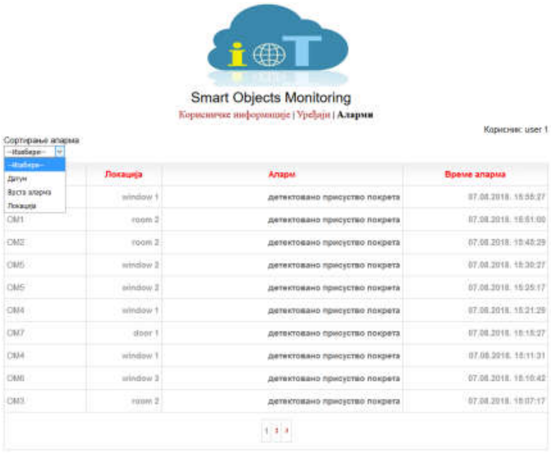
Корисник: user 1

Статус уређаја	
Локација	room 1
Напон батерије	3.58 V
Влажност	45%
Температура	27 C

Figure 4. IoT device details

List of alarms – user

Figure 5 shows a list of alarms generated from the user registered IoT devices. All the alarms can be sorted based on date, type, and location of the device.



Smart Objects Monitoring

Корисничке информације | Уређаји | Аларми

Корисник: user 1

Сортирање аларма

	Локација	Аларм	Време аларма
	window 1	детектовано присуство покрета	07.08.2018. 15:55:27
OM1	room 2	детектовано присуство покрета	07.08.2018. 15:51:00
OM2	room 2	детектовано присуство покрета	07.08.2018. 15:45:29
OM5	window 2	детектовано присуство покрета	07.08.2018. 15:30:27
OM5	window 2	детектовано присуство покрета	07.08.2018. 15:25:17
OM4	window 1	детектовано присуство покрета	07.08.2018. 15:21:29
OM7	door 1	детектовано присуство покрета	07.08.2018. 15:15:27
OM4	window 1	детектовано присуство покрета	07.08.2018. 15:11:31
OM6	window 3	детектовано присуство покрета	07.08.2018. 15:10:42
OM3	room 2	детектовано присуство покрета	07.08.2018. 15:07:17

Figure 5. List of alarms

Analysis of the results

IoT platform enables presentation and analysis of its performance through the number of API calls, the number of sent messages, execution time, and the number of services/applications. Based on the list of all registered IoT devices for object monitoring, all devices are sending information from their sensors that enable analysis. In real time, the IoT platform shows the information that comes from the sensors and enables presentation of all data that has been received in the defined period.

Through the detailed presentation of the parameters that are registered by the IoT device, depending on the user type, web application shows the data. Additionally, the application administrator can see the status of all sensors on the device. The list of alarms enables users to follow the changes in the proximity of the sensors, in order to register intrusion detection with the precise location and time. As it can be seen based on the results, all detected changes in humidity and temperature are shown in the generated graphs (Figures 6 and 7).

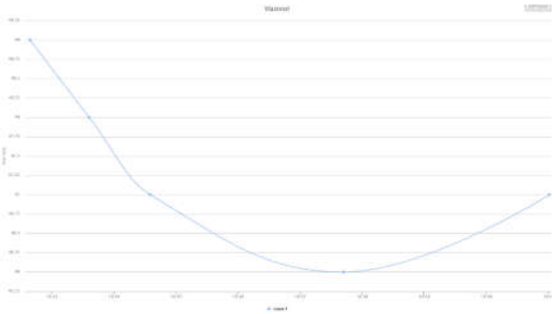


Figure 6. Humidity graph

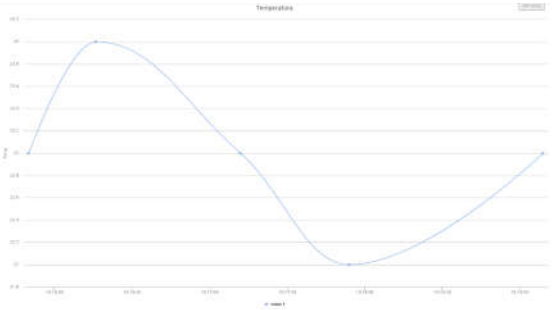


Figure 7. Temperature graph

2.4 3.3 Possibilities for Monetization

Service described above allows the operator different alternatives for the monetization. It can be a "pay-per-use" model based on the number of connected devices or the number of transactions made by them. The other alternative could be the fixed fee which covers a certain number of devices and/or transactions for a certain period. Additionally, the operator can charge for the data storage, geolocation information, and the analytical capabilities. There is a possibility for revenue share with the partner, e.g. application provider.

4. Conclusion

Since telecom operators are experiencing reduced revenues from the standard services, they need to follow new trends to stay competitive in the market [4]. The strategy of the operators is to keep the dominant position in the industry and to have a larger share of the additional services market through the support for innovative IoT services.

IoT architecture in the process of fast evolution is not standardized and is heterogeneous. There are different hardware components, networks, and IoT platforms [16]. Operators are using different approaches from the creation of private API-s to the management of access through transparency and collaboration via the open platform. Most of the operators are primarily focusing on the IoT connectivity enablement. The advantages of operators as providers of open IoT platform include possibilities of integration with different hardware and software standards and enablement of interoperability and scalability.

The main contribution of this paper is that it gives a practical example of an innovative service and the relevant business model in case where the telecom operator is providing the IoT platform. The innovative service for monitoring of objects, which is described in this example, uses the advantages of fast development of the applications via simple connectivity with the platform and usage of open API-s. Based on the data from the platform, the application is monitoring the state of the objects through the information that is generated by the sensors as well as through the predefined alarms.

Telecom operators as platform providers are offering IoT platform with additional functionalities through several modules and support for several different protocols and standards. Also, they are enabling additional revenues through data analysis, external interfaces and a possibility of management of a large number of connected devices. Additional functionalities include efficient development of different types of innovative IoT services though connectivity with IoT platform.

Future work concerns the development of a digital ecosystem of advanced IoT solutions and services, so telecom operators can more easily create new service packages. Combining their products and services with innovative services of other providers and developers, operators are transforming their business operations and enhancing customer experience. The advantage of the usage of the IoT platform and digital ecosystem by providers and developers is faster development and delivery of IoT services to the market.

REFERENCES

[1] Newman, M., “TM Forum Digital Maturity Model (DMM): A Blueprint for Transformation", https://www.tmforum.org/wp-content/uploads/2017/05/DMM-WP-2017-Web.pdf, 2017

[2] Kechiche, S., “GSMA Intelligence — Research — IoT: the next wave of connectivity and services”, https://www.gsmaintelligence.com/research/2018/04/iot-the-next-waveof-connectivity-and-services/, 2018.

[3] Bains, K., Giles, M., Rogers, M., Wyrzykovski, R. and Kechiche, S., “GSMA Intelligence — Research — IoT: the $1 trillion revenue opportunity”. https://www.gsmaintelligence.com/research/2018/05/iot-the-1-trillionrevenue-opportunity/670/, 2018.

[4] Van den Dam, R., “IoT monetization by telcos: Hype or hope? - TM Forum Inform”, https://inform.tmforum.org/internet-of-everything/2017/10/iot-monetization-telcos-hype-hope/, 2017

[5] World Economic Forum and Accenture, “Digital Transformation Initiative Telecommunications Industry”, http://reports.weforum.org/digital-transformation, 2017

[6] Krotov, V., “The Internet of Things and new business opportunities”, *Elsevier Inc*., Business Horizons, vol. 60, no. 6, 2017, pp. 831–841.

[7] Dijkman, R.M., Sprenkels, B., Peeters, T. and Janssen, A., “Business models for the Internet of Things”, *Elsevier*, International Journal of Information Management., vol. 35, no. 6, 2015, pp. 672–678.

[8] IoT-Ignite, “Role of Telcos in Internet of Things”, https://devzone.iot-ignite.com/wp-content/uploads/2017/01/Role-of-Telcos-in-IoT.pdf, 2017.

[9] Akpakwu, G. A., Silva, B. J., Hancke, G. P. and Abu-Mahfouz, A.M., “A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges”, *IEEE,* IEEE Access, vol. 6, 2018, pp. 3619–3647.

[10] Ericsson, “Exploring IoT strategies”, https://www.ericsson.com/en/internet-of-things/trending/exploring-iot-strategies/, 2018.

[11] GSM Association, “IoT Security Guidelines Overview Document Version 2.0”, https://www.gsma.com/iot/iot-security-guidelines-overview-document/.

[12] TM Forum, “TR271 Monetizing Internet of Everything (IoE) CEM R18.0.1 - TM Forum”. https://www.tmforum.org/resources/standard/tr271-monetizing-internet-of-everything-ioe-r18-0-0-cem/, 2018

[13] Da Xu, L., He, W. and Li, S., “Internet of things in industries: A survey”, *IEEE, I*EEE Transactions on Industrial Informatics, vol. 10, no. 4, 2014, pp. 2233–2243.

[14] Zhong, Y., “GSMA Intelligence — Research — Outcome-based pricing in IoT: a high-risk, high-return bet”, https://www.gsmaintelligence.com/research/2018/11/outcome-based-pricing-in-iot-high-risk-high-return-bet/706/, 2018

[15] Anitha, A., “Home security system using internet of things”, *IOP Publishing*, IOP Conf. Ser. Mater. Sci. Eng., vol. 263, no. 4, 2017, 042026.

[16] Guth, J., Breitenbucher, U., Falkenthal, M., Leymann, F, and Reinfurt, L., “Comparison of IoT platform architectures: A field study based on a reference architecture”, *IEEE*, 2016 Cloudification of the Internet of Things, CIoT 2016, 2017, pp. 1–6..

Srđan Kokolj received master’s degree in information technology from the Monash University, Australia in 2007 and the specialist degree in electronic commerce from the Faculty of Organizational Sciences, University of Belgrade in 2018. His research interests include IoT technologies, IoT business models for telecommunication operators, and the development of IoT digital ecosystems.

Mirjana Stojanović graduated in the Computer Science and Information Technology study program at the Faculty of Electrical Engineering, University of Belgrade, in 1993. She finished the executive MBA (Master of Business Administration) program at COTRUGLI Business School in Belgrade in 2014. She is currently PhD student at the Faculty of Organizational Sciences, University of Belgrade. She has 24 years of experience in the ICT industry. Her current professional engagement and research interest are focused on the digital transformation of mobile operators.

Zorica Bogdanović is an associate professor at Department of e-business at Faculty of Organizational Sciences, University of Belgrade. She is the secretary of IEEE Computer chapter C-16 and a co-chair of the Seminar of IEEE Computer chapter C-16. Her professional interests include e-business, internet technologies and internet of things.

Božidar Radenković is a professor at Department of E-business Faculty of Organizational Sciences, University of Belgrade, Serbia. He is the chief of Laboratory of e-business, and the president of IEEE computer chapter C-16. His research interests include internet technologies, computer networks, cloud computing and internet of things.

Bracketing an Extremum: DataFlow Implementation of the Golden Section Search Algorithm in One Dimension

Pejić, Dragana; and Arsić, Miloš

Abstract: One of the pressing issues of not only mathematics, but other sciences too, has been the calculations of the extreme values of the functions that model the given problem. Finding a model that corresponds to the problem has almost always been easier than finding the solution of the model.

Over the years, many techniques have been discovered and one of them is the Golden Section Search algorithm which is both simple compared to many other techniques and efficient.

This algorithm has been implemented using various programming languages, mainly those that correspond to the ControlFlow paradigm. As a result of this, some of the efficiency of the algorithm has been lost due to the size of the datasets that have to be processed.

With the increasing speed with which data is collected, datasets and the number of equations that models consist of have grown exponentially. ControlFlow implementations of the Golden Section Search are unable to follow this growth without losing precious time on simple calculations.

Taking all of this into account, in this article we would like to present one of the solutions that improves the efficiency of the algorithm. That is the implementation of the Golden Section Search using a different paradigm – DataFlow. This implementation decreases the execution time of the algorithm, thus speeding up the process of finding the extrema of the given set of functions.

Index Terms: ***acceleration, algorithm, approximation, bigdata, controlflow, dataflow, fpga, golden section search, maximization, minimization, numerical method, optimization, quadratic functions, unimodality.***

1. Introduction

One of the problems that is encountered the most often in modern scientific research is the optimization problem, that is finding the solution of a large number of equations that model a natural system or a process. Considering the number of present equations, this isn't an easy task since a large amount of time is spent on calculation – on simple tasks such as multiplication and addition. Even though the algorithms for solving optimization problems might be easy to understand, their execution time is long, which makes them time inefficient. One of the reasons that cause time inefficiency is the fact that the easiest way to translate mathematical algorithms into code is by using one of the languages that belong to the ControlFlow paradigm which is often unsuitable for processing large quantities of data that these methods require.

Thanks to the advancement in technology, the increased speed in which information is collected and multiplied causes the growth of optimization problems in regards to the number of equations that constitutes them.

Likewise, the information that used to be stored on traditional media is becoming digital too. The new data stored in digital media went over 92% in 2002 while the size of this new data has been more than five exabytes [1]. It is only logical that this percentage has risen in the following years. A small part of total data is used for optimization problems, but considering the size of total data, it is evident that even that part can be considered BigData.

According to Fisher et al. [2], BigData means that data is unsuitable for handling and processing by most if not all current information systems and methods because not only is data too big to be loaded into a single machine but it also suggests that most traditional data analytics and mining methods may not be applied to BigData, at least not directly. The problems that occur while processing BigData have helped the development of a new programming paradigm – DataFlow paradigm – which was introduced as early as 70s [3] but gained traction with the advances in the enabler technology (software (OpenSPL) and hardware (FPGA)).

To illustrate the DataFlow approach to solving optimization and BigData problems as well as to show the benefits of the new paradigm, we will implement a widely spread numerical algorithm called the Golden Section Search that is used for

Manuscript received November 30, 2018.

Pejić, Dragana is with the Faculty of Mathematics, University of Belgrade, Serbia (e-mail: draganapejic93@gmail.com).

Arsić, Miloš is with the Faculty of Mathematics, University of Belgrade, Serbia (e-mail: milosharsic.matf.bg.ac.rs@gmail.com).

finding the minimum or maximum of a given function.

The Golden Section Search (GSS) is a numerical technique used for solving optimization problems whose solution is the value in which the extremum of various functions is reached. Due to its nature, this algorithm more often presents a way to solve a sub-problem rather than it is used as a main algorithm that will offer the final solution, which means that the GSS algorithm encountered in many fields of science. Because it is widely spread, it is important that it's time efficient and precise.

Primary use of the GSS algorithm is for solving the problem of function minimization or maximization of strictly unimodal functions by successively narrowing down the range where the extremum is known to exist. The technique maintains the function values of three points which distances form *the golden ratio*. This is where the name of the algorithm comes from. This algorithm presents the limit of Fibonacci search for a big number of function evaluations and it was discovered by Kiefer [4].

Even though this method is used for finding any extrema of a given function, we will only consider the case of minimization since the minimum of a function *f* is the maximum of the function *-f*, and vice versa.

As mentioned above, the GSS algorithm is a part of many scientific procedures. Therefore it is of crucial importance that the implementation of the GSS is time efficient.

Creativity in this work follows the path referred to as Specialization in [5].

2. Derivation of the algorithm

There are many numerical methods used for solving the minimization problem. However, none of them are flawless. Depending on the characteristics of the problem, some are better than others. If we're required to find a point in which a unimodal continuous function reaches its minimum, and either don't want or know how to use derivatives, one of the best methods is the Golden Section Search.

Unimodality can be defined in different ways, depending on the properties of the given function and which of its extreme values is a unique one – minimum or maximum. As mentioned before, a maximum of a function *f* is a minimum of a function *-f*, we will consider only minima.

According to Osborne and Kowalik [6], function *f* is unimodal on *[a, b]* if and only if *f* has only one stationary value on this segment. However, looking closely at the definition, it can be noticed that there are two disadvantages to this definition:

- functions that have inflexion point(s) with a horizontal tangent are not allowed
- the definition is meaningless when *f* is not differentiable on *[a, b]*

There are functions in both of those classes that reach extreme values where they are defined, we reject this definition and turn to Wilde [7] whose definition assumes neither differentiability nor continuity. Since we consider the minimal value of a function and Wilde is interested in maxima, we have reversed his inequalities. After that modification, the definition of the unimodal function is as follows: *function f is unimodal on the segment [a, b] if, for all x_1, x_2 that are in [a, b],*

$$x_1 < x_2 \Rightarrow ((x_2 < x\square \Rightarrow f(x_1) > f(x_2)) \wedge (x_1 > x\square \Rightarrow f(x_1) < f(x_2)))$$

where x is a point in which f attains its least value in [a, b].*

Following this definition, to check the unimodality of a function *f*, it's required to find the point *x** (which must exist) and check whether it satisfies the condition above. Due to this, Wilde's definition is can't be easily applied. Luckily, there exists a definition that's equivalent to it, but which is not purely theoretical. Brent [8] defines a unimodal function in the following way: *function f is unimodal on [a, b] if, for all x_0, x_1, x_2 that are in [a, b] holds*

$$(x_0 < x_1 \wedge x_1 < x_2) \Rightarrow$$
$$((f(x_0) \square f(x_1) \Rightarrow f(x_1) < f(x_2)) \wedge (f(x_1) > f(x_2) \Rightarrow f(x_0) > f(x_1)))$$

The proof of equality of definitions given by Wilde and Brent can be found in [ref_d3].

To paraphrase the given definition, unimodal functions have only one extremum on the given interval. Taking this into account, it is easy to see why one of the prerequisites for the GSS algorithm is unimodality of functions.

The other prerequisite of the method is the continuity of the function, thus let's consider function *f* over the interval *[a, c]* that is both unimodal and continuous. Since *f* is unimodal, there is only one point in which it achieves its minimum on *[a, c]*. We want to find the point in which function *f* has minimal value in *[a, c]*. A way to achieve this is by bracketing that point.

A minimum of a function is bracketed when there is a triplet of points *(a, b, c)* that satisfy the following relations:

$$a < b < c$$
$$f(b) < f(a)$$
$$f(b) < f(c)$$

We would like to bracket the minimum even further. To be able to do that, a new point *x* that is either in the subinterval *(a, b)* or *(b, c)* needs to be chosen. If we assume that the new point is in *(b, c)* and evaluate *f* in that point, there are two possibilities:

- if *f(b)>f(x)*, then the new triplet is *(b, x, c)*
- *if f(b)<f(x)*, then the new triplet is *(a, b, x)*

In both cases, the middle point of the new triplet is the abscissa whose ordinate is the best minimum achieved so far. We continue to bracket the minimum until we reach the required accuracy. The accuracy is usually provided by an individual executing the algorithm or running the program that presents the implementation of the GSS algorithm.

Considering the accuracy and the interval where function *f* are defined, what is left to be done is finding a suitable way of choosing the new point *x* based on the bracketing triplet *(a, b, c)*. Let ω be a relative position of the point b:

$$\omega = \frac{b-a}{c-a} \qquad 1-\omega = \frac{c-b}{c-a}$$

We can assume that the next trial point *x* is an additional fraction *z* beyond *b*,

$$z = \frac{x-b}{c-a}$$

Taking this into consideration, the length of the next interval is either $z + \omega$ or $1 - \omega$. Since we want to eliminate the worst case scenario, the optimal choice is to set these two lengths equal. Thus we get

$$z+\omega = 1-\omega \Rightarrow z = 1-2\omega$$

The new point is symmetrical with the point *b* in the original interval:

$$|b-a| = |x-c|$$

Based on the previous equation, we can deducted that the new point *x* belongs to the bigger of the two subintervals. To find the exact position of the point *x*, we need to consider the value ω and its origin. Presumable, the value omega is a result of applying the same strategy in the previous iteration. This suggests that if value *z* was chosen to be optimal, and implies that that was the case with the value ω before it. Taking this into account, we can see that the distance from *x* to *b* compared to the interval *(b, c)* is equal to the distance of *b* from *a* compared to the interval *(a, c)*. In other words,

$$\frac{x-b}{c-b} = \frac{b-a}{c-a}$$

or in relative units

$$\frac{z}{1-\omega} = \frac{\omega}{1}$$

When we combine this result with the optimal choice for point *z*, we get

$$z = 1-2\omega$$
$$\frac{z}{1-\omega} = \omega$$

As a result of solving this system of equations, we get the following quadratic equation

$$\omega^2 - 3\omega + 1 = 0$$

which solution is

$$\omega = \frac{3-\sqrt{5}}{2} \approx 0.38197$$

The optimal triplet of points *(a, b, c)* is such that the point *b* is a fractional distance of 0.38197 away from one end of the interval and 0.61803 away from the other end.

