

# Evaluating Web Service QoE by Learning Logic Networks

Natalia Kushik<sup>1,2</sup>, Nina Yevtushenko<sup>1</sup>, Ana Cavalli<sup>2</sup>, Wissam Mallouli<sup>3</sup> and Jeevan Pokhrel<sup>2,3</sup>

<sup>1</sup>*Department of Information Technologies, Tomsk State University, Tomsk, Russia*

<sup>2</sup>*Department of Network Software, TELECOM SudParis, Evry, France*

<sup>3</sup>*Montimage, Paris, France*

**Keywords:** Web Service, Quality, QoE, Logic Network/Circuit.

**Abstract:** This paper is devoted to the problem of evaluating the quality of experience (QoE) for a given web service based on the values of service parameters (for instance, QoS indicators). Different self-learning algorithms can be used to reach this purpose. In this paper, we propose to use self-learning logic networks, called also circuits, for evaluating the QoE of web services, since modern software tools can efficiently deal with very large logic networks. As usual, for machine learning techniques, statistics are used to design the initial circuit that accepts service parameter values as inputs and produces the QoE value as an output. The circuit is self-adaptive, i.e., when a new end-user provides a feedback of the service satisfaction the circuit is resynthesized in order to behave properly (if needed). Such resynthesis (circuit learning) can be efficiently performed using a number of tools for logic synthesis and verification.

## 1 INTRODUCTION

As the number of web services increases quickly, more methods and tools are needed to evaluate their quality. Moreover, there exist a number of service repositories (Directory of Public SOAP Web Services; Online community for the Universal Description, Discovery, and Integration OASIS Standard). Thus, a question arises of choosing the best service among the set of services with the same functionality. In order to effectively perform such a service selection, a 'selection criteria' need to be formalized. As usual, such formalization is based on the service quality that is defined with respect to different metrics or indicators.

One of most common metrics to evaluate a web service quality is a QoS (Quality of Service). The QoS is usually represented as a set of attributes or parameters of this service (Khirman and Henriksen, 2002; Al-Masri and Mahmoud; Kim et al., 2008; Morais and Cavalli, 2012). Those parameters could be the service response delay, packet loss rate, reliability, etc. Such parameters are rather objective and the value of each parameter can be automatically measured relying on specific tools. Nevertheless, the main objective of every web service is the satisfaction of end-users and thus,

other subjective parameters need to be taken into account when estimating the QoE (Quality of Experience).

As the QoE is used to measure the end-user satisfaction, the problem of its evaluation is much more complicated than the QoS evaluating problem. In other words, in order to evaluate QoE it is necessary to 'guess' how much an end-user would like or dislike the web-service. Thus, the problem arises of modeling the user perception, that is nowadays one of most popular problems in the artificial intelligent area where self-adaptive models are used for this purpose. A self-adaptive model usually accepts service parameter values as inputs and returns the QoE value as an output. If a model behaves in a wrong way for some newly emerged input/output pairs the model can be trained by itself or by an external 'teacher' that could be a service provider. The most popular of these self-adaptive models are discussed in the section on related work.

In this paper, we are focusing on a novel self-adaptive model approach for evaluating and 'predicting' the QoE value of web services. In fact, we extend an approach based on decision tree learning (Mitchell, 1997) to logic network (circuit) learning. As a decision tree can be represented as a disjunctive form where each conjunction represents a path to the leaf, we are looking forward to a

scalable representation and manipulation of such Boolean formula. One of possible ways to effectively represent such formula is to use a logic circuit. Similar to a decision tree, the initial logic network is derived based on statistics that are gathered from experts, developers and those end-users who agreed to provide a feedback about the service quality. The circuit accepts the service parameter values encoded as Boolean vectors and outputs the Boolean vector that corresponds to the encoded QoE value. The circuit is a self-learning machine, i.e., when new statistical data appear the circuit is checked to have the corresponding behavior and if the behavior does not correspond to newly emerged data the circuit is resynthesized. Such resynthesis can be efficiently performed using various tools. In this paper, we use the ABC (Berkeley Logic Synthesis and Verification Group, ABC) tool for logic synthesis and verification to train the corresponding circuit in a scalable way. The main contribution of this paper is the proposal of a machine learning technique for automatic evaluating QoE of web services, in which a machine is represented by a logic circuit and a learning operator corresponds to the circuit resynthesis.

An iterative approach is known to be efficient for logic circuit optimization. Given a circuit under optimization, the circuit is incrementally improved at each optimization step. In this paper, we consider a special optimization criterion, namely, the circuit QoE 'prediction' ability, and illustrate that an iterative learning of the circuit under construction is promising.

The rest of the paper is organized as follows. Section 2 contains the preliminaries and basic concepts related to logic network paradigm, while a state of the art of using self-adaptive models for the QoE evaluation is given in Section 3. The approach on using logic networks for the QoE evaluation is presented in Section 4. Section 5 contains an algorithm for the corresponding logic network learning procedure and a brief discussion of a proposed approach. Section 6 contains preliminary experimental results while Section 7 concludes the paper.

## 2 PRELIMINARIES

A *web service* can be described as a software system designed to support interoperable machine-to-machine interaction over a network (Booth et al., 2004). The Quality of a web Service (QoS) is defined as a vector with components which are

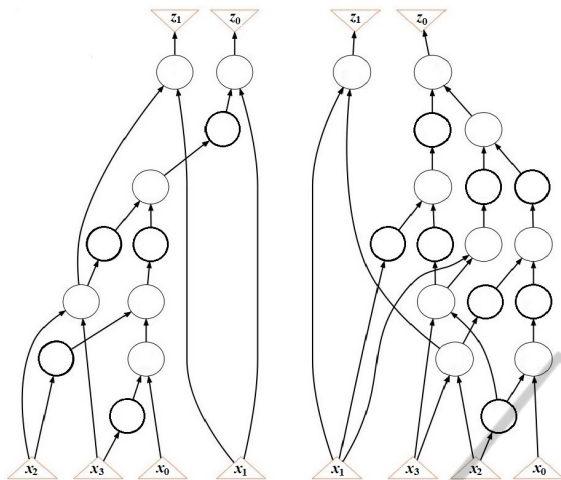
values of given attributes (parameters), such as time delay, packet loss rate, etc. The QoS is a metrics that represents some objective service parameter values that can be effectively measured, for example, based on the traffic analysis (Khirman and Henriksen, 2002). The QoE metrics is more complicated, since it represents a user satisfaction (Winckler et al., 2013) with a web service. Generally, the QoE also relies on different service parameters that are rather subjective than objective, those can be service design, comfort of proposed solutions, etc. In spite of the fact that the QoE is more difficult to evaluate, this metrics is more close to adequate description of the service quality, since the main purpose of each web service is to satisfy an end-user. In other words, the algorithm for the QoE evaluation has to be adapted to a human's brain in order to 'predict' what a user likes/dislikes. That is the reason why different self-adaptive models and algorithms are now used for this purpose. The advantage of a self-adaptive model is that it can be learnt or trained by a 'teacher' or by itself according to the feedback from people who use the service. As usual, an initial model/machine is derived based on some statistics that contain a number of user opinions about the service. Afterwards, the model can 'predict' a user satisfaction of the service for given values of service parameters. The more statistics are gathered the better is such 'prediction'. Moreover, as a model is self-adaptive, when new statistical data appear for which the model does not behave in an appropriate way, the model is adjusted to this new data and this process is the model training.

Given a self-adaptive model, different service parameters might be considered. One of most often case is to use QoS parameters, as those values can be measured automatically. On the other hand, for the QoE prediction the values of more subjective parameters, for instance, related to the business or trust quality should be considered as well.

In this paper, we are focusing on logic networks to be learnt for the QoE 'prediction' and we further briefly sketch the necessary definitions related to logic synthesis.