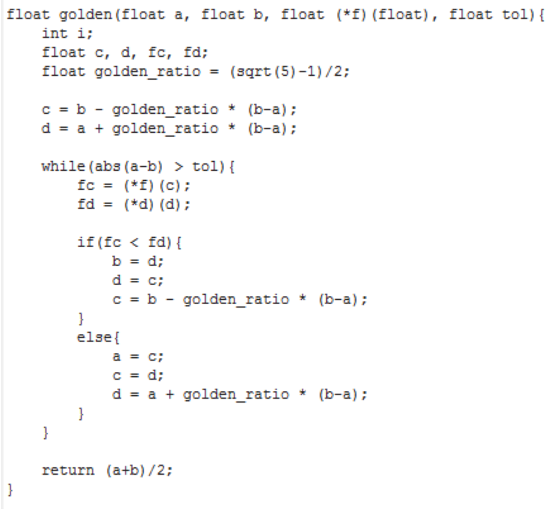
Taking everything above into consideration, the The GSS algorithm is as follows:

- the bracketing triplet of points *(a, b, c)* from the previous iteration is given at each stage
- find the new point *x* which is at the relative distance of 0.38197 from the point *b* in the direction of the longer subinterval
- the new triplet is the one that has the lower value at the midpoint
- return to the first step of the process and continue doing it until the given accuracy is achieved

The following sections contain the implementations of the GSS algorithm in two different programming paradigms.

3. CONTROLFLOW IMPLEMENTATION

After deriving the algorithm, we can see that it is quite straightforward and easy to follow. However, problems occur when we do it by hand since a lot of time is being wasted on doing simple calculations. To decrease the time that is needed to solve the optimization problem, we are going to translate the algorithm into a program that can be run on a machine. What comes to mind is using the programming language C since it is one of the most popular languages. It is representative of the ControlFlow paradigm. The following code shows the ControlFlow implementation of the GSS algorithm.



```
float golden(float a, float b, float (*f)(float), float tol){
    int i;
    float c, d, fc, fd;
    float golden_ratio = (sqrt(5)-1)/2;

    c = b - golden_ratio * (b-a);
    d = a + golden_ratio * (b-a);

    while(abs(a-b) > tol){
        fc = (*f)(c);
        fd = (*d)(d);

        if(fc < fd){
            b = d;
            d = c;
            c = b - golden_ratio * (b-a);
        }
        else{
            a = c;
            c = d;
            d = a + golden_ratio * (b-a);
        }
    }

    return (a+b)/2;
}
```

The function *golden()* is not part of the *main()* function of the program, but rather it is called from there. It requires four parameters – points *a* and *b* that are the end points of the interval where the minimum is known to exist, value *tol* that presents the accuracy that needs to be achieved, and a pointer to the function *f* which minimum we're seeking. This routine performs the GSS, isolating the minimum to the fractional precision of about *tol* and returning the abscissa of the point in which the minimum is reached as the value calculated as *(a+b)/2*.

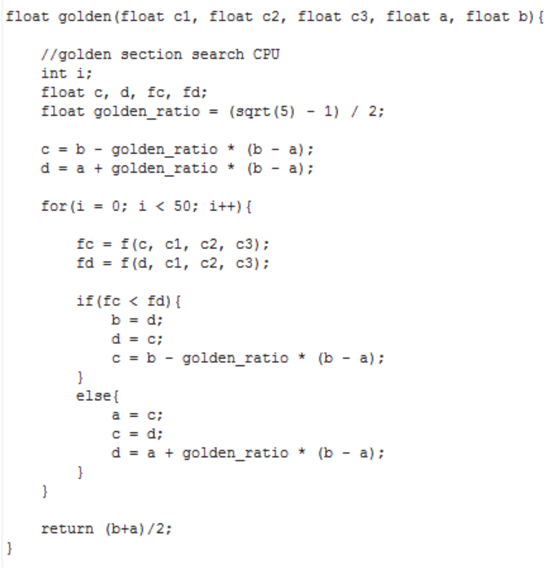
4. DATAFLOW IMPLEMENTATION

Looking at the ControlFlow implementation, we can see that the function value needs to be calculated almost countless of times. The calculating time depends on the nature of the function. Some require more time than others. To speed up the algorithm, we have decided to evaluate only quadratic functions. This can be done without the loss of generality because many, more complex, functions can be either reduced or approximated by quadratic functions.

To write suitable code, knowing the requirements of the dataflow application is crucial. A Maxeler supercomputer consists of CPUs and Dataflow Engines (DFEs), which means that a dataflow application is mostly made up of CPU code with small pieces of source code, and large amounts of data that runs on dataflow engines.

As a part of the CPU code, we define function *golden()* outside of the main function of the program. This function represents the implementation of the GSS algorithm where instead of calculating the value of the function in the routine *golden()* itself, we evaluate it by calling the function *f_val* that exists outside of the routine *golden()*.

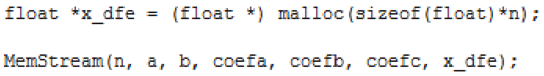
For the function *golden()* to run properly, we need to supply five arguments to it. Three arguments present the coefficients of the quadratic function which minimum we're looking for and the remaining two are the end points of the interval where the minimum is bracketed.



```
float golden(float c1, float c2, float c3, float a, float b){

    //golden section search CPU
    int i;
    float c, d, fc, fd;
    float golden_ratio = (sqrt(5) - 1) / 2;

    c = b - golden_ratio * (b - a);
    d = a + golden_ratio * (b - a);

    for(i = 0; i < 50; i++){

        fc = f(c, c1, c2, c3);
        fd = f(d, c1, c2, c3);

        if(fc < fd){
            b = d;
            d = c;
            c = b - golden_ratio * (b - a);
        }
        else{
            a = c;
            c = d;
            d = a + golden_ratio * (b - a);
        }
    }

    return (b+a)/2;
}
```

The difference between this snippet of code and the one presented in the previous section is that the *while* loop has been replaced with a *for* loop. Due to the importance of loop unrolling in the DataFlow implementation and since unwinding a *for* loop is much easier than unwinding a *while* loop, this change has been made. The *for* loop has a maximum of 50 iterations which is an estimated number of iterations needed for achieving the greatest precision of the calculation of the minimum of quadratic functions. Since the number 50 is arbitrary, it can be replaced with any number that the end user of the application deems appropriate.

Inside the CPU code, we pass a call to the Kernel where the code that will be projected in space is located. The name of the project and the function is *MemStream*. This function has six arguments – *n* is the number of functions whose minimum we want to evaluate, *a* and *b* present the end points of the interval where the minimum is located, *x_dfe* is an array where the abscissa values of the found minima will be saved while the remaining three arguments present coefficients of the functions. The following code presents the fraction of Kernel code that demonstrates the call of the *MemStream* function.

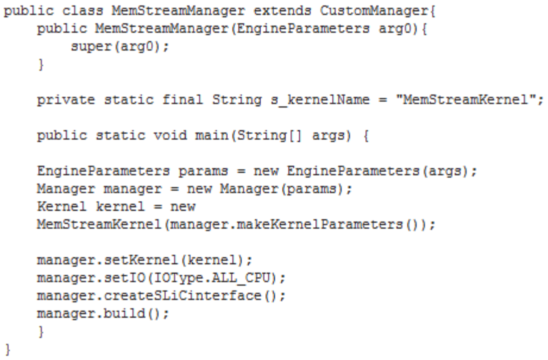


```
float *x_dfe = (float *) malloc(sizeof(float)*n);

MemStream(n, a, b, coefa, coefb, coefc, x_dfe);
```

The code that runs on the dataflow engine is written in MaxJ which is subset of Java. To describe it, we use a Java library, and to execute actions of the DFE (including sending sets of parameters and data streams to the DFE), the Simple Live CPU (SLiC) API functions need to be called. To use these functions, headers “Maxfiles.h” and “MaxSLiCInterface.h” need to be included.

After writing the MaxJ code, it is executed to generate the *.max* file (the dataflow implementation). After being generated, this file can be linked and called via the SlIC interface. A *.max* file consists of Kernel and Manager files.

The Manager acts as a bridge between the Kernels and the CPU code. It defines data streams between the accelerator and the host program as well as types of variable in those streams. The Manager code is used for creating the Kernel. The code below shows an example of a Manager file.

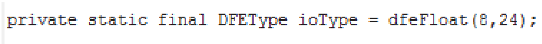


```
public class MemStreamManager extends CustomManager{
    public MemStreamManager(EngineParameters arg0){
        super(arg0);
    }

    private static final String s_kernelName = "MemStreamKernel";

    public static void main(String[] args) {

    EngineParameters params = new EngineParameters(args);
    Manager manager = new Manager(params);
    Kernel kernel = new
    MemStreamKernel(manager.makeKernelParameters());

    manager.setKernel(kernel);
    manager.setIO(IOType.ALL_CPU);
    manager.createSLiCinterface();
    manager.build();
    }
}
```

The Manager describes the data flow choreography between the DFEs memory, Kernels and various interconnects depending on the dataflow machine. By using Managers, high utilization of available resources such as memory bandwidth and arithmetic components is achievable. This is done by decoupling communication and computation, and using a flow model for off chip I/O to the CPU, DFE interconnects and memory. Maximum performance on a Maxeler dataflow computer is achieved through a combination of exploiting both inter- and intra-Kernel parallelism as well as deep-pipelining. The high I/O-bandwidth required by such parallelism is supported by flexible high-performance memory controllers and a highly parallel memory system [9].

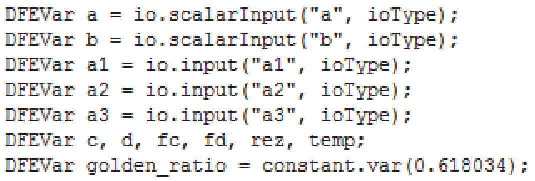
Kernels are graphs of pipelined arithmetic units which can be arithmetical or logical. If a dataflow graph doesn't contain a loop, data simply flows from inputs to outputs. If loops are present, data flows in a physical loop inside the DFE in addition to flowing from inputs to outputs. The execution of the computation is efficient as long as there is a lot more data than there are stages in the pipeline.

The time needed to compute the result depends on the type of arithmetic operations that need to be executed since not all operations have the same running time. During the time that's needed for computing, the accelerator is idle. To maximize the use of the accelerator, we need to shorten the computation time. To achieve this, we change the floating-point number representation to a fixed-point one. When a custom number representation is used, the computing time will often decrease [10]. Our fixed-point number representation is as follows:



```
private static final DFEType ioType = dfeFloat(8,24);
```

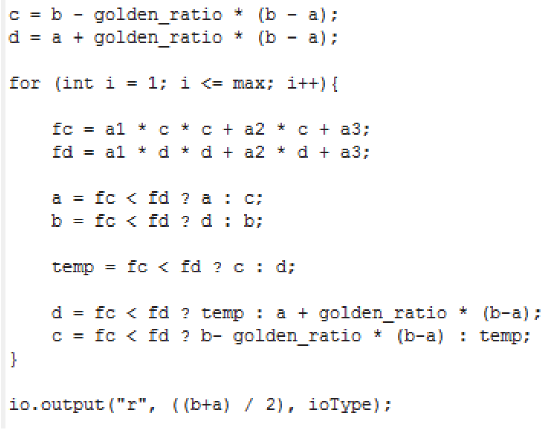
Since our host program is based on the ControlFlow paradigm, it doesn't allow a fixed-point number representation, that is all number variables are floating-point. When these variables arrive to the accelerator, they are converted to the fixed point type defined above. After this change, all operations perform computation with the fixed-point numbers. When the final result is reached, it is converted back to the floating-point type before it is returned to the host program. The code below show the transfer of data from the host program.



```
DFEVar a = io.scalarInput("a", ioType);
DFEVar b = io.scalarInput("b", ioType);
DFEVar a1 = io.input("a1", ioType);
DFEVar a2 = io.input("a2", ioType);
DFEVar a3 = io.input("a3", ioType);
DFEVar c, d, fc, fd, rez, temp;
DFEVar golden_ratio = constant.var(0.618034);
```

Due to the nature of the GSS algorithm, the differences between the ControlFlow and DataFlow implementations aren't drastic. The *while* loop is replaced with a *for* loop with a definite number of iterations that each can be unrolled, i.e. definite number of times replicated on the 2D card.

The code presents the essence of the DataFlow implementation. It demonstrates how the algorithm is executed after data has been transferred from the host program.



```
c = b - golden_ratio * (b - a);
d = a + golden_ratio * (b - a);

for (int i = 1; i <= max; i++){

    fc = a1 * c * c + a2 * c + a3;
    fd = a1 * d * d + a2 * d + a3;

    a = fc < fd ? a : c;
    b = fc < fd ? d : b;

    temp = fc < fd ? c : d;

    d = fc < fd ? temp : a + golden_ratio * (b-a);
    c = fc < fd ? b- golden_ratio * (b-a) : temp;
}

io.output("r", ((b+a) / 2), ioType);
```

5. Performance Evaluation

For choosing a perfect algorithm for solving problems, it should be taken into account not only its complexity but also its execution time which depends on the amount of data that needs to be processed. It is true that technology is developing fast and with it, the computer architecture is changing too. But often this growth isn't capable of following the speed with which data is multiplied and collected. Which is why the execution speed of algorithms is still important.

Due to this reason, comparison of execution times of two implementations of an algorithm in languages that are representative of two different paradigms is a good way to compare them. In the case of the GSS algorithm, the paradigms in question are ControlFlow and DataFlow. The DataFlow parasigm has shown better performance when it comes to analyzing BigData and thus offers an alternative to the ControlFlow paradigm.

To compare these two paradigms, we will measure the execution time of its respective implementations on the same testing set. Without the loss of generality, we have chosen a set of quadratic functions with the minimum in the interval *(0, 1000).* All quadratic functions are unimodal which presents one of the prerequisites of the GSS algorithm, and many other functions can either be reduced to quadratic functions or approximated with them.

The general form of the quadratic function is

$$f(x) = ax^2 + bx + c$$

where coefficients *a, b, c* are real numbers and *a ≠ 0.*

The vertex of a function is the point in which it reaches its minimal or maximal value. In our case, we're looking for the minimum α. To calculate it, we can use the following formula

$$\alpha = \frac{2a}{b}$$

To make sure that our minimum is in the desired interval, we require that it satisfies the condition

$$0 < \frac{2a}{b} < 1000$$

Since the coefficient *c* isn't used for the calculation of the minimum, it can be a randomly chosen number.

If we choose the coefficient *a* similarly, then the only restriction for the remaining coefficient *b* is the condition

$$\frac{a}{500} < b$$

Since we can generate as many random numbers as we want, then we can generate enough test functions whose coefficients satisfy the conditions above that their amount can be considered BigData. These functions are used for testing our implementations.

For solving BigData problems, the DataFlow paradigm relies on the hardware acceleration; that is, the way the programs are written is to configure the hardware. When an application is written this way, it's partly executed on the accelerator and partly on the host computer. This way the CPU nodes can use as many DFEs as are needed for the application that is being executed while the remaining DFEs are free to be used by other nodes that are not busy running this particular computation. Optimal balance of all cluster resources at runtime is achieved this way.

We have used the MAX4 Card which is contained in the MPC-X2000 node to measure the usage of block memory, multipliers, and logic utilization. The evaluation is shown in the Figure 1.

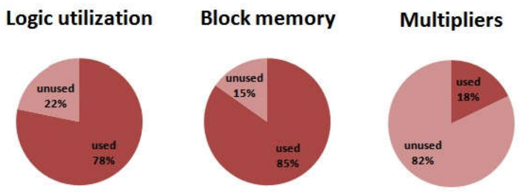


Figure 1: Usage Evaluation of MAX4 Card

To measure the execution time of our implementations, we have run the application on datasets of varying sizes with the smallest being just one function and the biggest sample size having sixty million functions. The results are shown on Figure 2.

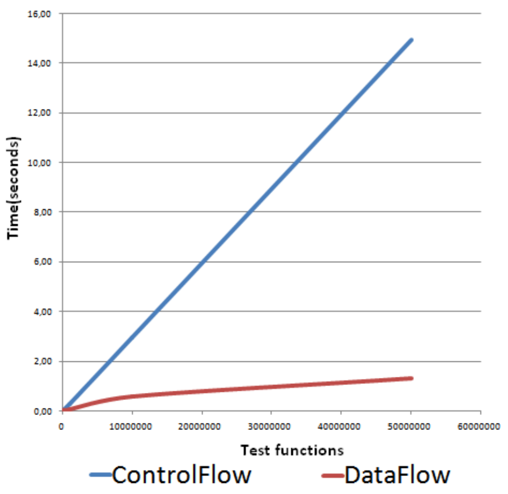


Figure 2: Execution speed of the Golden Section Search algorithm

For test examples where the number of functions is small, there is no drastic difference in execution time. With the increase of sample size, the difference becomes not only noticeable, but quite remarkable. When the number of test functions reaches one million, for the first time we can see that the DataFlow implementation is faster than the ControlFlow one. With the increase of the number of functions in the sample, this difference becomes more and more apparent. Using the MAX4 Isca @ 200MHz card over the Intel(R) Core(TM) i5-3350P CPU @ 3.10GHz microprocessor, the DataFlow implementation achieves acceleration of 11.5 times.

Other than dataflow and controlflow computers, clusters can process BigData as well. A cluster is a set of connected computers that work together and can be considered a single system in many ways. Because clusters present some of the fastest supercomputers in the world, we decided to test our application on them as well. The cluster we did the testing on is with Intel(R) Core(TM) i7-6670P CPU @ 4.00GHz microprocessor. When we compare the execution time of our application on a cluster with the time measured on a regular computer, the cluster is faster. Comparing that same time with the time achieved on the MAX4 Isca @ 200MHz card, we get different results shown on Figure 3.

Running our application on the dataflow computer is around 11.24 times faster than running the same application of the cluster. The speedup is lower compared to the one gotten when we compared regular and dataflow computers, but that is because a cluster is faster than a regular computer.

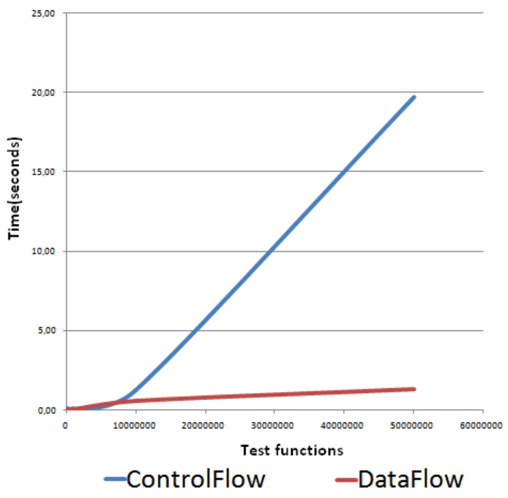


Figure 3: Execution speed of the Golden Section Search algorithm on a dataflow computer and a cluster

Looking at the results, the DataFlow implementation has shown better performance than the ControlFlow implementation of the GSS algorithm.

I. 6 . CONCLUSION

After a careful examination of the previous sections, it can be deducted that the execution time of an algorithm on a digital computer is not only dependent on the test examples and the amount of data that needs to be processed, but that the programming paradigm that the algorithm is implemented in plays a crucial role as well. Based on the results presented in the section *5. Performance Evaluation,* we can conclude in some cases it is worthy trading one of the traditional, familiar paradigm for a new, not widely spread paradigm because the change of approach and tactics when it comes to implementing the chosen algorithm can result in remarkable speedup.

In its essence, the Golden Section Search is a numerical method that is used for bracketing an extremum of a single function or a given set in one or more dimensions without using the function derivatives. Its application in one dimension was the focus of this article in order to showcase the power of the different programming paradigms, as well as to offer an implementation that can be an alternative to the ControlFlow implementations most often encountered in numerical analysis and other fields of science. The GSS algorithm has been in use decades before a digital computer became an important tool used by researchers. Despite its rather simple structure, the execution of it has been time consuming. In order to speed it up and save valuable time, this algorithm can be executed on a digital computer without facing significant changes in its structure.

The implementation of GSS algorithm can be done using representative languages of different programming paradigms. One of the widely spread paradigms is ControlFlow paradigm due to its almost intuitive approach to the implementation of various algorithms. Without too much trouble, any algorithm can be almost linearly translated into a language that is exemplary of the ControlFlow paradigm. However, due to the development of technology, the increasing speed in which information is created, and the growth of mathematical models that expound the problems encountered while doing research, we believe that there will come a time when the ControlFlow implementation of this algorithm won't be enough to offer solutions that are time efficient, which is why a different implementation which utilizes the benefits of the DataFlow paradigm is presented.

Summarizing the testing results presented in section *5. Performance Evaluation*, the following can be derived.