**Definition 1.** A *logic network* (circuit) consists of logic gates. Each logic gate has input (-s) and a single output. Outputs of some gates are connected to inputs of the others. The inputs of some gates that are not connected to any other gate output are claimed to be primary inputs while the outputs of some gates are claimed as primary outputs. In this paper, we consider combinational circuits, i.e., feedback-free circuits which have no latches.

Each gate implements a Boolean function. Most

Figure 1: a) A circuit  $S$ , b) A circuit  $S'$ .

common 2-input gates are AND/OR/XOR/NAND/NOR/XNOR that implement conjunction/disjunction/xor and their inversions. There are also 1-input gates such as NOT/BUFF that implement the inversion and the equality function, correspondingly.

As an example, consider a combinational circuit in Fig. 1a with a set  $X = \{x_0, x_1, x_2, x_3\}$  of inputs, a set  $Z = \{z_0, z_1\}$  of outputs and 11 AND and NOT gates. Hereafter, we assume that NOT nodes are taken in bold.

**Definition 2.** By definition, a logic circuit implements or represents a system of Boolean functions. A circuit accepts a Boolean vector as an input and produces the Boolean vector as an output according to the corresponding system of Boolean functions. Each logic circuit can be described by a Look-up-Table (LUT). A LUT contains a set of input/output pairs of a given circuit: if for the input  $\mathbf{i}$  the circuit produces an output  $\mathbf{o}$ , then the pair  $\mathbf{i}/\mathbf{o}$  is included into the LUT (see, for example, Table 2).

A LUT can be used as the specification when deriving a logic network that implements the system of Boolean functions, and there exist a number of methods how to synthesize a logic network that implements a given system of functions.

If this system is completely specified then 2-level network can be synthesized based on corresponding Disjunctive Normal Form (DNF) or Sum of Products (SoP) (McCluskey, 1965); otherwise, if the system of Boolean functions is only partially specified, then other methods are used for logic synthesis (see, for example, Kuehlmann, 2003). In this case, each input pattern where the circuit behavior is not specified is usually treated as a 'Don't\_Care' pattern, i.e., a circuit under design

can produce any output to this input. Correspondingly, the circuit behavior is specified for such 'Don't\_Care' patterns according to the designer's needs. It can be guarded by optimal criteria of a circuit under design, such as number of logic gates, time needed to produce an output, etc. In this paper, we are interested in deriving a circuit that models the QoE evaluation of web services, thus, one of optimal criteria could be the accuracy of the circuit prediction.

### 3 RELATED WORK

As mentioned above, the main priority of each web service is to satisfy an end-user, thus a service provider is trying to 'guess' what the end-user would like and what will be disliked. Therefore, the problem arises of modeling user perception that is nowadays one of most popular problems in the artificial intelligent area. A number of continuous and discrete models are developed for this purpose but nevertheless, no one can prove or disprove the accuracy of these models. Since a human mind is known to be non-deterministic, just now it is not possible to get an algorithm of the human behavior. That is the reason why researchers are focused on the study of self-adaptive models. Several self-adaptive models have been experimentally proven to behave similarly to a human being under certain conditions. In this paper, we do not discuss all the self-adaptive models that are developed in this artificial intelligent research area, instead we focus on those self-adaptive models that can be used to evaluate a user satisfaction of a given web service.

Almost all self-adapting models for evaluating the quality of web services rely on pre/post conditions that can be expressed as IF-THEN operators. In order to analyze a set of such conditions, machine learning algorithms can be of a big help. One of such algorithms is based on a Decision Tree (Mitchell, 1997; Pokhrel et al., 2013) that can be usually represented as a Boolean formula, i.e., a disjunction of conjunctions (cubes). Nevertheless, the size of a decision tree grows exponentially w.r.t. the number of parameters to be verified and thus, other more compact learning models that can be derived on the IF-THEN conditions should be considered.

One of popular models is a model of an Artificial Neural Network (see, for example, Ahmed et al., 2012; Al-Masri and Mahmoud Qusay, 2009) that accepts inputs and produces outputs depending on hidden nodes, their weights and connections. Neural

networks are widely used in various applications, such as modeling company marketing, oil flows, etc. Meanwhile, a number of methods for evaluating the QoE of web services are based on various kinds of neural networks. This network can be trained by an external ‘teacher’ or by itself to produce more precise results, i.e., it can be somehow adapted to the statistics being gathered from end-users of the service.

Another interesting model that can be used for the QoE evaluation is a fuzzy logic formula (Zadeh, 1965) where the operands do not have precise values but belong to some intervals which can be rather informal. In this case, a fuzzy logic formula represents a degree of a user satisfaction with the service. One may turn, for example, to papers (Lin et al., 2005; Torres et al., 2011) to see how fuzzy logic formulae can be used for evaluating QoE of web services.

Neural network and fuzzy logic formulae are experimentally proven to provide a reasonably accurate prediction of the user satisfaction. Nevertheless, when using these models for the QoE evaluation of web services as well as for the learning process a researcher faces a number of computational problems. The reason is that both models deal with continuous variables and thus, when evaluating the QoE floating-point numbers with a lot of decimal points are involved. As the discrete model, a decision tree does not have this drawback. On the other hand, a decision tree with the high prediction ability requires exponential memory and correspondingly needs a scalable representation. Such scalable representations for Boolean formulae include logic circuits, and in this paper, the Boolean formula corresponded to a decision tree is represented by a logic network that can ‘predict’ the QoE value even for patterns that are not specified in the provided statistics.

## 4 USING LOGIC NETWORKS FOR QOE PREDICTION

In this section, an approach for automatic evaluation/‘prediction’ of the QoE value for web services is proposed. Since the approach relies on logic circuit learning we first discuss how this (initial) circuit can be derived and then turn to a learning procedure itself.

Consider a web service  $W$  and a collection of parameters  $p_1, p_2, \dots, p_k$  that are used for the QoE evaluation. For the sake of scalability, we consider

each parameter  $p_i$  value as a nonnegative (unsigned) integer, bounded by the maximal value  $M_{p_i}$ ,  $p_i\_value \in \{0, \dots, M_{p_i} - 1\}$ . In order to evaluate the QoE of the service  $W$ , written  $QoE(W)$ , a logic circuit  $S$  is derived. Inputs of the circuit  $S$  correspond to service parameters  $p_1, p_2, \dots, p_k$  which values are encoded as Boolean vectors of length  $\lceil \log_2 M_{p_i} \rceil$ , where  $\lceil t \rceil$  denotes the minimal integer that is not less than  $t$ ; thus, the number of primary inputs of  $S$  equals  $\sum_{i=1}^k \lceil \log_2 M_{p_i} \rceil$ . The circuit calculates the  $QoE(W)$  value for given integers  $p_1\_value, p_2\_value, \dots, p_k\_value$  and  $QoE(W)$  that is bounded by the maximal value  $M_{QoE}$ ,  $QoE(W) \in \{0, \dots, M_{QoE} - 1\}$ , that is also represented as a Boolean vector. The length of a corresponding Boolean vector is  $\lceil \log_2 M_{QoE} \rceil$ , as  $\lceil \log_2 t \rceil$  bits are needed to represent an unsigned integer  $t$ . Therefore, the circuit  $S$  has  $\sum_{i=1}^k \lceil \log_2 M_{p_i} \rceil$  primary inputs and  $\lceil \log_2 M_{QoE} \rceil$  primary outputs.

### 4.1 Using Statistics for Circuit Learning

In order to derive the initial circuit  $S$ , we use statistics gathered from web service experts, from the automatic evaluation of service parameters and/or from end-users, who have experience of using a web service  $W$ . A running example is used for illustrating how such statistics can be utilized for logic synthesis.

**Example.** Let us consider an electronic service for booking a student dormitory. When entering the web site a student is asked for his personal information such as his/her name and family name and at the second step he/she indicates dates of the desired stay. The booking system checks the availability for the student to stay at the dormitory and outputs the result.