When we compare the codes of the two implementations, it's easy to tell that they aren't too different. This observation is a direct consequence of the GSS algorithm containing just one loop that must be unrolled. If our selected algorithm were to contain more loops, the differences in code would be bigger. To transform code from one implementation to the other, all that had to be done was finding a good way to change the *while* loop from the ControlFlow implementation to an equivalent *for* loop in the DataFlow implementation. More details on this can be found in appropriately titled section *4. DataFlow Implementation.*

Comparing the measured running time of the program using sets of examples of different sizes, going from one to sixty millions, that are derived the same way, it becomes apparent that the DataFlow implementation of the GSS is faster not only when it's run on a regular digital computer, but also on a cluster. Calculations show that the DataFlow is around 11.5 times faster compared to the ControlFlow implementation when we run them on a regular computer and 11.24 faster when they're run on a cluster. This speedup is achieved when the size of tested data is big enough to be considered BigData. The number is lower when it comes to cluster for the simple reason that a regular computer is slower than a cluster.

The crucial reason why ControlFlow is so much slower is because a big amount of time is spent on transferring data to and from memory. In the case of the DataFlow implementation, data transfer doesn't present a problem since the data is being streamed.

In conclusion, the DataFlow implementation of the GSS algorithm is more time efficient compared to the ControlFlow implementation of the same algorithm when the size of datasets that need to be processed is large enough to be considered BigData. With the fast accumulation of data, we believe that in the near future, the DataFlow implementations of not only numerical algorithms but others as well would be proven to be more efficient than their ControlFlow counterparts since time for transferring data to from memory would be reduced if not eliminated in DataFlow implementations due to the data streaming.

References

[1] Lyman, P., Varian, H., "How much information 2003?," *Tech. Rep*, 2004

[2] Fisher, D., DeLine, R., Czerwinski, M., Drucker, S., "Interactions with big data analytics," *Interactions*, 19(3): 509, 2012

[3] Hurson, A., Lee, B., "Issues in dataflow computing," *Advances in Computers,* vol. 37, 1993, pp. *285-333*

[4] Kiefer, J., "Sequential Minimax Search for a Maximum," *Proceedings of the American Mathematical Society*, vol. 3. no. 3, 1953, pp. 502-506

[5] Blagojevic, V., et al, "*A Systematic Approach to Generation of New Ideas for PhD Research in Computing*," Advances in Computers, Elsevier, Vol. 104, pp. 1-19, 2016

[6] Kowalik, J., Osborne, M.R., "Methods for unconstrained optimization problems," *American Elsevier Publishing Company*, 1968

[7] Wilde, D.J., "Optimum Seeking Methods," *Prentice Hall*, 1964

[8] Brent, R.P., "Algorithms for Minimization without Derivatives," *Prentice Hall*, 1973

[9] Maxeler, "*Multiscale Dataflow Programming,*" 2013

[10] Fu, H., Osborne, W., Clapp, B., Pell, O., "*Accelerating Seismic Computations on FPGAs From the Perspective of Number Representations,*" Rome, 2008

Toward Automatic Tagging of Cultural Heritage Documents

Tanasijević, Ivana

Abstract: ***The presented system provides intelligent suggestions of terms for tagging cultural heritage documents based on accompanying protocols. This system was made as an improvement of a previously designed system for organizing and manually tagging cultural heritage multimedia collection. The collection consists mainly of documents in the form of audio and video interviews with accompanying textual protocols. The labels for tagging are metadata (date and location of the*** record, participants in the conversation, language, ***ethnicity, and religious affiliation of the interlocutors), as well as the topics discussed. Another system was developed using natural language processing techniques for automatic context-based extraction of metadata and topics from those textual protocols. The system presented here combines these two systems to intelligently offer possible solutions for labels, as a help for a user who used to enter these terms manually. The system preselects appropriate choices, gives an option for additional corrections if necessary, and associates final tagging with the current document. This provides semi-automatic document tagging, because it uses automatic intelligent suggestions, but the user has to validate choices with a possibility for corrections.***

Index: Tagging Documents, Automatization of Tagging, Natural Language Processing, Cultural Heritage

1. Introduction

In the era of digitization and development of new technologies, a large amount of digitized material has to be stored and efficiently searched. Storage in an organized form is of great importance as it enables better search and retrieval of desired material. For this purpose, various types of databases that already have a developed management system for working with different types of data can be used. Working with databases without help of some additional applications that make it easier to insert, search and retrieve, can be very complicated and requires a lot of invested effort. Documents usually have some characteristics that can be obtained by insight into their content. This process is time-consuming if it is done manually. In the case of documents from a specific domain that is a subject of a research, it is possible to summarize characteristics of those documents that are relevant to that specific domain. Documents are then associated with some of these characteristics and this information is also stored in the database. Then, they can be searched and retrieved by those specific characteristics for reading only of for some further analysis. The problem can be solved by creating an infrastructure that provides comfortable work and automates the process of document tagging and inserting into the database together with the associated tags. Anything that automates inserting and retrieving of the documents allows a user to work with other, more advanced, analyses and tasks for which human interaction is still necessary.

An application prototype will be presented here that will combine two previously created applications, in order to automate the process of tagging documents of cultural heritage with information of interest. The first of the two applications is described in [1] and deals with organizing documents into a database by marking the information of interest for the domain of cultural heritage. The relevant data are in textual protocols in a form of free, unstructured text, so the process of finding information, by which documents should be labeled, has been left entirely to the person performing the tagging. It was necessary for the person to read the protocols, analyze the content, find the relevant data, and manually enter the data in order to create the appropriate tags. After tagging and storing, documents can be searched by tags that are assigned to them, as well as retrieved for further work. This greatly facilitates the task of finding, reviewing and analyzing the documents.

It has been noticed that there is certain regularity in the appearance of data of interest in those protocols in the meaning of context in which they appear, or the data itself are characteristic (for example, dates, locations, names, and surnames of persons). This led to another application that was made (described in [2]) that searches for relevant data and inserts tags into text according to content and some characteristic data. It carries out an automatic analysis of the protocol using natural language processing methods, makes labeling of the data, classifies them into appropriate categories and provides their extraction. This enables protocols containing some specific terms to be retrieved by such (automatically) extracted terms. When a protocol is specified, all the relevant already labeled data from the protocol can be extracted, which greatly improves the work of the person who had done it manually earlier, reading the text itself and extracting manually terms of interest.

The prototype of the application that is presented here combines the functionality of the two applications by providing extracted terms from a protocol as

Manuscript received December 5, 2018. This work was supported in part by "Infrastructure for Electronically Supported Learning", No. III47003, Project of the Ministry of Education, Science and Technological Development od Republic of Serbia. Author affiliation: Faculty of Mathematics, University of Belgrade (e-mail: ivana@matf.bg.ac.rs).

relevant terms that can be associated with documents related to that protocol. It is now on a person/user to select a protocol for a particular document, to let the system automatically assign tags, and to check if tagging is correct, to correct or add new tags that were not detected, or there was no information about them, and to complete tagging and storing the document. This application provides an intelligent suggestion of terms by which labeling should be performed. Although the user continues to read the protocol and checks if those terms are relevant and well formed, now he/she does not have to manually search for all the terms, since they are also marked in the original text, so they are easily visible in the text. In addition, he/she does not have to manually enter them, as they have already been forwarded to the fields for entering the relevant terms. The solution for this process will be described in this paper.

This text is organized as follows. The second chapter gives an overview of related works. The third chapter describes the motivation for the development of this system. Chapter four describes the methodology for solving the task of combining existing tools for the purpose of smart suggestions. In chapter five implementation of the used system is presented. Chapter six provides conclusion and ideas for further improvements.

2. Related Works

The task of annotating documents of cultural heritage exists for many years. A lot of effort has been invested in this task. Text document annotation can refer to the marking of the content in a particular text, but it can also associate with certain tags of the documents. The solution for the first task is proposed by [2], while the solution for the second task is given by this paper.

There are existing solutions for suggestion of tags that can be used for labeling documents. One such solution is presented in [3]. Paper [4] describes a content-based tag recommendation method that can be applied to any text document. Paper [5] presents human-competitive tagging using automatic keyphrase extraction. Proposed methods mostly use extraction by some machine learning techniques to give the possible predictions. In order to use the statistical methods of machine learning, a greater amount of texts are needed. In such a situation as with the collection of the documents we deal with in this paper, there is not enough material for such methods in order to be able to get a statistically generated solution that yields competitive results.

A different methodology would be semantic. With semantic methods the data are extracted based on the meaning, with possibility to have insights to a context, rather than based on a statistics that indicates what appears more frequently in a particular context. Semantic methods are more accurate, but also more demanding, because they request more human interaction for generating rules by which those data can be found in texts. Different approaches for natural language processing are given in [6]. One approach of document retrieval by semantic methods in specific domain of geological projects is given in [7]. In the case of Serbian language, there is no such solution for the cultural heritage domain.

The solution presented here incorporates application in which tags are extracted by semantic rule-based methods rather than by statistical methods. The used methodology enables work with domain-specific unstructured texts that significantly differ from average texts, or texts from different specific domains, with respect to type of labels and the structure of text constructions that are in use. So, this demands domain-specific solution for its managing.

This work presents an idea for a new solution [8] of improving the process of tagging documents with help of natural language system developed previously. Existing solutions for Serbian texts for specific domains stop within the phase of annotating and extracting data from text, while this solution provides an infrastructure to use the results of extraction in order to complete the process of tagging of documents from cultural heritage domain multimedia collection.

3. Motivation

A multimedia collection of documents on cultural heritage was collected as a result of many years of field research performed by researchers from the Balkan Institute of the Serbian Academy of Sciences and Arts. The collection consists of audio and video materials in the form of interviews with local people. There are also numerous photos and research papers related to this collection. Each of the audio and video material has a follow-up textual protocol in Serbian that contains various information about interviews that will be described below.

The aim of this field research was to study the language characteristics of languages/dialects used at specific locations in the Balkans. Mostly, it is about dialects that were previously in use, so they are significant for the purpose of preserving knowledge about those that are potentially disappearing. Materials, in addition to this, are also abundant with information about various customs that are still practiced in some parts of the Balkans, or are still living more or less in the memory of people as practices that were once used. There are various other topics from the lives of people, personal and collective memories, the way of life, the way of seeing and meeting life situations, as well as other stories from private lives. All of this is part of the cultural heritage of people in this region and as such has great value and contribution to the preservation of the identity of community and individuals in it.

The protocols provide basic information on where the material was recorded, the date on which it was recorded, who the participants are, and their roles - informers, researchers or others who only attend the interview, possibly the ethnicity of the informers, religious affiliation, the language in which the interview is conducted, as well as other data such as personal data, locations mentioned in their stories. These data will be called metadata. In addition, protocols contain some information about the flow of the conversation. Since protocols are not of a uniform structure, the flow of conversation can be found in the form of a list of topics being discussed or as a kind of free text that describes the flow of conversation. In some protocols, this can be described in the form of an informal transcript of conversations.

The collection was created by many researchers over time. All documents were organized through a system of directories and files without a single (unified) rule. As the collection was getting bigger, it was more difficult to handle it. On the other hand, this is a collection of valuable information about the cultural heritage, so there was the idea to organize this information in the way that can be more easily displayed to the interested community. These motives have led to the establishment of cooperation for development of applications for organizing, marking, searching and presenting this collection of documents. This application was created and described in [1]. It has graphical user interface (GUI) that enables selecting of audio and video documents, images, and papers and labeling these documents with metadata and topics discussed. These metadata and topics have to be entered manually according to the protocols related to the document.

Another application that was created was responsible for automatically searching for data in protocols. In addition to the data labeling, it is possible to search and retrieve protocols by the labels it contains. For example, all the protocols of one researcher that were made in a specific location can be retrieved, or all of those protocols in which a particular topic is discussed. The description of this application is given in [2].

The metadata marked are stamps, persons (informers, researchers, others), locations where the materials were recorded, locations mentioned in stories, dates, languages, ethnicities, religions, topics discussed, and others. There is a huge range of topics (can be found in [9] and [10]), and for the purpose of our research, we covered topics such as: house (houses, households, and farms) and national economy (domestic handicrafts, hunting and fishing, beekeeping, agriculture, mining, forestry, trade, and crafts).

4. *Methodology*

The application presented here is provided as a support for the document tagging and storing. It uses extracted terms from the textual protocols and gives intelligent suggestions that can be used for tagging documents. Data of interests can be metadata and topics discussed, as previously mentioned in the text, so the methodology for both of them will be given.

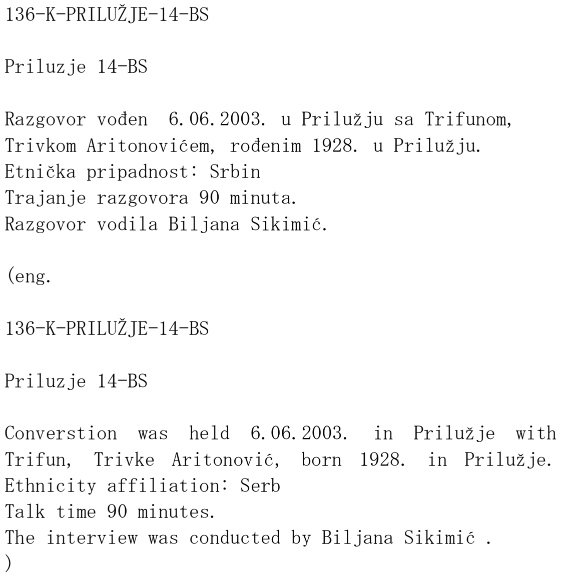
A. Metadata

Metadata can be simple and complex. Simple metadata are those that consist only of text, such as a stamp or date. Complex metadata are those that consist of an ordered pair (index, value), where the index is a unique identifier within that class of metadata, while the value is a term or construction of terms referred by that index. In this way, complex metadata can be used to label more than one document, so the search and retrieval of a document based on that value of the specific class of labels can be performed. Examples of complex metadata are people (classified by their roles), locations, ethnicities, religions. and languages.

The protocols are first transferred to the application described in [2]. Outcomes are the protocols with embedded XML tags. Metadata is further processed in two ways, depending on whether they are simple or complex. Simple metadata is only automatically entered into the fields provided for text input in the application described in [1]. The user can accept or correct them, and as such they can be attached to the document. They can also be rejected if the suggestion is not adequate.

Complex metadata, on the other hand, first have to be entered into the database so they can be used for other tagging. The system automatically makes value suggestions for these categories of metadata, also based on protocols with embedded XML tags, as in the case of simple metadata. When entering a new value for a given category, it is allowed to reformulate it, modify, and enter into the database or reject. The system then automatically assigns a unique index to them. Once entered, the value appears in a list of possible terms from that category of metadata, and then can be used for labeling this and other documents. For example, a researcher must be entered first time, during the first tagging, but can be found in more than one interview, which is always the case. Figure 1 depicts an example of a part of a protocol in which general information about the interview is given.

a)



```
136-K-PRILUŽJE-14-BS

Priluzje 14-BS

Razgovor vođen  6.06.2003. u Prilužju sa Trifunom,
Trivkom Aritonovićem, rođenim 1928. u Prilužju.
Etnička pripadnost: Srbin
Trajanje razgovora 90 minuta.
Razgovor vodila Biljana Sikimić.

(eng.

136-K-PRILUŽJE-14-BS

Priluzje 14-BS

Converstion was held 6.06.2003. in Priluže with
Trifun, Trivke Aritonović, born 1928. in Priluže.
Ethnicity affiliation: Serb
Talk time 90 minutes.
The interview was conducted by Biljana Sikimić .
)
```

b)

```
<div>
<stamp>136-K-<context p=″location″> <location
p=″toponym″>PRILUŽJE<oo lemma=″Prilužje″/>
</location></context>-14-BS</stamp> <stamp>Priluzje
14-BS</stamp></div>
<div p=″unknown″>
<unknown>Razgovor vođen</unknown>
<context p=″date″> <date>6.06.2003.<oo
lemma=″6.06.2003.″/> </date></context>
<context p=″location″>u <location
p=″context″>Prilužju<oo lemma=″Prilužju″/>
</location></context>
sa<person p=″informer″> Trifunom, Trivkom
Aritonovićem</person>,
<context p=″year″><year>rođenim 1928.<oo
lemma=″rođenim 1928.″/></year></context>
<context p=″location″>u <location
p=″context″>Prilužju<oo lemma=″Prilužju″/>
</location>.
<context p=″ethnicity″>Etnička pripadnost: <ethnicity
p=″context>Srbin<oo
lemma=″srpski″/></ethnicity></context>
</div>
<div>Trajanje razgovora 90 minuta.</div>
<div p=″researcher″><researcher>Razgovor
vodila</researcher><person p=″none″>Biljana
Sikimić</person>.
</div>
```

c)

```
--- Stamps---
[136-K-PRILUŽJE-14-BS]
[Priluzje 14-BS]

--- Informers---
Trifunom, Trivkom Aritonovićem

--- Researchers---
Biljana Sikimić

--- Locations ---
Prilužje

--- Dates ---
6.06.2003.

--- Years ---
rođenim 1928.

--- Ethnicities ---
srpski
```

Figure 1. Labeling and extraction of metadata

Figure 1.a) denotes the beginning of a protocol in which the basic information about the interview is provided. Figure 1.b) shows how the XML tags are embedded in this text by automatic tagging of metadata. Figure 1.c) consists of the metadata extracted from this XML text. The data that has been extracted is not always normalized, such as the name and surname of the informer. For example, "Trifunom, Trivkom Aritonovićem" is not given in its basic form, so this information has to be modified before it is entered into the database (in this example, "Trifun, Trivko Aritonović"). Only the data contained in the linguistic dictionary [11] is normalized, while the other data derived according to the context are mostly given in a non-normalized form. For example, the context of the location is "u (eng. 'in') [word with capital letter]", the context of a person is "sa (eng. 'with') [two words with a capital letter]", the context of the year is "rođen (eng. 'born') [year]". So, the structure of the sentence "Razgovor vođen 6.06.2003. u Prilužju sa Trifunom, Trivkom Aritonovićem, rođenim 1928. u Prilužju." (eng. 'Conversation was held 6.06.2003. in Prilužje with Trifun, Trivke Aritonović, born in 1928. in Prilužje.') is "Razgovor vođen <date> <context location> <context person>, <context year> <context location>. From that structure terms for date, locations, person and year can be derived.

B. Topics

Topics discussed combine characteristics of a complex and simple metadata. They represent an ordered four-tuple (domain, sub-domain, normalized value, free text). For example, in the first sentence for the Figure 2, the domain is the "national economy", the sub-domain is "agriculture", the normalized value is "tool", while the free text is "terminology of agricultural tools". Normalized value can be omitted if it is unknown. These values must first be entered into the database, in order to appear as options for tagging the document. Examples of some parts of the protocols with topics are given in Figure 2.

a)

```
Terminologija poljoprivrednih alatki i opreme za
rezanje.

Sagovornik Stantić pripoveda o životu salašara i
predstavlja svoj rad na istraživanju salaša.

Detinjstvo na salašu: od malena se nauči raditi

došli prvi kombajni i prestala je ručna žetva

anegdota o vinogradu

Živana: priča o zdravicama; Draga: kako je tkala;
kako je pravila crepulju; Živana: o crepulji; Draga:
kakav je narod sada; kako je nekad izgledala kuća
unutra, termini pokućstva;

(eng.

Terminology of agricultural tools and cutting
equipment.

Interlocutor Stantic talks about the life of the
farmers and presents his work on the farm research.

Childhood in the farm: learn how to work from a
childhood

came the first combines and the manual harvest
stopped

anecdote about vineyard

Ž ivana: a story of toasts; Draga: how she was
weaving; how she was making the crepe; Ž ivana: about
the crepe; Draga: what kind of people are now; how
used to look like a house inside, the terms of
furniture;
)
```

b)

```
<topic p="nationalEconomy" pp="agriculture">
Terminologija poljoprivrednih <oo
lemma="poljoprivredno"/> alatki<oo lemma="alatka"/>
</topic> i opreme za prezanje.

Sagovornik Stantić pripoveda <topic p="house"
pp="farm">o životu salašara<oo
lemma="salašar"/></topic> i predstavlja svoj rad na
istraživanju salaša.

Detinjstvo <topic p="house" pp="farm">na salašu<oo
lemma="salaš"/></topic>: od malena se nauči raditi.

došli prvi kombajni i prestala je ručna <topic
p="nationalEconomy" pp="agriculture">žetva <oo
lemma="žetva"/></topic>

anegdota <topic p="nationalEconomy"
pp="agriculture">o vinogradu<oo
lemma="vinograd"/></topic>

Živana: priča o zdravicama; Draga: <topic
p="nationalEconomy" pp="domesticHandycrafts">kako je
tkala<oo lemma="tkati"/></topic>; kako
je <topic p="house" pp="households">pravila
crepulju<oo lemma="crepulja"/></topic>; Živana:
<topic p="house" pp="households">o crepulji<oo
lemma="crepulja"/> </topic>; Draga: kakav je narod
sada; <topic p="house" pp="house">kako je nekad
izgledala kuća<oo lemma="kuća"/></topic>unutra,
<topic p="house" pp="households">termini pokućstva<oo
lemma="pokućstvo"/></topic>
```

c)

```
--- Topics ---
nationalEconomy, agriculture:
[poljoprivredno][alatka] terminologija
poljoprivrednih alatki

house, farm: [salašar] o životu salašara

house, farm: [salaš] na salašu

nationalEconomy, agriculture: [žetva] žetva

nationalEconomy, agriculture: [vinograd] o vinogradu

nationalEconomy, domesticHandycrafts:  [tkati] kako
je tkala

house, households: [crepulja] pravila crepulju

house, households: [crepulja] o crepulji

house, house: [kuća] kako je nekad izgledala kuća

house, households: [pokućstvo] termini pokućstva
```

Figure 2. Labeling and extraction of topics

In Figure 2.a) parts of the sentences from the text dealing with the topics analyzed in [2] are presented. In Figure 2.b) are given processed protocols with embedded XML tags. Figure 2.c) represents separated four-tuples (domain, sub-domain: [normalized value] free text). In a panel for entering new terms, proposed terms are automatically added so the user can check, edit and enter or reject them. Similar to the complex metadata, these values can further be used to label other documents if other documents contain appropriate constructions.