After using such electronic service, a student is asked whether he/she is willing to evaluate its quality and to provide his/her feedback to the service provider. If the student agrees then he/she is asked for evaluating the design quality of the system as well as the speed of it. Thus, two parameters  $p_1$  and  $p_2$  are considered where  $p_1$  represents the service design while  $p_2$  represents its speed. In this toy example, we assume that each parameter is evaluated as an integer between 0 and 3. The overall user satisfaction which value belongs to the same

interval is also asked for the student evaluation. To sum it up, the student is asked three questions:

1. *How do you find the design of the electronic service? Please evaluate the design by an integer between 0 and 3 (0 means that the service design is very bad and 3 means that the service design is excellent).*

2. *How do you find the speed of electronic service interaction? Please evaluate the speed by an integer between 0 and 3.*

3. *What is your overall User Satisfaction within this electronic service? Please evaluate your satisfaction by an integer between 0 and 3.*

A student can ignore the above questions but if a student replies then new statistical data are retrieved about the service quality.

Consider three users who have already left their scores. The user *A* evaluates the design quality and the speed of the service by the highest score and thus, the QoE evaluation is 3 (*excellent*). The user *B* gives the same evaluation for speed while he/she does not like the design (the score is 1) that does not influence his/her overall satisfaction that equals 3. The last user *C* that gives a feedback likes the design (the score is 3) while the speed is scored as 0, and the overall satisfaction is 2 (Table 1).

Table 1: Statistics gathered for dormitory booking electronic service.

User identifier	Design score	Speed score	User Satisfaction
<i>A</i>	3	3	3
<i>B</i>	1	3	3
<i>C</i>	3	0	2

The statistics described above are converted into input/output vectors for the circuit *S*. The circuit has two inputs for representing the value of the *design\_score*, two inputs for representing the value of the *speed\_score* and two outputs for representing the *QoE (dormitory\_booking)*. Based on statistics in Table 1 we derive a LUT (Look-up-Table) for a system of two output functions  $z_0, z_1$ , depending on four input variables denoted as  $x_0, x_1, x_2, x_3$ , i.e., we partially specify the behavior of the circuit *S* (Table 2). Table 2 describes the system of partially specified Boolean functions and thus, it can be used as the specification for designing a corresponding logic circuit.

Table 2: LUT for the statistics from Table 1.

$x_0$	$x_1$	$x_2$	$x_3$	$z_0$	$z_1$
1	1	1	1	1	1
0	1	1	1	1	1
1	1	0	0	1	0

## 4.2 Designing a Logic Circuit by ABC

We have derived the circuit *S* for the system of partially specified Boolean functions in Table 2 using the software tool ABC (Berkeley Logic Synthesis and Verification Group, ABC) that is a growing software system for synthesis and verification of binary synchronous sequential logic circuits. ABC provides scalable logic synthesis and optimization based on And-Inverter Graphs (AIGs) and other internal circuit representations. Moreover, ABC includes a number of commands for synthesis, resynthesis, optimization and verification of logic circuits as well as for printing the circuit statistics and parameters. The ABC can be easily downloaded from its official web site and run against a LUT for web service *W* to synthesize a corresponding circuit *S*. We have run the ABC tool against the LUT corresponded to Table 2. For this purpose, we have presented the set on input/output vectors in the PLA format. The resulting circuit *S* with 11 gates is shown in Fig. 1a. This circuit computes the QoE value of the dormitory booking service based on statistics in Table 1. If a service provider wants to ‘predict’ the overall user experience then the output of the circuit *S* can be automatically calculated for a corresponding input vector.

The logic circuit *S* that is derived based on the provided statistics, evaluates and ‘predicts’ the QoE value of web service *W* for any values of input parameters. The Don’t\_Care patterns can be used in a proper way when deriving the initial circuit for the QoE evaluation and below we present an algorithm for deriving the initial circuit *S*.

### Algorithm 1 for deriving an initial logic circuit to evaluate the QoE value

**Inputs:** Service parameters  $p_1, p_2, \dots, p_k$  with nonnegative (unsigned) integer values bounded by  $M_{p_1}, M_{p_2}, \dots, M_{p_k}$ ; maximal value of the QoE  $M_{QoE}$ ;

Statistics, i.e., feedbacks from users  $U_1, \dots, U_r$  represented as a list of patterns  $p_1\_value, p_2\_value, \dots, p_k\_value, UserSatisfaction\_value$ .

**Output:** a logic circuit *S*

1. Determine the number of primary inputs and primary outputs of *S*:

The number of primary inputs equals  $\sum_{i=1}^k \lceil \log_2 M_{p_i} \rceil$  while the number of primary outputs equals  $\lceil \log_2 M_{QoE} \rceil$ .

2. Derive a LUT

**2.1** For each user  $U_i$ ,  $i \in \{1, \dots, r\}$ , convert its statistic scores  $p_1\_value$ ,  $p_2\_value$ , ...,  $p_k\_value$ ,  $UserSatisfaction\_value$  into Boolean vectors and add the corresponding line to LUT.

**3.** Synthesize the circuit  $S$  from a partial system of Boolean functions and **Return**  $S$ .

□

Given an integer  $x$ , let  $\mathbf{B}(x)$  denote a corresponding Boolean vector for  $x$ . Due to steps 2 and 3 of the algorithm, the following statement holds.

**Proposition 1.** Given a service  $W$  and a statistic pattern  $p_1\_value$ ,  $p_2\_value$ , ...,  $p_k\_value$ ,  $UserSatisfaction\_value$ , a circuit  $S$  returned by Algorithm 1 produces the output  $\mathbf{B}(UserSatisfaction\_value)$  when the concatenation  $\mathbf{B}(p_1\_value) \cdot \mathbf{B}(p_2\_value) \cdot \dots \cdot \mathbf{B}(p_k\_value)$  is applied as an input vector.

Once a circuit  $S$  is derived, the circuit can be used for evaluating the QoE of the web service. As in this paper we are interested in self-adaptive models, we discuss how such circuit can be modified if the circuit behavior does not match new statistical data.

## 5 LOGIC NETWORK LEARNING PROCEDURE

A circuit  $S$  derived in the previous section has to be self-adaptive, i.e., when a new end-user agrees to leave his/her feedback about the service quality the circuit behavior has to be modeled under a corresponding input  $\mathbf{i}$  and if the result produced by the circuit differs significantly from the expected then the circuit has to be resynthesized. To evaluate the difference between the circuit output and the user satisfaction value we introduce some value  $\tau$  that represents a confidence interval, i.e., the  $QoE(W)$  produced by the circuit  $S$  has to belong to the interval  $[UserSatisfaction\_value - \tau, UserSatisfaction\_value + \tau]$ . This fact can be considered as an optimization step when constructing a logic network for evaluating/'predicting' QoE of web services. Given a new pattern  $\mathbf{i}/\mathbf{o}$  where  $\mathbf{o}$  represents the  $UserSatisfaction\_value$  and  $QoE(W)$  is an integer that corresponds to the  $S$  output for the input  $\mathbf{i}$ , if  $|QoE(W) - UserSatisfaction\_value| \leq \tau$  then the circuit  $S$  is not resynthesized.

The confidence interval  $[UserSatisfaction\_value - \tau, UserSatisfaction\_value + \tau]$  specifies the permissible distance between circuit behavior and

new statistical data. If the  $QoE(W)$  value produced by the circuit  $S$  does not belong to this interval several cases are possible:

1) The input part  $\mathbf{i}$  of a new statistical pattern  $\mathbf{i}/\mathbf{o}$  where  $\mathbf{o}$  is a Boolean vector for the expected  $UserSatisfaction\_value$ , was not in the initial LUT (not described in the previous statistics); and thus, the circuit  $S$  was not correctly synthesized for a Don't\_Care input  $\mathbf{i}$ . In this case, the new input pattern  $\mathbf{i}/\mathbf{o}$  is added to the LUT for the circuit and a circuit is resynthesized.