Anything that is not recognized in the text, the user can still manually enter by typing in values through the graphical user interface.

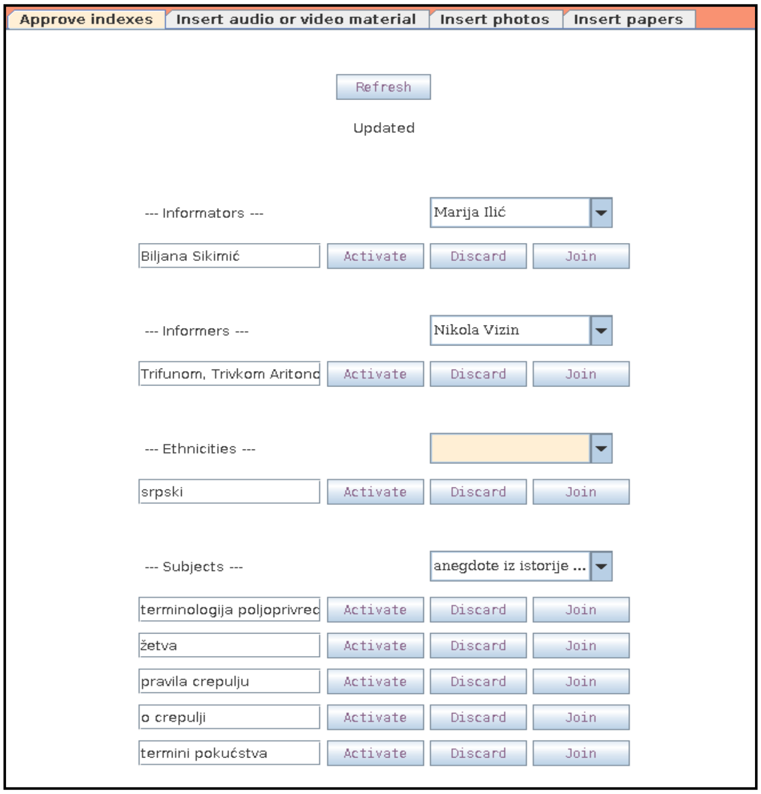
5. IMPLEMENTATION

The application is written in Java programming language and consists of a client and a server part. The server part is responsible for processing protocols with embedded XML tags. The client part is responsible for displaying information to the end user, and for entering new labels and new documents into the database. Complex metadata and topics are forwarded to the panel for review, modification, and acceptance or refusal of the offered value (Figure 3).

Figure 3 shows a panel with values extracted from the processed protocols. These values are automatically entered in the input fields. They can also be joined to existing values. By joining, the old value will be replaced with a new value, while the new value will be assigned an index of the old value. This is also a way to reformulate once inserted value.

Once the value is accepted, it appears in the list of possible values for marking. This is enabled on the panel given in Figure 4.

Panel in Figure 4 consists of four logical parts. First part contains categories of metadata, second part gives overview of selected tags, third part enables inserting of new values for metadata and topics, while fourth part provides search of documents that are already labeled in order to change or modify any of the assigned label, or to assign new labels. In the second part values extracted from protocols are chosen automatically by system presented here. User then can check them, delete, and add other values from those categories as well as from other categories.



Updated

--- Informators --- Marija Ilić

Biljana Sikimić

--- Informers --- Nikola Vizin

Trifunom, Trivkom Aritonc

--- Ethnicities ---

srpski

--- Subjects --- anegdote iz istorije ...

terminologija poljoprivred

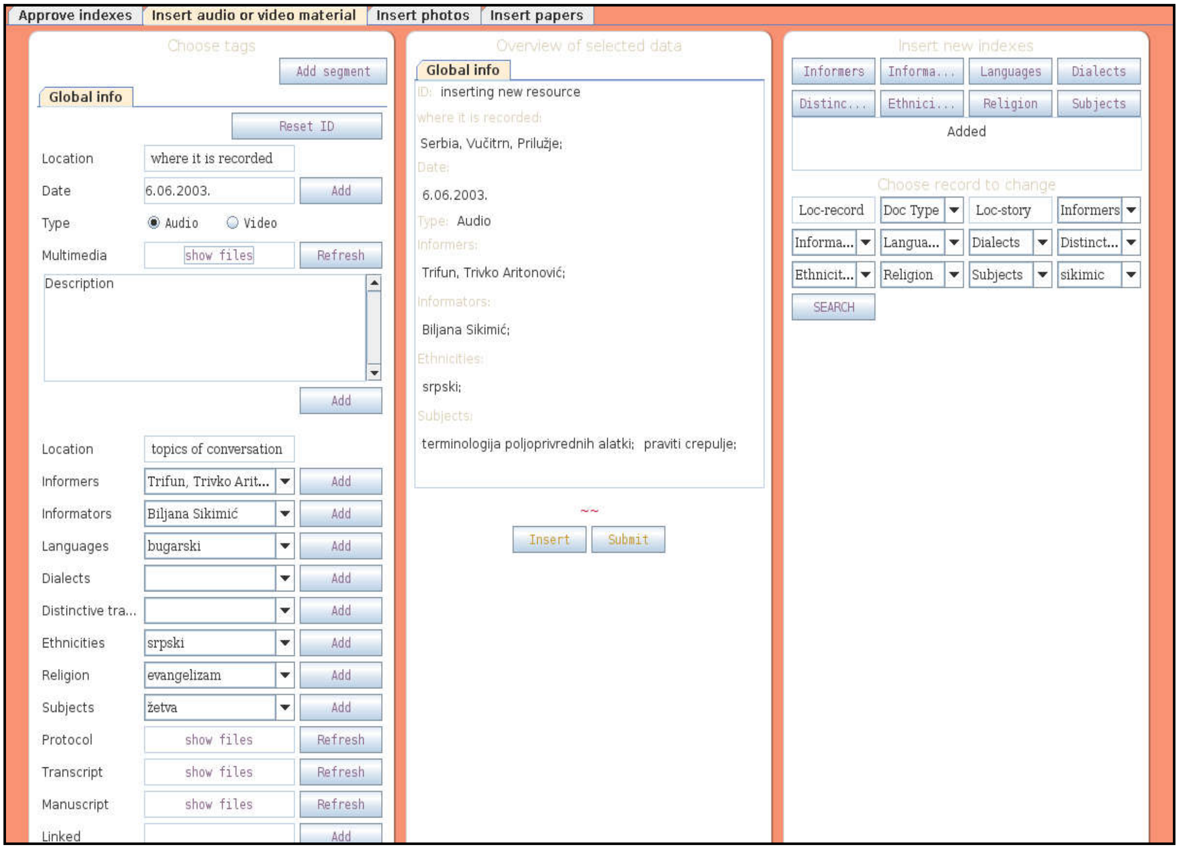
žetva

pravila crepulju

o crepulji

termini pokućstva

Figure 3. Panel for approval of terms



Location	where it is recorded
Date	6.06.2003.
Type	● Audio ○ Video
Multimedia	
Description	
Location	topics of conversation
Informers	Trifun, Trivko Arit...
Informators	Biljana Sikimić
Languages	bugarski
Dialects	
Distinctive tra...	
Ethnicities	srpski
Religion	evangelizam
Subjects	žetva
Protocol	
Transcript	
Manuscript	
Linked	

ID: inserting new resource

where it is recorded:

Serbia, Vučitrn, Priluže;

Date:

6.06.2003.

Type: Audio

Informers:

Trifun, Trivko Aritonović;

Informators:

Biljana Sikimić;

Ethnicities:

srpski;

Subjects:

terminologija poljoprivrednih alatki; praviti crepulje;

~~

Added

Loc-record | Doc Type | Loc-story | Informers
Informa... | Langua... | Dialects | Distinct...
Ethnicit... | Religion | Subjects | sikimic

Figure 4. Panel for annotation of documents

6. CONCLUSION

This prototype illustrates the improvement of the process of tagging documents by automatization of the extraction of values for tags from textual protocols, while the entire process can be considered as semi-automatic because a user has to check the accuracy of the solutions offered.

This application depends on the quality of the protocol and its content, to the extent to which the information required for tagging is well described. The response would be better if the text was structured, but that makes the task for finding relevant data more challenging. Having in mind the problem whose solution was presented here, everything that has been marked by XML annotation system is forwarded to the application as a suggestion of values by which document could be tagged. As long as the accuracy measured is based on mapping protocols with embedded XML tags to the values being sent to the application, the proposed solution is accurate. There is no previously proposed solution for this problem so this prototype gives an idea how to use management system and automatically annotated text protocols to provide solution for semi-automatic tagging of multimedia documents. Further improvement of the methods for automatic extraction of data by natural language processing techniques would lead to lower needs for a user's engagement in reviewing the proposed tags which is known as time consuming and is much more expensive.

REFERENCES

[1] Tanasijević, I., Sikimić, B., Pavlović-Lažetić, G., "Multimedia database of the cultural heritage of the Balkans", ELRA, Proceedings of the Eighth International Conference on Language Resources and Evaluation, {LREC} 2012, Istanbul, Turkey, May 23-25, 2012, pp 2874-2881

[2] Tanasijević, I., Pavlović-Lažetić, G., "Content-based Information Retrieval of an Intangible Cultural Heritage Multimedia Database", 35th Anniversary of Computational Linguistics in Serbia, Belgrade, 2014, pp 87-98

[3] Theodosiou, Z., Georgiou, O., Tsapatsoulis, N., Kounoudes, A., Milis, M., "Annotation of cultural heritage documents based on XML dictionaries and data clustering", Springer, Berlin, Heidelberg, In Euro-Mediterranean Conference, 2010, pp. 306-317

[4] Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C., "Automatic keyphrase extraction and ontology mining for content□based tag recommendation", International Journal of Intelligent Systems, 25(12), 2010, pp 1158-1186.

[5] Medelyan, O., Frank, E., Witten, I. H., "Human-competitive tagging using automatic keyphrase extraction", ACL, In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, 2009, pp. 1318-1327

[6] Jackson, P, Isabelle, M.. "*Natural language processing for online applications: Text retrieval, extraction and categorization"* , Vol.ume 5. John Benjamins Publishing, 2007.

[7] Stanković, R., Krstev, C., Obradović, I., Kitanović, O., "Improving document retrieval in large domain specific textual databases using lexical resources." Transactions on Computational Collective Intelligence XXVI. Springer, Cham, 2017. pp 162-185.

[8] Blagojevic, V., et al, "A Systematic Approach to Generation of New Ideas for PhD Research in Computing", Advances in Computers, Elsevier, Vol. 104, 2016, pp. 1-19.

[9] Jovanović, S., "Građa za tezaurus iz oblasti etnologije", December 2003.

[10] Plotnikova, A., "An addition to the 'folk calendar' section of the ethnolinguistic questionnaire of the Small Dialect Atlas of the Balkan Languages", Proceedings of the Second Workshop. St. Petersburg, December 19, 1997, Russian Academy of Sciences, Institute of Linguistic Studies, Department of Comparative Indo-European and Areal Studies; St. Petersburg, pp 137-139

[11] Vitas, D. Krstev, C., Obradovic, I., Popovic, L., Pavlovic-Lazetic, G., "An overview of resources and basic tools for processing of Serbian written texts." Proceedings of the Workshop on Balkan Language Resources, 1st Balkan Conference in Informatics, 2003

Formalizing Algebrization of Geometry Statements

Simić, Danijela

Abstract: ***For several decades, algebraic methods have been successfully*** **used in automated deduction in geometry. Objects in Euclidean geometry and relations between them are expressed as polynomials, and algebraic methods (e.g., Gröbner bases) are used over such a set of polynomials. We describe a formalization of an algorithm in Isabelle/HOL that accepts a term representation of a geometry construction and returns a corresponding set of polynomials. Our further work involves using the method of Gröbner bases within Isabelle system on the generated polynomials, in order to implement a fully formally verified algebraic prover for geometry.**

Index Terms: algebrization of geometry statements, automated proving in geometry, proof assistants.

1. INTRODUCTION

1.1. Automated Reasoning in Geometry

The greatest advance in the automatic proving in geometry was made by Wu. He limited the problem set so that it does not contain inequalities. He was able to apply a powerful method to prove complicated theorems [1]. The method became more popular as many theorems were successfully proved (among which are the Feuerbach theorem, Carnot theorem, the tangent - secant theorem, Ptolemy's theorem, Euler's theorem, Stuart's theorem, and many others) [15].

Numerous authors have implemented and improved on this algorithm by various heuristics [12, 6]. However, it soon become clear that Wu's approach could be derived from Ritt's work [7], so this method is often called the Wu - Ritt method. The success of this method has influenced the development of new methods. One of the successful ones is the Gröbner basis method, which is based on Buchberger's algorithm [17] and can be applied to the same class of problems as the Wu method. Buchberger's algorithm has been improved by numerous authors and today there are various heuristics that serve to increase the efficiency of the algorithm. By using this method many geometric claims have been demonstrated [17], such as Gauss theorem, Pappus' theorem, Desargues's theorem, the theorem of Euler's real triangle and many others. There are numerous implementations, and some of them are in the commercial programs (e.g. Matlab and Mathematica).

The main disadvantage of these two methods is that they cannot be used to prove statements with inequalities, and therefore the theorems about point order cannot be considered. To solve problems with inequalities, Wu suggested a method based on finding minimum or maximum values of the polynomial function under certain conditions [3]. In addition to proving in elementary geometry, Wu introduced the method for proving in differential geometry [4]. There are also extensions that make it possible to use the method for hyperbolic geometry [2].

All listed methods translate geometry statements into equations using coordinates of the points of the geometry objects that are observed and then apply algebraic techniques on these equations. These provers give the answer "yes" or "no", but do not provide any information about the justification that would be understandable to a human and similar to evidence seen in school textbooks. There are numerous attempts to make provers that would produce legible evidence. One of the most important is the area method [14].

1.2. Formal Theorem Proving

Formal mechanized theorem proving assumes formalizing mathematical statements within proof assistants — specialized software tools used to find and check proofs semi-automatically, guided by user interaction. Formal theorem proving is used both in classical mathematics and in hardware and software production. Using mechanical theorem provers significantly increases the confidence in mathematical results, because many mathematical problems are usually so complex that there is no strong confidence that pen-and-paper reasoning about them is sound.

Manuscript received November 1, 2018. This work was supported in part by the Serbian Ministry of Science grant 174021.

Danijela Simić is with the Department for Computer Science, Faculty of Mathematics, University of Belgrade (e-mail: danijela@matf.bg.ac.rs).

The most important results achieved to date are the formalization of the theorem about free numbers [18], the formal proof of the four-color theorem [11], Brauer's theorem of a fixed point [9], the basic theorem of algebra [20, 21], Gedel's incompleteness theorem [5], and many theorems of real analysis [10, 13].

One of the leading proof assistants used for interactive theorem proving is Isabelle/HOL [22]. Isabelle is a generic system for implementing logical formalisms, and Isabelle/HOL is the specialization of Isabelle for Higher-Order Logic (HOL). It could be said that Isabelle/HOL merges Functional Programming and Logic. Isabelle/Isar is a high-level language for writing proofs in a declarative manner. Working with Isabelle assumes creating theories. Roughly speaking, a theory is a named collection of types, objects, functions, and theorems.

1.3. Motivation and main results

The central motivation of this work is to connect automated and formal proving in geometry and to construct a formally verified automated prover for geometry. Before applying algebraic methods, geometry constructions and statements are transformed into a set of polynomial equations, but there is no unique, nor verified algorithm for this. Usually transformation is done by ad-hoc methods and there is no formal link between obtained polynomials and given geometric objects. With a formally verified translation method this problem would be resolved and this work is a step in that direction. So, our work is original and brings new insight on the subject as advised in [19].

The best solution to this problem is described in paper by Narboux at al. [8] but though they deal with algebraic prover very efficiently using certificates, they do not check weather transforming geometry statements into algebraic form is correct. Rather than that, they just input geometry statements already written in algebraic form with checking if the process of obtaining them is correct.

2. ALGEBRAIC ALGORITHMS

Once the geometric theorem has been algebrized, algebraic theorem proving methods themselves can be applied. Algebraic theorem provers use specific algorithms over polynomial systems (each polynomial equation of the form $p_1(v_1, \dots, v_n) = p_2(v_1, \dots, v_n)$ is transformed to $p_1(v_1, \dots, v_n)\ p_2(v_1, \dots, v_n) = 0$, i.e., to the form $p(v_1, \dots, v_n) = 0$). If $f_1, \dots, f_k$ are polynomials obtained from the construction, and $g_1, \dots, g_l$ are polynomials obtained from the statement, then the conjecture is reduced to checking if for each g_i it holds that

$$\forall\ v_1, \dots\dots, v_n \in \mathrm{R} \bigwedge_{i=1}^{k} f_i(v_1, \dots\dots, v_n) = 0 \Rightarrow g_i(v_1, \dots\dots, v_n) = 0$$

As Tarski noted, this could be decided by a quantifier elimination procedure for the reals. In practice, it is hard to prove non-trivial geometric properties in this fashion, because even sophisticated algorithms for real quantifier elimination are relatively inefficient. Therefore, another approach is taken. The main insight, given by Wu in 1978 is that remarkably many geometrical theorems, when formulated as universal algebraic statements in terms of coordinates, are also true for all complex values of the coordinates. So, instead of checking polynomials over reals, the field of complex numbers is used and the following conjecture is considered:

$$\forall\ v_1, \dots\dots, v_n \in \mathrm{C}\ f_i(v_1, \dots\dots, v_n) = 0 \Rightarrow g_i(v_1, \dots\dots, v_n) = 0$$

This is true when g belongs to the radical of the ideal $I = (f_1, \dots, f_k)$, generated by the polynomials f_i, that is when there exists an integer r and polynomials $h_1, \dots, h_k$ such that

$$g_i^r = \sum_{i=1}^{k} h_i f_i$$

The two most famous methods use a kind of Euclidean division to check the validity of a conjecture of the form 1. Buchberger's method consists in transforming the generating set into a Gröbner basis, in which a division algorithm can efficiently used, while in the Wu's method a pseudo-division is used which closely mimics Euclidean division.

The main operation over polynomials in Wu's method is pseudo division which, when applied to two polynomials $p(v_1, \dots, v_n)$ and $q(v_1, \dots, v_n)$ produces the decomposition

$$c^r p = tq + r,$$

where $c(v_1, \dots, v_{n-1})$ is the leading coefficient of q in the variable v_n, r is the number of non-zero coefficients of p, $t(v_1, \dots, v_n)$ is the pseudo-quotient, $r(v_1, \dots, v_n)$ is the pseudo-remainder, and the degree v_n in r is smaller than in q. Since $r = c^r p - tq$, it is clear that r belongs to the ideal generated by p and q.

The first step of Wu's method uses the pseudo-division operation to transform the construction polynomial system to triangular form, i.e., to a system of equations where each successive equation introduces exactly one dependent variable. After that, the final reminder is calculated by pseudo dividing polynomial for statement (g_i) by each polynomial from triangular system.

Summarizing, Wu's method, in its simplest form, allows to compute some polynomials *c*, h_1, . . . , h_k and *r* such that

$$cg_i = \sum_{i=1}^{k} h_i f_i + r$$

If the final remainder *r* is equal to zero, then the conjecture is considered to be proved. This simple method of Wu is not complete (in algebraic sense). A more complex and complete version of the method uses ascending chains which are considered in the Ritt-Wu principle.