2) The output of the circuit  $S$  for the input  $\mathbf{i}$  was defined in the initial LUT but the output is not within the confidence interval. If this mismatching is caused by an expert error that provides initial statistics, we consider an end-user as an expert and resynthesize the circuit w.r.t. the new output for the input  $\mathbf{i}$ .

3) The output of the circuit  $S$  for the input  $\mathbf{i}$  is not within the confidence interval and this mismatching is due to different end-user opinions. In other words, for the same service parameter values, two different users specify the user satisfaction values with the difference greater than the constant  $\tau$ . In this case, a problem arises of solving a conflict of user opinions. In this paper, we do not go deep into solving this problem, since the probability of such situation is very low when the number of service parameters is large enough. Thus, we again rely on the last end-user opinion and resynthesize the circuit in a corresponding way.

**Algorithm 2 for learning the logic circuit that evaluates/ 'predicts' the QoE value for web service**

**Inputs:** QoE parameters  $p_1, p_2, \dots, p_k$  with nonnegative values bounded by  $M_{p_1}, M_{p_2}, \dots, M_{p_k}$ ; maximal value of the QoE  $M_{QoE}$ ;

The circuit  $S$  that evaluates the QoE value for web service  $W$ ;

A new user feedback  $p_1\_value, p_2\_value, \dots, p_k\_value, UserSatisfaction\_value$ ;

Maximal difference  $\tau$  for corresponding confidence interval.

**Output:** a modified logic circuit  $S$

**1.** Integers  $p_1\_value, p_2\_value, \dots, p_k\_value, UserSatisfaction\_value$  are converted into Boolean vectors  $\mathbf{v\_p1}, \mathbf{v\_p2}, \dots, \mathbf{v\_pk}, \mathbf{v\_us}$ .

**2.** The output  $QoE(W)$  of the circuit  $S$  is computed for the input  $\mathbf{v\_p1}, \mathbf{v\_p2}, \dots, \mathbf{v\_pk}$ .

**3.** If  $|QoE(W) - UserSatisfaction\_value| > \tau$  then

**3.1** If the line  $\mathbf{v\_p1}, \mathbf{v\_p2}, \dots, \mathbf{v\_pk}$  is specified as input in the LUT, then change the corresponding output into  $\mathbf{v\_us}$ ,

*Otherwise*

Add the new line  $v_{p1}, v_{p2}, \dots, v_{pk}, v_{us}$  to the LUT.

**3.2** Synthesize the new circuit  $S'$ ; assign  $S = S'$  and **Return**  $S$ . □

Similar to Proposition 1, the following statement holds.

**Proposition 2.** Given a service  $W$ , a statistic pattern  $p_1\_value, p_2\_value, \dots, p_k\_value, UserSatisfaction\_value$  and the length  $\tau$  of a confidence interval, a circuit  $S$  returned by Algorithm 2 produces the output  $\mathbf{B}(x)$ , such that  $|x - UserSatisfaction\_value| \leq \tau$  when the concatenation  $\mathbf{B}(p_1\_value) \cdot \mathbf{B}(p_2\_value) \cdot \dots \cdot \mathbf{B}(p_k\_value)$  is applied as an input vector.

In the running **example**, consider another user  $D$  that agrees to provide a feedback about using the electronic service of the dormitory booking system, and an output of the circuit  $S$  is computed according to his/her feedback. If  $S$  provides the output that does not belong to the user  $D$  confidence interval with  $\tau = 1$ , then the circuit should be resynthesized. Suppose the user  $D$  has scored the service design by 2 and its speed by 1 while the overall user satisfaction he/she has evaluated as 2, thus the QoE value is 2. These scores converted into Boolean vectors are equal to (1001)/(10). The circuit  $S$  in Fig. 1a outputs (00) which corresponds to 0 score when an input (1001) is applied, thus it has to be resynthesized taking into account this new input/output pair. We have performed such resynthesis using ABC and obtained another circuit  $S'$  with 17 gates (Fig. 1b).