2.1. Applying Gröbner Basis Method for Proving a Geometry Theorem — an Example

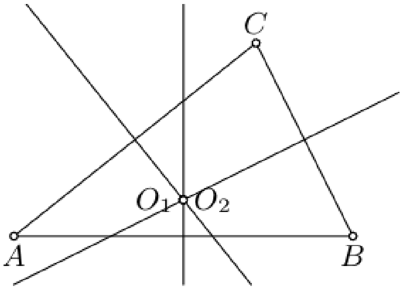


Figure 1Intersection of bisector

The next simple example we illustrate how a simple geometric statement can be reduced to the ideal membership problem and solved by using the Gröbner basis method. Bisectors of the sides of a triangle are intersecting in one point. The basic idea is to place the figure above in the coordinate plane and then to interpret the hypotheses of the theorem as statements in coordinate, rather than Euclidean, geometry. So we begin by coordinatizing the parallelogram by placing the point A at the origin, so A = (0, 0). Now we can say that the point B corresponds to (c, 0), and that C corresponds to (a, b) (it is obvious that the theorem should be proved for any set of coordinates, but it can be easily shown that we can translate any coordinates into these ones). Now, this geometric construction is going to be translated into set of polynomials.

Assume that bisectors of sides AB and BC are intersecting in point $O_1 = (x_1, y_1)$. Bisectors are completely determined by points A, B, C and O_1 and this yields the following equations:

$$P1 : x_1 - \frac{c}{2} = 0$$

$$P2 : \frac{c-a}{b} \cdot x_1 - y_1 + \frac{b^2-c^2+a^2}{2 \cdot b} = 0$$

Assume that bisectors of sides AB and (x_2, y_2). This yields another two equations:

$$p_1{}' : x_1 - \frac{c}{2} = 0$$

$$P3 : \frac{a}{b} \cdot x_2 + y_2 - \frac{b}{2} - \frac{a^2}{2 \cdot b} = 0$$

Thus, there is a set:

$$G = \{p_1, p_2, p_1', p_3\}$$

These polynomials describe the construction. Now, it should be proved that $O_1 = O_2$. i.e. $(x_1, y_1) = (x_2, y_2)$. In order to apply the Gröbner basis method, Görbner basis G' of the set G is going to be calculated and the aim is to prove $x_1 - x_2 \rightarrow_G{}' = 0$ and $y_1 - y_2 \rightarrow_G{}' = 0$. With the polynomials on the left hand side of these equations the statement is described.

In order to compute Gröbner basis of the set G Buchberger's algorithm is used and finally calculated result is:

$$G' = \{g_1, g_2, g_3, g_4, g_5, g_6\} = \{ x_1 - \frac{c}{2}, x_2 - \frac{c}{2}, \frac{c-a}{b} \cdot x_1 - y_1 + \frac{b^2-c^2+a^2}{2 \cdot b}, \frac{a}{b} \cdot x_2 + y_2 - \frac{b}{2} - \frac{a^2}{2 \cdot b}, - y_1 + \frac{b}{2 \cdot} + \frac{a^2}{2 \cdot b} - \frac{a \cdot c}{2 \cdot b}, y_2 - \frac{b}{2 \cdot} - \frac{a^2}{2 \cdot b} + \frac{a \cdot c}{2 \cdot b} \}$$

Using this $x_1 - x_2 \rightarrow_G{}'$ 0 can be easily shown. Indeed,

$$x_1 - x_2 \rightarrow_{g1} -x_2 + \frac{c}{2} \rightarrow_{g3} - x_2 + \frac{c}{2} + x_2 - \frac{c}{2} = 0 .$$

Similarly $y_1 - y_2 \rightarrow_G{}'$ 0 can be shown

$$y_1 - y_2 \rightarrow_{g5} -y_2 + \frac{b}{2 \cdot} + \frac{a^2}{2 \cdot b} - \frac{a \cdot c}{2 \cdot b} \rightarrow_{g6} -y_2 + \frac{b}{2 \cdot} + \frac{a^2}{2 \cdot b} - \frac{a \cdot c}{2 \cdot b} + y_2 - \frac{b}{2 \cdot} - \frac{a^2}{2 \cdot b} + \frac{a \cdot c}{2 \cdot b} = 0.$$

2.2. Algebraization of Geometry Statements

Algebraic methods, used as methods for automated theorem proving in geometry for theorems of constructive type (i.e., conjectures about geometric objects obtained by geometric constructions), introduce (symbolic) coordinates for geometric objects involved (points, and possibly lines), express geometric constructions and statements as algebraic (multivariate polynomial) equations involving introduced co-ordinates and then use algebraic means to prove that the statement follows from the construction.

The standard algebrization procedure introduces fresh symbolic variables for point coordinates and introduces (polynomial) equations that characterize every construction step and the statement to be proved. Although for lines involved in the construction unknown coefficients could be introduced, the standard procedure avoids that and uses only points (while lines are specified only implicitly). Each construction starts from a set of free points and introduces dependent points along the way. In some cases, dependent points are chosen with a degree of freedom (e.g., choosing a random point on line). Each point gets a pair of coordinates represented by symbolic variables. Free variables are usually denoted by ui, while the dependent ones are denoted by xi. If a point is free, both its coordinates will be free variables. If a point is, dependent, but with a degree of freedom, one coordinate will be a free, while the another one will be a dependent variable. If a point is dependent both its coordinates will be dependent ones.

Geometry constraints over points can be formulated as algebraic constraints over the point coordinates (i.e., as polynomial equations over the introduced symbolic variables). For example, assume that symbolic coordinates of the point *A* are (x_a, y_a), the point *B* are (x_b, y_b), and the point *C* are (x_c, y_c). The fact that *A* is the midpoint of the segment *BC* corresponds to an algebraic condition 2 $x_a = x_b + x_c$ and 2 $y_a = y_b + y_c$. The fact that *A*, *B*, and *C* are collinear corresponds to the algebraic condition $(x_a \ x_b) \ (y_b \ y_c) = (y_a \ y_b) \ (x_b \ x_c)$. Similar connections are formulated for other basic geometry relationships (parallel lines, perpendicular lines, segment bisectors, etc.).

Example 1 Let ABC be a triangle, and let B_1 be the midpoint of the edge AC and C_1 be the midpoint of the edge AB. Then, the midsegment B_1C_1 is parallel to BC.

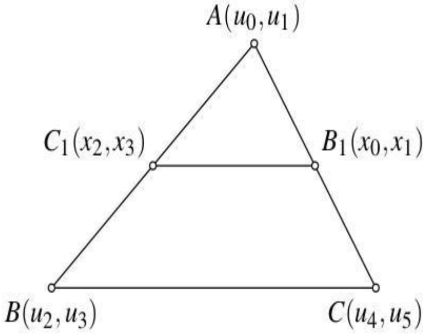


Figure 2. Parallel midline

In this example, A, B, and C are free points so they are introduced symbolic variables A(u_0, u_1), B(u_2, u_3), and C(u_4, u_5). Points B_1 and C_1 are dependent so they are introduced symbolic variables B_1(x_0, x_1) and C_1(x_2, x_3). Since B_1 is the midpoint of AC, it holds that 2 $x_0 = u_0 + u_4$ *and* 2 $x_1 = u_1 + u_5$. *Since C_1 is the midpoint of AB, it holds that* 2 $x_2 = u_0 + u_2$ *and* 2 $x_3 = u_1 + u_3$. *These four equations come from the description of the construction, i.e., from the premises of the conjecture. In order to show that B_1C_1 is parallel to BC, it suffices to show that* $(x_2 \ x_0) \ (u_5 \ u_3) = (x_3 \ x_1) \ (u_4 \ u_2)$ *holds. This equation corresponds to the conclusion of the conjecture. So, the geometric problem is reduced to showing that every n-tuple satisfying the first four equations (stemming from the construction) also satisfies the last equation (stemming from the conclusion), i.e., to show that*

$\forall u_0\, u_1\, u_2\, u_3\, u_4\, u_5\, x_0\, x_1\, x_2\, x_3 \in$ R. $2 \cdot x_0 = u_0 + u_4 \wedge 2 \cdot x_1 = u_1 + u_5 \wedge 2 \cdot x_2 = u_0 + u_2 \wedge 2 \cdot x_3 = u_1 + u_3 {=}{\Longrightarrow} (x_2 - x_0) \cdot (u_5 - u_3) = (x_3 - x_1) \cdot (u_4 - u_2).$

Note that in the above example, the condition that *ABC* is a triangle is not translated into conditions that *A*, *B*, *C* are pairwise different. Also, the condition that B_1C_1 is parallel to *BC* is represented by equation $(x_2 \ x_0) \ (u_5 \ u_3) = (x_3 \ x_1) \ (u_4 \ u_2)$. However, this algebraic equation is actually equivalent to the following weaker condition: *B C* or B_1 C_1 or B_1C_1 is parallel to *BC*. Therefore, when translated back in geometry terms, the conjecture that is to be proved by an algebraic method is:

*Let B_1 be the midpoint of the segment AC and C_1 be the midpoint of the segment AB. Then, the segment B_1C_1 is parallel to BC or B is identical to C or B_1 is identical to C_1.*Since, B_1 ƒ≡ C_1 follows from *B* ƒ≡ *C*, the above conjecture is equivalent with

Let B and C are two distinct points, let B_1 be the midpoint of the segment AC and C_1 be the midpoint of the segment AB. Then, the segment B_1C_1 is parallel to BC.

This shows that translating a conjecture from geometry terms to algebraic terms and vice versa involves dealing with important details. A hypothesis of the form *AB CD* is typically represented by equation of the form $(x_b \ x_a) \ (y_d \ y_c) = (x_d \ x_c) \ (y_b \ y_a)$. Moreover, in most systems, this equation is used as a definition for *AB CD* which, unfortunately, breaks the link with synthetic geometry.

It can be shown that most geometry properties are invariant under isometric transformations. If P_1 and P_2 are two free points, there always exists an isometry (a composition of a translation and a rotation) that maps P_1 to the point (0, 0) (i.e., the origin of the Cartesian plane) and P_2 to a point on the *x*-axis. Therefore, without loss of generality, it can be assumed that one free point has coordinates (0, 0), while another one has coordinates (u_0, 0) (or (0, u_0)). This assumption can significantly reduce the amount of work needed by the algebraic methods. In addition, there are heuristics (aimed at improvement of efficiency) for choosing among the free points which two will get these distinguished coordinates.

Example 2 *Without loss of generality in the conjecture from Example 1, the points B and C can be assigned coordinates* $B(0, 0)$*, and* $C(u_4, 0)$*. Therefore, the algebraic conjecture to be proved is as follows:*

$\forall u_0\ u_1\ u_4\ x_0\ x_1\ x_2\ x_3 \in R.2 \cdot x_0 = u_0 + u_4 \wedge 2 \cdot x_1 = u_1 \wedge 2 \cdot x_2 = u_0 \wedge 2 \cdot x_3 = u_1 \Longrightarrow (x_2 - x_0) \cdot 0 = (x_3 - x_1) \cdot u_4.$

which is trivially valid (since $x_3 - x_1 = 0$ *follows from* $2 \cdot x_1 = u_1$ *and* $2 \cdot x_3 = u_1$*).*

3. Formalization in Isabelle/HOL

Being able to show geometry statements using polynomials and Gröbner basis gives the opportunity to automate proving geometric statements. However, the connection between geometry and algebra is usually not formally given. Our main goal is to do this and prove the correctness of the whole Gröbner basis method. The most important part is the verification of the step that translates geometry constructions into polynomial equations and to the ideal membership problem.

A. Term Representation of Geometry Constructions

First, it is necessary to represent geometry constructions in a convenient way such that it could be easily processed by a computer, i.e. used by our algorithm. Thus, geometry constructions are represented using terms. Currently, two types of objects are supported – points and lines. Also, geometry statements are represented as terms. In Isabelle corresponding data types are defined by:

```
datatype
point_term =
   MkPoint nat (* Independent points determined
by their index *)
| MkIntersection line_term line_term
| MkMidpoint point_term point_term

and line_term =
  MkLine point_term point_term
| MkNormal line_term point_term
| MkParallel line_term point_term
| MkBisector point_term point_term
datatype statement_term =
  EqualP point_term point_term
| EqualL line_term line_term
| Incident point_term line_term
| Midpoint point_term point_term point_term
| Parallel line_term line_term
| Normal line_term line_term
| Colinear point_term point_term point_term
```

As it can be seen, a point can be specified by its identifier, or constructed as an intersection of two lines. Alternatively, it can be constructed as a midpoint between two points. Similarly, a line can be determined by two points or it can be constructed as a bisector of a segment, etc. Also, there are different types of statements. For example, Incident point_term line_term denotes that a point belongs to a line, EqualP point_term point_term denotes that two points are equal etc.

Example 1: *The term Incident (MkPoint 1) (MkLine (MkPoint 1) (MkPoint 2)) represents the statement that a point belongs to a line determined by that point and another one.*

Example 2: *The term*

```
let c = MkBisector (MkPoint 1) (MkPoint 2);
    b = MkBisector (MkPoint 1) (MkPoint 3);
    a = MkBisector (MkPoint 2) (MkPoint 3);
    O1 = MkIntersection a b;
    O2 = MkIntersection a c in
    EqualP O1 O2
```

is the one that represents the example from the previous section — two points given from a line intersection are equal. Having defined term representation of geometry constructions, the next step is to translate a term into a set of polynomials so that Gröbner basis method can be applied.

B. Brief Description of the Translation Algorithm

This algorithm is used to translate term representation of geometry constructions and statements into their polynomial counterparts. The algorithm is recursive and produces two sets. The first set is the set of polynomials representing geometry construction and is thus called *the construction-set*. The second one is the set of polynomials representing the statement and we it will be called the *statement-set*. Gröbner basis method relies on showing that each polynomial in the *statement-set* can be reduced to zero by using the Gröbner basis calculated for the construction-set.

Algorithm recursively process terms and for all unknown objects new coordinates are added. Also, at the same time, new polynomials are added to the corresponding sets regarding identities in analytic geometry.

As an example, let us show the translation step for the statement of the form Incident point_t line_t, where point_t and line_t can be arbitrarily complex terms for a point and a line. The algorithm for this particular example works like this:

- add new variables x_0 and y_0. These variables are unknown coordinates for the point *O* given by the term point_t — $O(x_0 , y_0)$
- add variables a_0, b_0, and c_0 representing coefficients for the line *p* given by the term line_t — $p = a_0 \cdot x + b_0 \cdot y + c_0$.
- call function *point_poly(point_t,* x_0 , y_0*)* that constructs polynomials connecting the variables x_0 and y_0 with the term point_t.
- call function *line_poly(line_t,* a_0 , b_0 , c_0*)* that constructs polynomials connecting the variables a_0, b_0 and c_0 with the term line_t.
- add the polynomial $a_0 \cdot x_0 + b_0 \cdot y_0 + c_0$ in statement-set

Having in mind that the idea of this work is to formally prove correctness of the method, it should be clear that everything used in the description of the method must be formally proved and such is the case with polynomials. This means that it is desirable to have such a formally proven theory in order to be able to represent polynomials and perform different calculations on them. Since this is not the focus of this work and has already been developed before, the theory used here is the Theory of Executable Multivariate Polynomials. In this work the authors represent polynomials using lists and formally prove many of their properties.

Just as an illustration, we show a fragment of Isabelle code implementing this translation step.

```
algebrize (Incident p l) ==
let x = point_id_x 0;
     y = point_id_y 0;
     a = line_id_a 0;
     b = line_id_b 0;
     c = line_id_c 0;
    (s', pp) = point_poly p x y (| maxp = 0,
                                   maxl = 0 |);
    (_, lp) = line_poly l a b s'
in
(sup pp lp,
Fset.Set[poly_of (PSum [PMult[PVar a,
                              PVar x],
                        PMult[PVar b,
                        PVar y], PVar c])])"
```

As it may be noticed there are two new functions *point_poly* and *line_poly* that have two arguments - term and two variables. These functions are mutually recursive and used to calculate the construction-set. That will be demonstrated by this example: MkIntersect line1_t line2_t. Same as before, line1_t and line2_t are terms representing lines and this entire term is representing a point. What should be calculated here are the polynomials describing the point. The polynomials depend on terms and variables. In this example the method will work like this:

- add variables a_1, b_1 and c_1 and these are unknown coefficients for the line *p* given by the term line1_t — $p = a_1 \cdot x + b_1 \cdot y + c_1$.
- add variables a_2, b_2 and c_2 and these are unknown coefficients for the line *q* given by the term line2_t — $q = a_2 \cdot x + b_2 \cdot y + c_2$.
- call function line_poly(line1_t, a_1, b_1, c_1)
- call function line_poly(line2_t, a_2, b_2, c_2)
- add polynomials $x \cdot (a_2 \cdot b_1 - a_1 \cdot b_2) + b_1 \cdot c_2 - c_1 \cdot b_2 = 0$ and $y \cdot (a_2 \cdot b_1 - a_1 \cdot b_2) + a_2 \cdot c_1 - a_1 \cdot c_2 = 0$ to the construction-set

These polynomials are derived using geometry identities so that a given geometry property holds.

C. Proving Correctness

The main part of our work is to prove the correctness of our translation by showing the connection between the geometric statement and the obtained sets of polynomial equations. To do so, analytic geometry will be used as a connection between the synthetic geometry and algebra. It should be shown that everything that is proved using the algebraic method also holds in models of the synthetic geometry. On the one hand, it could be shown that analytic geometry is a model of synthetic geometry, and furthermore it could be shown that all models are isomorphic, so everything that holds in one model holds in all others. On the other hand, it should be shown that everything that is proved using the algebraic method is also correct in analytic geometry.

Connection between synthetic and analytic geometry. Let us consider the first problem — show that analytic geometry is a model of synthetic geometry. Basic objects in analytic geometry have to be formally defined. For example, a point
can be defined as a pair of two real numbers:

```
type_synonym point = "real * real"
```

Now, defining the line is slightly more complex. Lines can be identified by triples of their coefficients. However, first two components cannot be zero simultaneously. Additionally, lines can have different coefficients and still be the same (if coefficients are proportional). For example, $x + 2 \cdot y + 1 = 0$ and $2 \cdot x + 4 \cdot y + 2 = 0$ determine the same line. Two lines $a_1 \cdot x + b_1 \cdot y + c_1 = 0$ and $a_2 \cdot +b_2 \cdot y + c_2 = 0$ are equal if and only if $\exists k. a_1 = k \cdot a_2$, $b_1 = k \cdot b_2$, $c_1 = k \cdot c_2$. In order to represent a line we define the equivalence relation between triples *(a_1, b_1, c_1)* and (a_2, b_2, c_2) such that *$\exists k . a_1 = k \cdot a_2, b_1 = k \cdot b_2, c_1 = k \cdot c_2$*. A line can be defined as an equivalence class over the set *$\{(a, b, c) \mid a, b, c \in R, a \neq 0 \vee b \neq 0\}$*. Isabelle code formalizing this definition is:

```
typedef line_coeffs = "{(A::real, B::real, C::real).
                                    A ≠ 0 | B ≠ 0}"
by auto
definition
line_coeffs_eq :: "line_coeffs => line_coeffs =>
bool" where: "line_coeffs_eq c c1 =
 (EX A B C A1 B1 C1.
     (Rep_line_coeffs c = (A, B, C) &
      Rep_line_coeffs c1 = (A1, B1, C1) &
      (EX k. k ~= 0 & A1 = k*A & B1 = k*B & C1 =
k*C)))"

lemma line_coeffs_eq_equivp:
                                    "equivp
line_coeffs_eq"
(* prove that line_coeffs_eq is an equivalence
relation *)

quotient_type line = line_coeffs /
"line_coeffs_eq"
by (rule line_coeffs_eq_equivp)
```

Based on these definitions, additional geometric primitives (e.g., incidency) can be defined and their geometric properties (e.g., Hilbert’s axioms) can be proved.

```
lemma
  assumes "incident P1 l1" "incident P2 l1"
                "incident P1 l2" "incident P2 l2" "P1 ≠
P2"
  shows "l1 = l2"
```

Connection between analytic geometry and algebra. The other thing we need to do is to show that our method is correct, i.e. if we prove something using Gröbner basis method, it really holds in analytic geometry. To do so we need to show this:

$$(\forall(u,x))(\forall g \in G)((\forall f \in F. f(u,x) = 0) \Rightarrow g(u,x) = 0) \Rightarrow \quad \text{geometric statment}$$

where *$F(u,x)$* is a construction-set and *$G(u,x)$* is a statement-set. The first part is to show that $\forall f \in F$ and $\forall (u,x)\ f(u,x) = 0$. The second part is to show that if *$(\forall g \in G)(\forall f \in F. f(u,x) = 0) \Rightarrow g(u,x) = 0)$* holds then the geometric statement in analytic geometry holds. This statement is inductively proved in Isabelle/HOL:

```
theorem “let (cp, sp) = algebrize term in
        (ALL ass. ((ALL p : cp. eval_poly ass p = 0) →
                  (ALL p : sp. eval_poly ass p = 0)) →
AnalyticGeometry.valid s)“
```

While proving this statement all objects used in algebraic proof gain coordinates and then the proof can be connected with analytic geometry. Consider the following example:

```
Incident (MkMidpoint (MkPoint 0) (MkPoint 1))
          (MkLine (MkPoint 0) (MkPoint 1))
```

(MkPoint 0) (MkPoint 1) are assigned coordinates *(p^x_0, p^y_0)* and *(p^x_1, p^y_1)* that are fixed and do not depend on other objects. Then (MkMidpoint (MkPoint 0) (MkPoint 1)) are assigned coordinates *(x_1, y_1)* and these depend on coordinates of (MkPoint 0) and (MkPoint 1). The same thing with (MkLine (MkPoint 0) (MkPoint 1)) where are added coordinates *(a_1, b_1, c_1)* for the line. The following equations should be satisfied:

$$2 \cdot x_1 = p^x_1 + p^x_1$$
$$2 \cdot y_1 = p^y_0 + p^y_1$$
$$a_1 \cdot (p^x_1 \cdot p^y_0 - p^y_1 \cdot p^x_0) - c_1 \cdot (p^y_1 - p^y_0) = 0$$
$$b_1 \cdot (p^x_1 \cdot p^y_0 - p^y_1 \cdot p^x_0) + c_1 \cdot (p^x_1 - p^x_0) = 0$$

Using identities in analytic geometry this is easily shown.

4. CONCLUSION AND FUTURE WORK

In this paper we present a formalization of algebrization of geometry statements. Each statement is given using term representation. It is then transformed into a set of polynomials using our algorithm for transformation and finally it is proved that this process is correct (the statements stay the same after being transformed into polynomial form).

This part of the algorithm is never formally verified before. It is written many times, but never formally verified. And since this algorithm is used to prove something it is important to be able to trust that it is correct. So, formal analysis is done and that its correctness is proved.

Since the work presented here is in its early stages there are many things that should be improved. More geometric objects (circles, ellipses etc.) should be included and more types of geometry statements should be added.

Further, the translation method should be connected to trusted implementation of Gröbner basis construction (already available in Isabelle/HOL). In this way, we would have a fully formally verified automated prover for geometry.

Since our translation does not depend on the algebraic method used for the generated sets of polynomials, a significant part of our formalization can be reused for making a fully verified implementation of other algebraic methods — most notably, Wu's method.

If we show that a statement holds in a Cartesian plane (e.g. in analytic geometry), does it follow that the statement can be proved from the axioms of Hilbert or Tarski? There is a conjecture that all models of the Hilbert axiom are mutually isomorphic. There is a similar statement for the models of the axiom of Tarski. In addition, it has been shown that the axiomatic systems of Tarski and Hilbert are deductively complete, which means that each statement of the any model could be proved from the axioms. Using this, we know that if the algebraic prover shows connection between polynomials, then this statement is valid in the Cartesian plane (analytic geometry), and hence in all other models of geometry, which implies that the conjecture can be proved from, for example, Hilbert axioms. The formalization of these meta-theoretical properties of geometry is a very demanding task and it would be necessary to formalize the notion of provability within the geometry of Hilbert or within geometry Tarski. However, this paper takes certain steps in that direction.

References

[1] W.-t. Wu, "On the decision problem and the mechanization of theorem-proving in elementary geometry," *Scientia Sinica,* vol. 21, pp. 159-172, 1978.

[2] L. Yang, X. Gao, S. Chou and Z. Zhang, "Automated proving and discovering of theorems in non-euclidean geometries," *Proceedings of Automated Deduction in Geometry (ADG98), Lecture Notes in Artificial Intelligence,* vol. 1360, pp. 171-188, 1998.

[3] W. Wen-Tsun, "On a finiteness theorem about optimization problems," 1992.

[4] W. Wen-Tsun, "Mechanical theorem proving of differential geometries and some of its applications in mechanics," *Journal of Automated Reasoning,* vol. 7, pp. 171-191, 1991.

[5] N. Shankar, Metamathematics, machines and Gödel's proof, Cambridge University Press, 1997.

[6] S. E. N. WANG DONG-MING HU, "A mechanical proving system for constructible theorems in elementary geometry," 1987.

[7] J. F. Ritt, Differential algebra, vol. 33, American Mathematical Soc., 1950.

[8] Narboux, Julien. "A decision procedure for geometry in Coq." *International Conference on Theorem Proving in Higher Order Logics*. Springer, Berlin, Heidelberg, 2004.

[9] J. Harrison, "The HOL Light theory of Euclidean space," *Journal of Automated Reasoning,* vol. 50, pp. 173-190, 2013.

[10] J. Harrison, Theorem proving with the real numbers, Springer Science & Business Media, 2012.

[11] G. Gonthier, "Formal proof--the four-color theorem," *Notices of the AMS,* vol. 55, pp. 1382-1393, 2008.

[12] X. Gao, "Transcendental functions and mechanical theorem proving in elementary geometries," *Journal of Automated Reasoning,* vol. 6, pp. 403-417, 1990.

[13] L. Cruz-Filipe, "A constructive formalization of the fundamental theorem of calculus," in *Types for Proofs and Programs*, Springer, 2002, pp. 108-126.

[14] S.-C. Chou, X.-S. Gao and J.-Z. Zhang, "Automated production of traditional proofs for constructive geometry theorems," in *Logic in Computer Science, 1993. LICS'93., Proceedings of Eighth Annual IEEE Symposium on*, 1993.

[15] S.-C. Chou, "Proving elementary geometry theorems using Wu's algorithm," 1984.

[16] B. Buchberger, "Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal," *Journal of symbolic computation,* vol. 41, pp. 475-511, 2006.

[17] B. Buchberger and F. Winkler, Gröbner bases and applications, vol. 251, Cambridge University Press, 1998.

[18] J. Avigad, K. Donnelly, D. Gray and P. Raff, "A formally verified proof of the prime number theorem," *ACM Transactions on Computational Logic (TOCL),* vol. 9, p. 2, 2007.

[19] V. e. a. Blagojevic, "A Systematic Approach to Generation of New Ideas," *Advances in computers,* vol. 104, pp. 1--31, 2017.

[20] R. Milewski, "Fundamental theorem of algebra1," 2001.

[21] T. Nipkow, L. C. Paulson and M. Wenzel, "Isabelle/HOL: a proof assistant for higher-order logic," vol. 2283, 2002.

[22] H. Geuvers, F. Wiedijk and J. Zwanenburg, "A constructive proof of the Fundamental theorem of algebra without using the rationals," *Types for Proofs and Programs,* p. 96–111, 2000.

Use of Dataflow Computing for Bitcoin Mining

Meden, Rok and Kos, Anton

Abstract: ***Bitcoin mining is a heavy energy consuming computational process of updating the public ledger of Bitcoin transactions (blockchain) in the decentralized peer-to-peer Bitcoin network. The bitcoin mining algorithm (also known as PoW, proof-of-work), had been implemented in many programming languages for various computing hardware, starting with CPU (Central Processing Unit) and GPU (Graphics Processing Unit), continuing with FPGA (Field-Programmable Gate Array) and ending with custom ASIC (Application-Specific Integrated Circuit) solutions. In this paper, we implemented the bitcoin mining algorithm on two FPGA based Maxeler dataflow computers, MAX2B and MAX5C, and compared their performance to CPU and GPU based solutions in three categories: computation speed (hash rate), electric power and energy efficiency. Thanks to the exploit of massive parallelism afforded by both the bitcoin mining algorithm and Maxeler dataflow computers, we achieved up to 102 times faster and up to 256 times more energy-efficient bitcoin mining compared to general multi-core CPU solutions. However, Maxeler dataflow computers are quite inferior to ASIC solutions which were first developed in year 2013; those proved to be the fastest and most energy-efficient for mining bitcoins and have thus been widely used for the task ever since.***

Index Terms: ***bitcoin, cryptocurrency, mining, dataflow computing, Maxeler, MAX2B, MAX5C***

1. Introduction

The first digital currency projects date back to early 1990s. The early digital currencies were usually backed by a national currency or precious metal, such as gold and silver. Although those digital currencies worked well, they were centralized (issued by single clearing houses or entities) and therefore easy targets for malicious hackers and opposing governments.

This has changed in October 2008, when a pseudonymous developer Satoshi Nakamoto published a whitepaper titled “Bitcoin: A Peer-to-Peer Electronic Cash System” [1]. Bitcoin was presented as a completely decentralized digital payment system without relying on any central authorities or control points to deal with the currency. One of the most important and integral parts of the Bitcoin network, which was established in January 2009, is an intensive and difficult computational process called mining.

So-called miners are volunteers (competitors) who run their mining hardware called mining rigs to secure the Bitcoin network by processing transactions and attempting to chain blocks of transactions into a public ledger called blockchain (Fig. 1). If they are successful at adding blocks to the blockchain, they are rewarded with a certain amount of digital coins called bitcoins which can be traded for real-world currency at high price.

Each block in the blockchain contains its previous block header’s hash to ensure chronological order of blocks. It also contains the Merkle root which is a hash of all transaction hashes from the particular block and it is used for a quick and efficient verification of transactions.

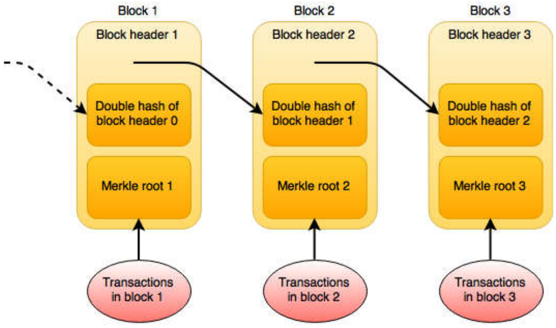


Fig. 1: A simplified view of the blockchain [2].

Bitcoin mining rigs improved rapidly in only few years since the Bitcoin network was established [3]. In its early stage (2009), bitcoins were mined with desktop computers and laptops (CPU, Central Processing Unit). In years 2010-2011, gaming cards (GPU, Graphics Processing Unit) were used for mining as those were more suitable for repetitive and parallel computing, which resulted in about 50 to 100 times faster and more energy-efficient mining than with CPUs. In years 2011-2012, few companies started redesigning and selling modified FPGA (Field-Programmable Gate Array) boards for bitcoin mining; those were equally fast at mining as GPUs, but consumed about 5 times less electrical energy. In 2013, the first custom ASIC (Application-Specific Integrated Circuit) mining rigs were developed for the sole purpose of mining bitcoins; those approved to be at least 1000 times faster and about 100 times more energy-efficient than FPGA, and have thus been widely used for mining bitcoins ever since.

Our primary objective was to assess the capabilities and energy efficiency of Maxeler dataflow computers for bitcoin mining. During our research and the implementation of the bitcoin miner, we have followed the creativity method known as *specialization* [4]. The method starts from a well-established general approach, which is used to derive a specific knowledge for a specific domain or task. In this paper we have used and implemented a known bitcoin mining algorithm for Maxeler dataflow computers.

Our expectations, based on experience with some other dataflow applications [5-10], were that dataflow implementation of bitcoin mining would outperform the CPU and GPU solutions, but would be inferior to ASIC solutions [11,12].

The existing CPU and GPU solutions focus primarily on speed and use a lot of energy to get to the result. ASIC solutions are unbeatable in both speed and energy efficiency, but they are not flexible; they are designed to solve one problem only! Our solution employs dataflow computers that can perform highly parallelized computation at high speeds and energy efficiency. As shown by the results presented in section 5, our solution outperforms CPU and GPU solution in both speed and energy efficiency. While ASIC is still better at both of the aforementioned parameters, the solution on Maxeler dataflow computers is flexible and reprogrammable. Such advantage is indispensable and invaluable for solutions where ASICs do not exist (yet): for example, in applications where the development of an ASIC solution is not cost efficient, in new and emerging cryptocurrencies where ASICs are not available yet, and for the solutions where ASICs are not implementable at all.

2. Approach to Mining

It was very easy to mine bitcoins individually in early years of the Bitcoin network (2009-2010). However, more miners with better mining rigs joined the bitcoin mining process over time, and mining bitcoins became so difficult that it was only profitable by forming mining pools. Miners contribute their mining rigs' computational power (measured as hash rates) and share rewards in bitcoins.

2.1 Shares

Mining rigs generate partial solutions called shares which may not satisfy a global difficulty target set by the Bitcoin network, but may satisfy an easier local difficulty target set by the mining pool. Sometimes, a share is generated which satisfies both the mining pool's easier (local) difficulty target and the Bitcoin network's harder (global) difficulty target. The mining pool uses it to create a valid block with transactions which is then propagated to the Bitcoin network. If it is added to the blockchain and a few more blocks are added onto it (thus making it immutable), the reward belongs to the winning mining pool, and it is shared among all participating miners based on their provided computational power (hash rates).

2.1.1 Hashing Block Header

Block header is a 640 bits long data structure, which consists of six fields (Table 1) and it is an important part of a block filled with transactions; bitcoin miners repeatedly apply SHA-256 (Secure Hashing Algorithm 256) [13] on the block header twice while changing its nonce value in order to find a 256-bit hash that is less than the global difficulty target of the Bitcoin network; simply said, a hash has to start with several leading zeros, for example `0000000000000001e8d6829a8a21adc5d38d0a473b144b6765798e61f98bd1d`.

Table 1: Structure of block header [3]

Field	Description	Size (bits)
Version	Block version number	32
Previous block header's hash	Double SHA-256 hash of the previous block header	256
Merkle root	SHA-256 hash as the summary of all transactions in the block	256
Timestamp	Current timestamp in seconds since 1. 1. 1970, 00:00 UTC	32
Difficulty target	Current target in compact format	32
Nonce	A variable used for proof-of-work mining algorithm	32

3 Dataflow Computing

CPU frequency has stopped increasing several years ago due to hitting physical limitations at shrinking transistors, increased energy consumption and generated heat. Parallel computing is, therefore, being researched as new approach to process big data. One of the alternatives is dataflow computing using Maxeler dataflow computers, where data (bits) simply flow from inputs to outputs through a field of simple arithmetic units which execute simple computational operations [14-18].

3.1 Dataflow Engines

One or more DFEs (Dataflow Engine) are part of workstation computers and are dedicated to accelerating algorithms (Fig. 2). The workstation computer CPUs execute regular operations, run control-flow programs, and control communication between entities while DFEs run parallel dataflow computation.

DFE consists of an integrated FPGA circuit and several types of memory; FMem (Fast Memory) can be found directly on the chip and it can store few megabytes of data with very fast access, and LMem (Large Memory) can be found outside of the chip and can store several gigabytes of data.

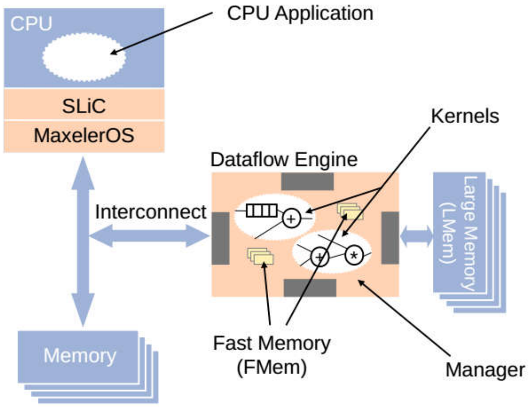


Fig. 2: Structure of Dataflow Engine [14]

3.1.1 *Arithmetic units*

DFE contains thousands of simple arithmetic units which execute simple operations (e.g. addition, multiplication, bit shift) on moving data. Those arithmetic units are connected together into a corresponding dataflow graph [14,18], see Fig. 3.

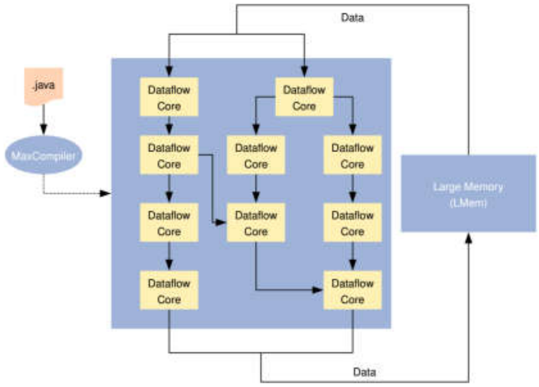


Fig. 3: Principle of dataflow computing [14]

Dataflow computing is executed in time steps called ticks. On every tick, input data move from current arithmetic units to the next ones and data are one step closer to the output.

In contrast with sequential control-flow computing where operations are executed at different times on same functional parts ("computing in time"), dataflow computing is distributed on the whole chip ("computing in space").

4 DATAFLOW IMPLEMENTATION

Our bitcoin miner application was implemented and tested on two Maxeler dataflow computers, MAX2B and MAX5C (Table 2). Our application consists of two parts (Fig. 4):

- A control-flow program (written in C and run by workstation CPU) controls the communication with bitcoin mining pools and supplies required data to DFE;
- A dataflow program (written in MaxJ and run by DFE) runs the bitcoin mining algorithm where appropriate nonce values are searched for by hashing block headers with SHA-256 hashing algorithm while changing its nonce value and comparing its hashes against the difficulty target.
- Both parts of our code have access to a mapped memory (FMem) that contains nonce values. A dataflow program (DFE) writes suitable nonce values into it while the control-flow program (CPU) reads nonce values from it and uses those values to create and send shares to the mining pool.

Table 2: Workstation properties, DFE properties, and DFE resource usage for implementation

	MAX2B	**MAX5C**
FPGA	Virtex-5 XC5VLX330T	Altera Stratix V *5SGSMD8N2F45C2*
LMem	12 GB DDR2	48 GB DDR3
Workstation CPU	Intel Core 2 Quad Q9400, 2.66 GHz	Intel Core i7-6700K, 4.00 GHz
Operating system	Linux CentOS 6.5	Linux CentOS 6.9
PCI Express	x4	x8
MaxIDE Version	2013.3	2015.2
Pipelines	3	7
Stable frequency	95 MHz	210 MHz
Expected hash rate	285 Mhash/s	1470 Mhash/s
Actual hash rate	282 Mhash/s	1430 Mhash/s
Logic utilization	199295 / 207360 (96.11%)	225083 / 262400 (85.78%)
Look-Up tables (LUT)	178919 / 207360 (86.28%)	/
Primary flip-flops (FF)	169889 / 207360 (81.93%)	387413 / 524800 (73.82%)
Secondary flip-flops (FF)	/	49456 / 524800 (9.42%)
Multipliers	0 / 192 (0.00%)	0 / 3926 (0.00%)
Digital Signal Processors (DSP)	0 / 192 (0.00%)	0 / 1963 (0.00%)
Block memory	68 / 648 (10.49%)	1268 / 2567 (49.40%)

The actual hash rate *hr* is approximate to the product of stable frequency *f* and number of pipelines *N* (Table 2), as shown in Equation (1).

$$hr[\tfrac{hash}{s}] \approx f[Hz] * N \qquad (1)$$

Because our bitcoin miner application was realized with multiple pipelines, multiple nonce values could be tested per each tick (3 on MAX2B and 7 on MAX5C board), which led to a massive acceleration of the bitcoin mining algorithm (Fig. 5).

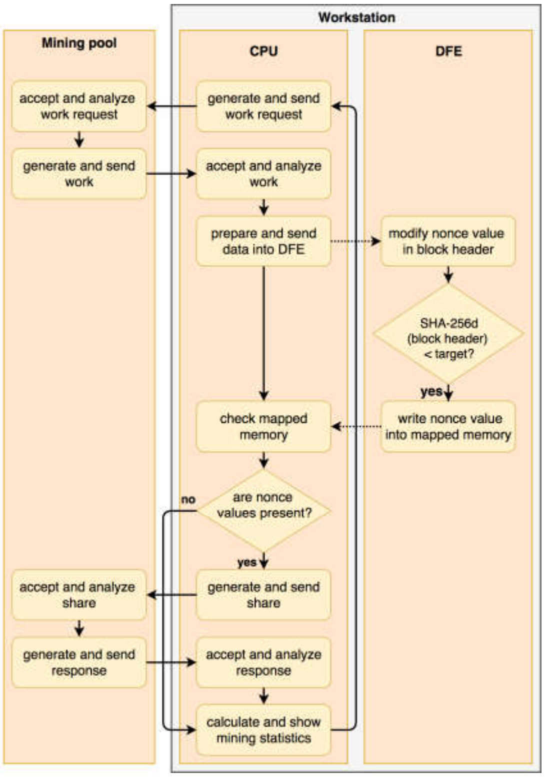

Workstation
Mining pool
CPU
DFE
accept and analyze work request
generate and send work request
generate and send work
accept and analyze work
prepare and send data into DFE
modify nonce value in block header
SHA-256d (block header) < target?
yes
check mapped memory
write nonce value into mapped memory
no
are nonce values present?
yes
accept and analyze share
generate and send share
generate and send response
accept and analyze response
calculate and show mining statistics

Fig. 4: Workflow of bitcoin miner application

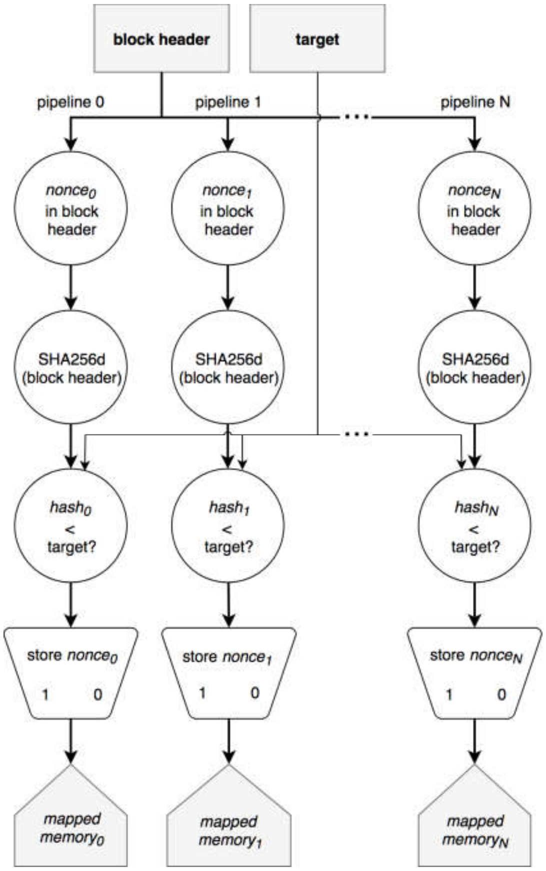


Fig. 5: Abstraction of bitcoin mining algorithm in DFE; testing multiple nonce values per tick

5 *Performance Analysis*

We measured hash rates and electric power of several mining rigs (Table 3) during bitcoin mining, calculated their energy efficiency, and compared the results.

Table 3: Configurations of tested mining rigs

Hardware	Software
CPU Intel 2 Core Quad Q9400, 2.66 GHz	CPUminer - minerd 2.4.5 (open-source)
CPU Intel i7-6700K, 4.00 GHz	CPUminer - minerd 2.4.5 (open-source)
GPU AMD Radeon 6950 HD	Cgminer 3.7.2 (open-source)
DFE MAX2B	implemented
DFE MAX5C	implemented

5.1 Relative computation speed

One of the main drawbacks of our bitcoin miner application was the requirement to stream dummy output elements in order to actually run computation in DFEs even though in our case there was no need for streaming output elements.

Dataflow computing with DFE is quite inefficient at processing small data sets compared to control-flow computing with CPU. This is because streaming data between a workstation computer and DFE takes some time; input data have to be streamed from a workstation computer to DFE inputs, then through dataflow graphs (inside DFE) to DFE outputs, and, finally, result data have to be streamed back to the workstation computer. In our example, it took at least 685 µs for MAX2B board and at least 572 µs for MAX5C board to stream 16 dummy output elements (minimum for MAX5C board) from DFE outputs back to the workstation computer.

In order to achieve the top computation speed, the time needed to stream data between the workstation computer and DFE has to be made negligible compared to the time needed to complete computation. This can be achieved by processing large data sets in order to run dataflow computation on DFE for a long time.

Both MAX2B and MAX5C boards achieved the top computation speed at streaming at least 2^{27} (134,217,728) dummy output elements (Fig. 6).

As mentioned earlier, our bitcoin miner application was realized with 3 pipelines on MAX2B board and with 7 pipelines on MAX5C board, and therefore MAX2B board would have to test at least $3{*}2^{27}$ (402,653,184) nonce values and MAX5C board would have to test at least $7{*}2^{27}$ (939,524,096) nonce values in order to reach the top computation speed.

Since all 2^{32} (4,294,967,296) nonce values have to be tested per single block header, the amount of dummy output data did not pose an issue for dataflow computing performance.

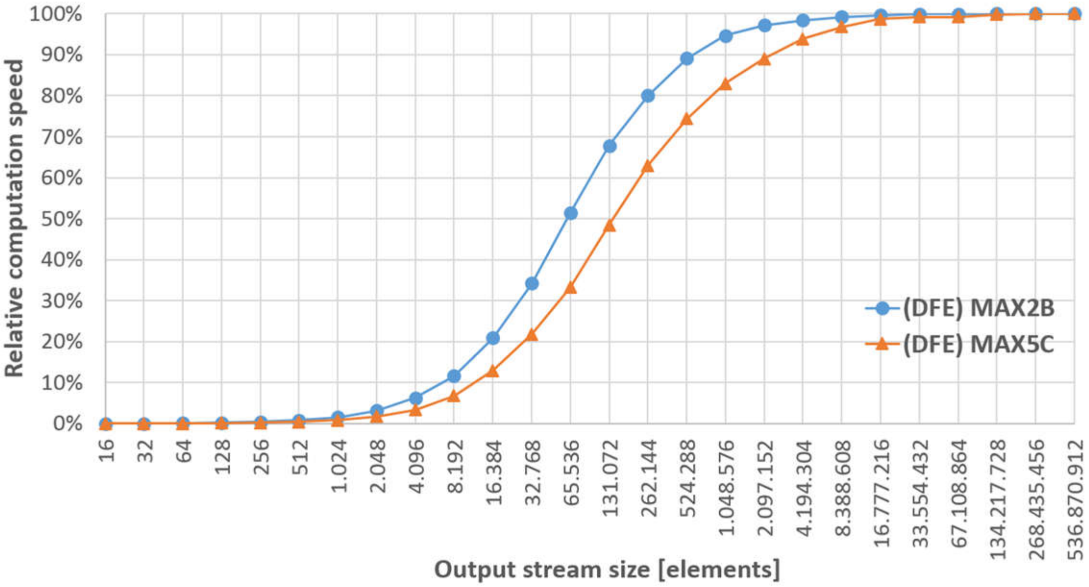


Fig. 6: Relative DFE computation speed based on output data stream size.

5.2 Hash rate

Hash rate represents a quantity of double SHA-256 hash computations per second, and is given in derived units, such as [h/s] or [hash/s]. As hash rate is proportional with generated income for miners, the higher hash rate, the better (Fig. 7).

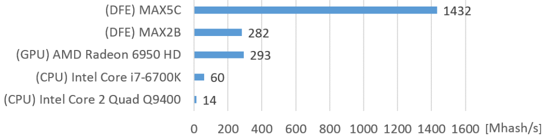


Fig. 7: Measured hash rates (higher is better)

MAX2B board was slightly slower at mining than GPU, and up to 20 times faster than CPU, while MAX5C board was about 5 times faster than both MAX2B board and GPU, and up to 102 times faster than CPU.

5.3 Electric power

Electric power is the rate of electric energy consumption in an electrical circuit per time unit and is generally given in watts [W]. Mining rigs may consume a lot of electric energy that has to be paid for, and therefore the lower electric power, the better (Fig. 8).

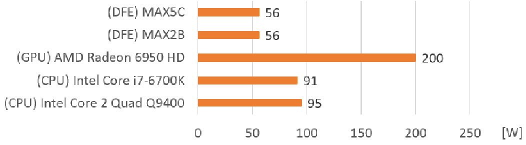


Fig. 8: Measured (DFE) and theoretical (CPUs and GPU) electric powers (lower is better)

Electric power of mining rigs was measured in different ways:

- DFE MAX2B:
  Measured with an energy consumption meter *VOLTCRAFT Energy Logger 4000;*
- DFE MAX5C:
  terminal command in Linux CentOS *maxtop –v*;
- GPU and CPU:
  TDP (Thermal Design Power) which are theoretical values given in hardware specifications. Empirical measures were taken in [19] and proved to be approximate to TDP values.

5.4 Energy efficiency

The most important information about a mining rig is its energy efficiency during mining. It is calculated as a ratio between its hash rate and electric power as shown in Equation (2),

$$eff\left[\text{hash/s/W}\right]=\frac{hr\left[\text{hash/s}\right]}{P\left[\text{W}\right]} \qquad (2)$$

where *eff* is the calculated energy efficiency, *hr* is hash rate and *P* is electric power of the mining rig. Since it is preferred to run mining rigs with higher hash rate and lower electric energy consumption, the higher energy efficiency is, the better a mining rig is (Fig. 9).

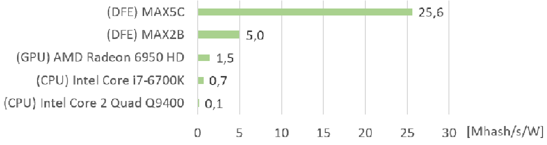


Fig. 9: Calculated energy efficiency (higher is better)

MAX2B board was 3 times more energy-efficient than GPU and up to 50 times more efficient than CPU. Meanwhile, MAX5C board was 5 times more energy-efficient than MAX2B board, 17 times more energy-efficient than GPU and up to 256 times more energy-efficient than CPU.

5.5 ASIC Mining Rigs

MAX5C board performed best at mining bitcoins in all three categories (hash rate, electric power, and energy efficiency). However, when compared to a specialized ASIC mining rig, e.g. Antminer S7 with 4.73 Thash/s at 1300 W (which yields 3.64 Ghash/s/W) [20], Antminer S7 mining rig appears to be 3303 times faster and 142 times more energy-efficient than MAX5C board despite the fact it consumes 23 times more electric energy.

6 CONCLUSION

In this paper, we implemented our dataflow implementation of bitcoin miner algorithm on two Maxeler dataflow computers, MAX2B and MAX5C, and compared their performance to CPU and GPU based solutions in terms of computation speed (hash rate), electric power, and energy efficiency. We achieved up to 102 times higher hash rate and up to 256 times higher energy efficiency at mining compared to CPU and GPU based solutions due to the exploit of massive parallelism of both bitcoin mining algorithm and Maxeler dataflow computers.

Unfortunately, mining bitcoins with Maxeler dataflow computers has not been profitable since year 2013, when ASIC mining rigs were deployed and took over bitcoin mining due to their superior computation speed (hash rate) and energy efficiency. ASIC mining rigs have increased bitcoin mining difficulty to the extent that regular bitcoin miners who do not own ASIC mining rigs moved onto mining less known and more computer friendly alternative cryptocurrencies, also known as altcoins (e.g. litecoin, ethereum).

General purpose Maxeler dataflow computers cannot compete against customized ASIC circuits in terms of computation speed and energy efficiency. However, Maxeler dataflow computers are flexible and can be reprogrammed for other computation, e.g. mining altcoins with very low electric energy consumption for which ASIC solutions do not yet exist or are currently cost prohibitive to implement.

ACKNOWLEDGMENT

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0246). We thank Belgrade team at Maxeler Technologies for providing us access to Maxeler MAX5C board for testing purposes.

REFERENCES

[1] Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed: 1.7.2017].

[2] "Developer guide - Bitcoin," 2009. [Online]. Available: https://bitcoin.org/en/developer-guide. [Accessed: 1.7. 2017].

[3] Antonopoulos, A., Mastering Bitcoin: Unlocking Digital Cryptocurrencies. Sebastopol: O'Reilly Media, 2015.

[4] Blagojević, V., et al, "A Systematic Approach to Generation of New Ideas for PhD Research in Computing," Advances in Computers, Elsevier, Vol. 104, 2016, pp. 1-19.

[5] "Apps | Maxeler AppGallery". [Online]. Available: http://appgallery.maxeler.com/. [Accessed: 1.7.2017].

[6] Trifunović, N., Milutinović, V., Korolija, N. and Gaydadjiev, G., 2016. An AppGallery for dataflow computing. Journal of Big Data, 3(1), p. 4.

[7] Ranković, V., Kos, A., Tomažič, S. and Milutinović, V., "Performance of the bitonic mergesort network on a Dataflow computer," 2013 21st Telecommunications Forum Telfor (TELFOR), Belgrade, 2013, pp. 849-852.

[8] Kos, A., Ranković, V., and Tomazič, S. (2015). Sorting Networks on Maxeler Dataflow Supercomputing Systems. Advances in Computers, 96, pp. 139-186.

[9] Kotlar, M. and Milutinović, V., "Implementing Neural Networks Using the DataFlow Paradigm," IPSI BgD Transactions on Advanced Research (TAR), January 2017, Volume 13, Number 1, ISSN 1820 – 4511

[10] Milanković, I., Mijailović, N., Peulić, A., and Filipović, N., "Application of Data Flow Engines in Biomedical Images Processing," IPSI BgD Transactions on Advanced Research (TAR), January 2018, Volume 14, Number 1, ISSN 1820 – 4511

[11] L. Reese, "Comparing Hardware for Artificial Intelligence: FPGAs vs. GPUs vs. ASICs | Embedded Intel® Solutions", Eecatalog.com, 2018. [Online]. Available: http://eecatalog.com/intel/2018/07/24/comparing-hardware-for-artificial-intelligence-fpgas-vs-gpus-vs-asics/. [Accessed: 1.8.2018].

[12] Amara A., Amiel F., Ea T., "FPGA vs. ASIC for low power applications", 2006, Microelectronics Journal, 37, pp. 669–677

[13] "Descriptions of SHA-256, SHA-384, and SHA-512,". [Online]. Available: http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf. [Accessed: 1.7.2017].

[14] Maxeler Technologies, "Multiscale Dataflow Programming," 2015

[15] Milutinović, V., Furht, B., Obradović, Z. and Korolija, N., 2016. Advances in High Performance Computing and Related Issues. Mathematical Problems in Engineering, 2016.

[16] Korolija, N., Popović, J., Cvetanović, M. and Bojović, M., 2017. Dataflow-Based Parallelization of Control-Flow Algorithms. In Advances in Computers (Vol. 104, pp. 73-124). Elsevier.

[17] Milutinović, V., Salom, J., Veljović, D., Korolija, N., Markovic, D. and Petrovic, L., 2017. Transforming Applications from the Control Flow to the Dataflow Paradigm. In DataFlow Supercomputing Essentials (pp. 107-129). Springer, Cham.

[18] Milutinović, V., Salom, J., Veljovic, D., Korolija, N., Markovic, D. and Petrovic, L., 2017. Discrepancy Reduction Between the Topology of Dataflow Graph and the Topology of FPGA Structure. In DataFlow Supercomputing Essentials (pp. 19-66). Springer, Cham.

[19] Meden, R., "Mining Bitcoins using Maxeler Data Flow Computer", M.S. thesis, Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia, 2017

[20] "Bitmain Antminer S7 Review: Is it Profitable to Buy? (Probably Not)," [Online]. Available: https://www.buybitcoinworldwide.com/mining/hardware/antminer-s7/. [Accessed: 1.7.2017].

Rok Meden received his B.S. and M.S. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 2013 and 2017, respectively. His research interests cover deep web, cryptocurrencies (mining algorithms) and dataflow computing with Maxeler systems.

Anton Kos received his Ph.D. in electrical engineering from University of Ljubljana, Slovenia, in 2006. He is an assistant professor at the Faculty of Electrical Engineering, University of Ljubljana. He is a member of the Laboratory of Information Technologies at the Department of Communication and Information Technologies. His teaching and research work includes communication networks and protocols, quality of service, dataflow computing and applications, usage of inertial sensors in biofeedback systems and applications, signal processing, and information systems. He is the (co)author of thirty papers in international engineering journals and of more than fifty papers presented at international conferences.

Variety of DataFlow Computing Applications

Korolija, Nenad; Milutinović, Veljko; and Popović, Jovan

Abstract: ***DataFlow computing has drawn noticeable attention of researchers in recent years, due to advancements in DataFlow hardware technology. Most of the work in this field focuses on describing the technology and specific algorithm performance improvements, measured both in execution time and power consumption. Some of the work compares different DataFlow hardware applications from the same point of view. This paper presents a variety of DataFlow computing hardware implementations, and shows one representative application for each of them. Examples include applications for simulating natural DataFlow processes, uniform processing of data, simulating the neuron, and network data manipulation. Results show that DataFlow hardware could be utilized for various applications and not only for CPU demanding applications with uniform data processing.***

Index Terms: ***DataFlow paradigm, high performance computing, network programming***

I. *Introduction*

FOR many decades, the control flow computing paradigm has been dominant in most aspects of computing, including general applications with non-uniform processing, high performance applications with uniform processing (where the same set of instructions is run over and over again), as well as network programming, where data arriving from the network should be processed and the results should be returned to the network. The main reason can be found at the very beginning of the development of computer architectures. Namely, although processes in nature happen in parallel, the process of planning a task usually includes splitting the task which has to be completed into parts, and focusing on each of the parts separately. Similarly, when a certain algorithm has to be run, it is easier to focus on one specific action (i.e. a single instruction) which has to be executed at a particular moment than to model the whole system. Even if we would think differently, an algorithm could often not be run in parallel at all. Not only is it easier to think of executing a single action at a time, but it is also easier to implement the hardware that would be capable of executing one instruction at a time, while offering the possibility to run any instruction at any moment, than to develop the DataFlow hardware for each application separately, which would be applicable to a set of high performance computing algorithms, where the same set of instructions is repeated many times. Therefore, it is logical that the process of computer hardware development included developing processors based on the control-flow paradigm. It should be noted that the DataFlow paradigm existed from the very beginning, but the technological limitations influenced the development of early computers. Later, when the technology improved, the inertia related to producing hardware, and even more related to programming practices led to the continuous domination of the control-flow paradigm. However, since business margins tend to shrink, the industry recognized the advantages of the DataFlow computing paradigm, which was later followed by researchers [1]. Changes in computer architectures and programming paradigms are not new − they already happened many times in the past [2, 3, 4, 5].

II. *Control-flow vs. DataFlow Computing*

Control-flow computer architectures are based on von-Neumann model or Princeton architecture, which dates back to 1945. Control-flow processors are capable of executing all types of instructions defined by the architecture. This imposes having an arithmetical logical unit and additionally a control unit. Figure 1 depicts this architecture, where both data and instructions are loaded from the main memory into the CPU, and depending on the fetched instruction, the CPU processes the data.

When executing a code which could be run in parallel, the main drawback is obviously the fact that the processor is capable of executing only a single instruction at any given moment. This problem is partially solved by introducing pipelines, multi- and many-core processor architectures, etc.

Manuscript received December 1, 2018. This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, grant 451-03-00605/2012-16/198, project “Cloud Services for Applications with High Performance Requirements,” by School of Electrical Engineering, University of Belgrade, and Ministry of the Republic of Serbia number TR32047.

Nenad Korolija is with the School of Electrical Engineering, University of Belgrade (e-mail: nenadko@gmail.com).

Veljko Milutinović is Fellow of the IEEE (USA), Honorary Treasurer of Academia Europaea (GBR), Member of the Serbian National Academy of Engineering (SRB), Foreign Member of the Montenegro Academy of Sciences and Arts (MNE), Department of Computer Science, Indiana University, Bloomington, USA, Mathematical Institute, Serbian Academy of Sciences and Arts, SRB,

Senior Advisor to Maxeler Technologies in London, GRB, and CEO of IPSI Belgrade and Member of the Board of MECOnet, SRB and MNE.

Jovan Popović is with Microsoft Development Center Serbia.

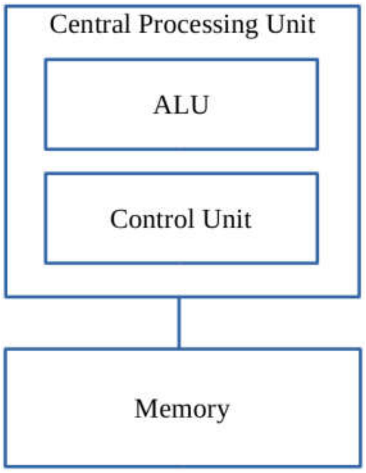


Fig 1. Von Neumann computer architecture

However, one of the main problems with such architecture is the necessity to access the main memory more often than it is needed. This problem is partially solved by introducing cache memories and techniques for tolerating memory latencies [6, 7, 8, 9, 10, 11, 12, 13]. Nevertheless, that comes at a cost. The numbers of transistors spent on cache memories of modern processors are close to the numbers of transistors which processors have for all other functionalities.

Unlike control-flow computer architectures, DataFlow architectures treat executing instructions as a factory floor, where pieces (in this case data) travel through the processor, and at each place, a certain instruction is executed. Therefore, the DataFlow hardware has preconfigured instructions, and data travels through the hardware.

This is shown in the DataFlow graph presented in Figure 2, where a stream of data is shown at the top, and a stream of produced results at the bottom. As it can be seen from the figure, DataFlow hardware can execute many instructions in parallel, offering superior performance compared to the control-flow processor.

The advantages of this approach are rather obvious. Many instructions can run in parallel. Another advantage is that the hardware that will only be capable of running a single instruction is small enough, since it is less complex. This leads to significantly lower power consumption compared to control-flow computer architectures. Using the DataFlow hardware, researchers have reported acceleration of even two or three orders of magnitude [14, 15, 16, 17, 18].

However, DataFlow hardware requires uniform processing in order to be fully unitized. Most high-performance computing applications include uniform processing, but also initialization of data. When it comes to the size of the code necessary for the part of the algorithm which should run in parallel and the rest of the algorithm which includes the initialization of data, storing results and other operations which should be run only once (before or after the part of the algorithm that should run in parallel), the size of the former one is usually much smaller. On the other hand, when it comes to execution time, the part which should be run in parallel usually takes much longer than the rest of the algorithm. Typically, the rest of the algorithm is responsible for less than 1% of total execution time. Since DataFlow hardware is not suitable for this part of the algorithm, DataFlow hardware is usually combined with the control-flow processor or processors in such a way that only the part of the algorithm which should run in parallel is executed on the DataFlow hardware. DataFlow hardware could be connected to the PCIe bus, or it may exist in a rack. In either case data is streamed from the control-flow type of processor to DataFlow hardware, where it is processed, after which the results are returned to the control-flow type of processor.

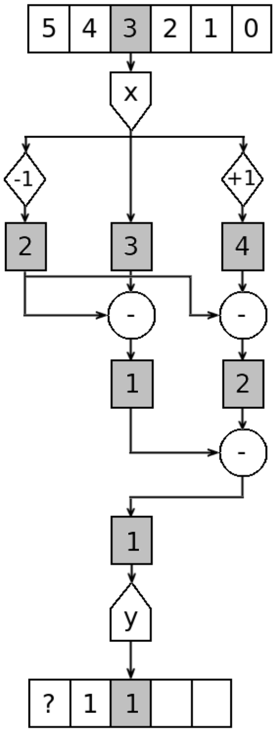


Fig 2. DataFlow graph

Another limitation of DataFlow hardware is that it is capable of executing only the algorithm for which it is developed or configured. Therefore, if one wants to use the same DataFlow hardware for running various algorithms, DataFlow hardware must be reconfigurable. Usually, this is realized by means of FPGA technology, which is developing [19, 20] and we could therefore expect better performances in the future. For now, using the FPGA limits the frequency of DataFlow hardware, but it also further reduces power consumption.

There are various implementations of DataFlow architectures depending on their purpose. One possible configuration is shown in Figure 2.

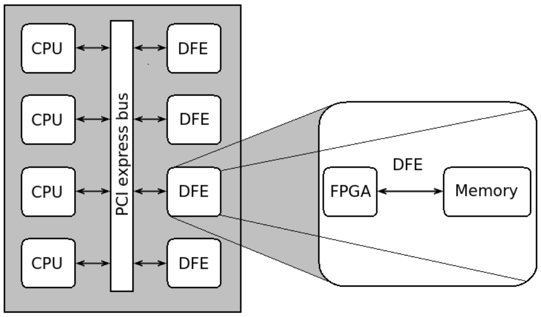


Fig 3. DataFlow computer architecture

Early computers were able to solve simple operations, while anything more complex than that could be simulated using available instructions. Later, it was easier to follow researchers and programming practice by improving the already achieved, utilizing available technology, tools, and programming languages no matter how primitive they were, than to implement a new programming paradigm from scratch. The same pattern continued for decades. Nowadays there are tools for transforming various control-flow types of algorithms into DataFlow algorithms [21].

It is important to note that DataFlow architectures have existed basically from the very beginning of computing. However, the technology at that time was not developed enough to make it competitive. Therefore, even the processes which were best suited for DataFlow architectures were still executed on control-flow architectures, by executing a single instruction at a time.

However, implementing high performance computing applications often justifies switching the technology and exploiting the best suiting one. Even more, sometimes is worth implementing applications in hardware directly. Recent examples include high frequency trading and mining virtual currencies. As a result, various types of architectures have been introduced.

III. *Types of DataFlow Architectures*

This section describes a variety of applications of DataFlow architectures, where each application is implemented for specific DataFlow architecture. Note that new DataFlow architectures could be implemented on demand.

A. Natural DataFlow Processing

As already stated, many processes in nature happen in parallel. Accordingly, many algorithms which are designed to simulate natural processes could be run in parallel. Examples include simulations which enable modern weather forecasting, simulating objects in a wind tunnel [22], simulating fluids [23], or simulating casting of materials. In each of these cases, the volume that is simulated can be divided into elementary cubicles, each defined by the same parameters. For each elementary cubicle, a set of equations could be used for calculating the state of the elementary cubicle based on the surrounding elementary cubicles. In the simulation the same set of instructions are repeatedly executed in order to simulate the natural behavior of each of the elementary cubicles. It is fairly obvious that DataFlow hardware could be utilized for uniform processing in order to achieve higher instruction throughput, as well as lower power consumption.

An example is a network sorting application [24]. The main principle lies in the fact that parts of an array could be sorted in parallel, which justifies using DataFlow architectures for relatively big arrays.

B. Uniform Processing of Data

When we talk about the DataFlow paradigm, we typically think of simulating natural processes. Nevertheless, certain algorithms also include uniform processing of data, and are not used for simulating natural processes. For example, some processing of all of the accounts in a bank requires uniform processing over a relatively big set of data. Another example would be processing of links by search engines. Another example includes improving the price-performance ratio for estimating the value and risk of large and complex credit derivatives [25]. Basically, much of the information produced by relatively big companies requires uniform processing of relatively big amounts of data. Therefore, it is justified that these companies should lead the future developments of computer architectures.

C. Non-uniform Processing of Data

There are many examples when an algorithm can be reimplemented, so that it would become more suitable for running on modern computer architectures which are capable of processing data in parallel on many cores. In many cases, these reimplemented algorithms would include uniform processing of data, for which DataFlow hardware can be efficiently utilized.

For example, when simulating the brain, which consists of a relatively high number of neurons, each connected with a certain number of its neighboring neurons, one could implement an algorithm so that data processing which each neuron is performing at the moment happens in parallel. If the maximum number of connected neurons is relatively close to the average number of connected neurons, it is even possible to develop the DataFlow hardware which would be capable of simulating the network of neurons as if the number of connected neurons was always equal to the maximum number of connected neurons. Although this leaves some of the DataFlow hardware underutilized, DataFlow hardware utilization could still easily be much higher than it would be using the control-flow hardware.

Therefore, while it might not be obvious that an algorithm could be efficiently implemented using DataFlow hardware, sometimes a modified algorithm that might not be optimal could be suitable for DataFlow hardware and still performing better than the original algorithm using control-flow hardware.

Example applications include brain network simulation for modeling brain activity by extracting a dynamic network from slices of a mouse's brain. This application implements a linear correlation analysis of brain images. Although the network is not uniform in shape and the number of connections from each node, DataFlow computing can offer better performances than a control-flow counterpart [26].

D. Network programming

At the time when control-flow hardware was introduced, the main focus was on data processing. However, recent needs for computer processing often include processing of network data. In some cases, it is required that the processing of the data if fast enough. Examples include high frequency trading and hardware firewall implementations.

Typically, network data processing using the control-flow type of processor requires bringing the data to the processor, processing itself, and sending the data over the network to the consumer. If the amount of required processing is small enough, the cost of bringing the data to and from the processor might be too high in terms of the needed hardware, as well as the time needed from the moment when the data reaches the network card until the moment when the processed data reaches the network.

DataFlow hardware could be directly connected to the network, offering superior performances compared to its control-flow counterpart. A typical example of network application could be the one which measures network latency [27].

IV. Conclusion

As a result of technological advancements, DataFlow computing has gained in popularity in recent years. This paper focuses on the variety of applications which lead to the development of multiple DataFlow computing hardware implementations. For each of them, a representative application is shown. In case of simulating natural DataFlow processes and uniform processing of data, DataFlow hardware is superior. In case of non-uniform processing of data, DataFlow hardware could achieve noticeable improvements, but only in case of certain algorithms. A network application which requires a fast response would not be suitable for typical DataFlow hardware implementations, but could have a dramatically faster response when run on the DataFlow hardware which is directly connected to the network. Therefore, DataFlow hardware can be utilized for various applications, not only for the uniform processing of data which simulates natural processes.

Acknowledgment

This research was supported by School of Electrical Engineering, University of Belgrade, and projects of Ministry of the Republic of Serbia number TR32047 and III 44006.

References

[1] Korolija, N., Popović, J., Cvetanović, M. and Bojović, M., “DataFlow-Based Parallelization of Control-Flow Algorithms,” In Advances in Computers, Vol. 104, pp. 73-124, Elsevier, 2017.

[2] Milutinovic, V., "High-level language computer architecture," Computer Science Press, Inc., 1989.

[3] Milutinovic, V., "Surviving the Design of a 200 MHz RISC Microprocessor: Lessons Learned," IEEE Computer Society, 1996.

[4] Kos, A., Tomažič, S., Salom, J., Trifunovic, N., Valero, M. and Milutinovic, V., "New benchmarking methodology and programming model for big data processing," International Journal of Distributed Sensor Networks, Vol.11, 2015.

[5] Trifunovic, N., Milutinovic, V., Salom, J. and Kos, A., "Paradigm shift in big data supercomputing: DataFlow vs. controlflow," Journal of Big Data, Vol. 2, 2015, pp. 4.

[6] Milenkovic, A. and Milutinovic, V., "Cache injection: A novel technique for tolerating memory latency in bus-based SMPs," European Conference on Parallel Processing, Springer, Berlin, 2000, pp. 558-566.

[7] Furht, B. and Milutinovic, V., "A survey of microprocessor architectures for memory management," Computer, Vol. 20, 1987.

[8] Tomasevic, M. and Milutinovic, V., "Hardware approaches to cache coherence in shared-memory multiprocessors," IEEE Micro, Vol. 14, 1994, pp. 61-66.

[9] Grujic, A., Tomasevíc, M. and Milutinovic, V., "A simulation study of hardware-oriented DSM approaches," IEEE Parallel & Distributed Technology: Systems & Applications, Vol 4, 1996, pp.74-83.

[10] Tomasevic, M. and Milutinovic, V., "A simulation study of snoopy cache coherence protocols," System Sciences, Proceedings of the Twenty-Fifth Hawaii International Conference, Hawaii, Vol 1, 1992, pp. 427-436.

[11] Tartalja, I. and Milutinovic, V., "The Cache Coherence Problem in Shared-Memory Multiprocessors: Software Solutions," IEEE Computer Society Press, 1997.

[12] Milutinovic, V., "Caching in distributed systems," IEEE Concurrency, Vol. 8, 1997, pp.14-15.

[13] Milutinovic, V. and Stenstrom, P., "Special issue on distributed shared memory systems," Proceedings of the IEEE, Vol. 87, 1999, pp.399-404.

[14] Stojanovic, S., Bojic, D., and Milutinovic, V., “Solving Gross Pitaevskii Equation Using DataFlow Paradigm,” Transactions on Internet Research, Vol. 9, No. 2, July 2013.

[15] Rankovic, V., Kos, A., Milutinovic, V., "Bitonic Merge Sort Implementation on the Maxeler DataFlow Supercomputing System," Transactions on Internet Research, Vol. 9, No. 2, July 2013, pp. 34-42.

[16] Stanojevic, I., Senk, V., and Milutinovic, V., "Application of Maxeler DataFlow Supercomputing to Spherical Code Design," Transactions on Internet Research, Vol. 9, No. 2, July 2013, pp. 1-4.

[17] Bezanic, N., Popovic-Bozovic, J., Milutinovic, V., and Popovic, I., "Implementation of the RSA Algorithm on a DataFlow Architecture," Transactions on Internet Research, Vol. 9, No. 2, July 2013, pp. 11-16.

[18] Gan, L., Fu, H., Mencer, O., Luk, W. and Yang, G., “Data flow computing in geoscience applications,” In Advances in Computers, Vol. 104, Elsevier, 2017, pp. 125-158.

[19] Milutinovic, V., "Guest Editor's Introduction GaAs Microprocessor Technology," Computer, Vol. 10, 1986, pp.10-13.

[20] Milutinovic, V., Fura, D., Helbig, W. and Linn, J., "Architecture/compiler synergism in GaAs computer systems," Computer, Vol 20, 1987.

[21] V. Milutinovic, J. Salom, D. Veljovic, N. Korolija, D. Markovic, L. Petrovic, “Transforming Applications from the Control Flow to the DataFlow Paradigm,” In DataFlow Supercomputing Essentials, Springer, Cham, 2017, pp. 107-129.

[22] N. Korolija, V. Milutinovic, S. Milosevic, “Accelerating Conjugate Gradient Solver: Temporal Versus Spatial Data,” The IPSI BgD Transactions on Advanced Research, 2007, pp. 21.

[23] Korolija, N., Djukic, T., Milutinovic, V. and Filipovic, N., “Accelerating Lattice-Boltzman Method Using Maxeler DataFlow Approach,” The IPSI BgD Transactions on Internet Research, 2013, pp. 34.

[24] Kos, A., Ranković, V. and Tomažič, S., “Sorting networks on Maxeler DataFlow supercomputing systems,” In Advances in computers, Vol. 96, pp. 139-186, Elsevier, 2015.

[25] S. Weston, J.T. Marin, J. Spooner, O. Pell, O. Mencer, “Accelerating the computation of portfolios of tranched credit derivatives,” IEEE Workshop on High Performance Computational Finance (WHPCF), November, 2010, pp. 1-8.

[26] Trifunovic, N., Milutinovic, V., Korolija, N. and Gaydadjiev, G., “An AppGallery for DataFlow computing,” Journal of Big Data, 3(1), pp.4, 2016.

[27] Trifunovic, N., Perovic, B., Trifunovic, P., Babovic, Z. and Hurson, A.R., “A novel infrastructure for synergistic DataFlow research, development, education, and deployment: the Maxeler AppGallery project,” In Advances in Computers, Vol. 106, pp. 167-213), Elsevier, 2017.

Reviewers:

Australia
Abramov, Vyacheslav; Monash University
Begg, Rezaul; Victoria University
Bem, Derek; University of Western Sydney
Betts, Christopher; Pegacat Computing Pty. Ltd.
Buyya, Rajkumar; The University of Melbourne
Chapman, Judith; Australian University Limited
Chen, Yi-Ping Phoebe; Deakin University
Hammond, Mark; Flinders University
Henman, Paul; University of Queensland
Palmisano, Stephen; University of Wollongong
Ristic, Branko; Science and Technology Organisation
Sajjanhar, Atul; Deakin University
Sidhu, Amandeep; University of Technology, Sydney
Sudweeks, Fay; Murdoch University
Austria
Derntl, Michael; University of Vienna
Hug, Theo; University of Innsbruck
Loidl, Susanne; Johannes Kepler University Linz
Stockinger, Heinz; University of Vienna
Sutter, Matthias; University of Innsbruck
Brazil
Parracho, Annibal; Universidade Federal Fluminense
Traina, Agma; University of Sao Paulo
Traina, Caetano; University of Sao Paulo
Vicari, Rosa; Federal University of Rio Grande
Belgium
Huang, Ping; European Commission
Canada
Fung, Benjamin; Simon Fraser University
Grayson, Paul; York University
Gray, Bette; Alberta Education
Memmi, Daniel; UQAM
Neti, Sangeeta; University of Victoria
Nickull, Duane; Adobe Systems, Inc.
Ollivier-Gooch, Carl; The University of British Columbia
Paulin, Michele; Concordia University
Plaisent, Michel; University of Quebec
Reid, Keith; Ontario Ministry og Agriculture
Shewchenko, Nicholas; Biokinetics and Associates
Steffan, Gregory; University of Toronto
Vandenberghe, Christian; HEC Montreal
Czech Republic
Kala, Zdenek; Brno University of Technology
Korab, Vojtech; Brno University of technology
Lhotska, Lenka; Czech Technical University
Finland
Lahdelma, Risto; University of Turku
Salminen, Pekka; University of Jyvaskyla
France
Cardey, Sylviane; University of Franche-Comte
Klinger, Evelyne; LTCI – ENST, Paris
Roche, Christophe; University of Savoie
Valette, Robert; LAAS - CNRS
Germany
Accorsi, Rafael; University of Freiburg
Glatzer, Wolfgang; Goethe-University
Gradmann, Stefan; Universitat Hamburg
Groll, Andre; University of Siegen
Klamma, Ralf; RWTH Aachen University
Wurtz, Rolf P.; Ruhr-Universitat Bochum
India
Pareek, Deepak; Technology4Development
Scaria, Vinod; Institute of Integrative Biology
Shah, Mugdha; Mansukhlal Svayam
Ireland
Eisenberg, Jacob; University College Dublin
Israel
Feintuch, Uri; Hadassah-Hebrew University
Italy
Badia, Leonardo; IMT Institute for Advanced Studies
Berrittella, Maria; University of Palermo
Carpaneto, Enrico; Politecnico di Torino
Japan
Hattori, Yasunao; Shimane University
Livingston, Paisley; Linghan University
Srinivas, Hari; Global Development Research Center
Obayashi, Shigeru; Institute of Fluid Science, Tohoku University
Netherlands
Mills, Melinda C.; University of Groningen
Pires, Luís Ferreira; University of Twente
New Zealand
Anderson, Tim; Van Der Veer Institute
Portugal
Cardoso, Jorge; University of Madeira
Natividade, Eduardo; Polytechnic Institute of Coimbra
Oliveira, Eugenio; University of Porto
Singapore
Tan, Fock-Lai; Nanyang Technological University
South Korea
Kwon, Wook Hyun; Seoul National University
Spain
Barrera, Juan Pablo Soto; University of Castilla
Gonzalez, Evelio J.; University of La Laguna
Perez, Juan Mendez; Universidad de La Laguna
Royuela, Vicente; Universidad de Barcelona
Vizcaino, Aurora; University of Castilla-La Mancha
Vilarrasa, Clelia Colombo; Open University of Catalonia
Sweden
Johansson, Mats; Royal Institute of Technology
Switzerland
Niinimaki, Marko; Helsinki Institute of Physics
Pletka, Roman; AdNovum Informatik AG
Rizzotti, Sven; University of Basel
Specht, Matthias; University of Zurich
Taiwan
Lin, Hsiung Cheng; Chienkuo Technology University
Shyu, Yuh-Huei; Tamkang University
Sue, Chuan-Ching; National Cheng Kung University
United Kingdom
Ariwa, Ezendu; London Metropolitan University
Biggam, John; Glasgow Caledonian University
Coleman, Shirley; University of Newcastle
Conole, Grainne; University of Southampton
Dorfler, Viktor; Strathclyde University
Engelmann, Dirk; University of London
Eze, Emmanuel; University of Hull
Forrester, John; Stockholm Environment Institute
Jensen, Jens; STFC Rutherford Appleton Laboratory
Kolovos, Dimitrios S.; The University of York
McBurney, Peter; University of Liverpool
Vetta, Atam; Oxford Brookes University
WHYTE, William Stewart; University of Leeds
Xie, Changwen; Wicks and Wilson Limited
USA
Bach, Eric; University of Wisconsin
Bolzendahl, Catherine; University of California
Bussler, Christoph; Cisco Systems, Inc.
Charpentier, Michel; University of New Hampshire
Chong, Stephen; Cornell University
Collison, George; The Concord Consortium
DeWeaver, Eric; University of Wisconsin - Madison
Gans, Eric; University of California
Gill, Sam; San Francisco State University
Hunter, Lynette; University of California Davis
Iceland, John; University of Maryland
Kaplan, Samantha W.; University of Wisconsin
Langou, Julien; The University of Tennessee
Liu, Yuliang; Southern Illinois University Edwardsville
Lok, Benjamin; University of Florida
Minh, Chi Cao; Stanford University
Morrissey, Robert; The University of Chicago
Mui, Lik; Google, Inc
Rizzo, Albert ; University of Southern California
Rosenberg, Jonathan M. ; University of Maryland
Shaffer, Cliff ; Virginia Tech
Sherman, Elaine; Hofstra University
Snyder, David F.; Texas State University
Song, Zhe; University of Iowa
Wei, Chen; Intelligent Automation, Inc.
Yu, Zhiyi; University of California

Authors of papers are responsible for the contents and layout of their papers.

Welcome to IPSI BgD Conferences and Journals!

http://tar.ipsitransactions.org

http://www.ipsitransactions.org

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd
ISSN 1820 – 4511
The IPSI BGD Transactions
on Advanced Research
COBISS.SR - ID 119128844

ISSN 1820-4511
9 771820 451006