As soon as necessary statistics is gathered, Algorithm 1 can be applied to synthesize the initial circuit  $S$ . In this case, the initial LUT contains  $r$  lines where  $r$  is number of feedback patterns. As it is well known, the problem of gathering such statistics as well as the problem of possible future circuit resynthesis are hard computational problems. Thus, we need appropriate heuristics in order to minimize the number of 'hard' operations. One of possible solutions could be to specify some Don't\_Care patterns before applying Algorithm 1 according to the value  $\tau$  related to the output confidence interval. In other words, a confidence interval can be considered for inputs as well, and all patterns that are close enough (with respect to this input confidence interval) can correspond to the same output. The latter means that for each input  $v_{p1}, v_{p2}, \dots, v_{pk}$ , a set of 'close' inputs can be determined and every input of this set is added to the LUT with the same  $v_{us}$  output. The procedure can

be simple enough if such confidence interval is represented by a corresponding conjunction (cube) but this issue needs more research. Additional experimental research is also needed to evaluate the optimal length of confidence intervals for different services, since the length of confidence intervals significantly influences both, the accuracy of predictions and the computational complexity contradicting each other. The bigger is the  $\tau$  value the less resynthesis steps are needed, along with reducing the accuracy of circuit predictions. Studying optimal values for the length of input and output confidence intervals remains one of challenging topics for future work.

## 6 EXPERIMENTAL RESULTS

We applied the logic circuit learning approach for evaluating the quality of media services using the statistics gathered by an innovative French company Montimage (more details about the company at [montimage.com](http://montimage.com)). In order to collect the statistical data for the QoE estimation of media services, Montimage researchers have conducted a subjective test with 25 end-user participants. This test involves a video service QoE evaluation based on network perturbations. Different video clips have been presented to the participants and according to the perceived video impairment, participants have evaluated each video clips quality as an integer from 1 to 5. The test had twenty seconds videos of different types (sports, movie, animation, and interview). To perform experiments, video clips have been streamed from a server to a client and corresponding perturbation has been added into an emulated network. *Packet loss* and *jitter* have been selected for perturbation where *jitter* corresponds to the variation in video packet delays and *packet loss* is the percentage of packets being lost during video transmitting. In the subjective test, video clips were randomly provided for participants which had five minutes break after watching 20 video clips. Given statistics gathered as described above, we have randomly extracted a sample of 17 user statistical data (Low number of users since we are only dealing a proof of concept experimentation). In this sample, the *jitter* ranges between 2 ms and 20 ms while the *packet loss* ranges from 0,02% to 0,25%. The latter means, five circuit inputs can be used to represent each parameter scores. To represent the QoE value, the corresponding circuit under construction has three outputs. We also mention that during the experiment we considered the output confidence

difference  $\tau = 1$ . Moreover, in this sample we also consider the corresponding confidence interval for each input with the same  $\tau = 1$  difference.

The current sample has been divided into three parts. At the first step, we have used 13 patterns to derive the initial circuit. Each pattern has been converted into a Boolean vector of length 13. We have run the ABC system against the corresponding system of partial Boolean functions represented in the PLA format. The obtained circuit  $S$  contains 121 gates while the maximal length of the path from primary inputs to primary outputs equals 22.

Following the given statistics, we have randomly chosen two more input patterns and simulated the behavior of the circuit  $S$  for new patterns. The simulation results coincide with the given statistical data for the user satisfaction with respect to the output confidence interval. Therefore, the circuit was not resynthesized.

Furthermore, we have added two additional patterns to the corresponding PLA file. We have simulated the circuit behavior on corresponding inputs and for one input the difference between expected and produced outputs has reached two. Correspondingly, the circuit has been resynthesized according to the new statistical data. Finally we have obtained the combinational circuit with 154 AIG nodes and with the maximal length from primary inputs to primary outputs being equal 29.

Preliminary experimental results show that updating the circuit with respect to new statistical data does not significantly increase the length of the path from primary inputs to primary outputs that can influence the speed of simulation procedure. In fact, given statistical data for the media service, the ABC system has produced the logic synthesis results almost immediately. This allows to claim that our proposed approach is promising, as the simulation procedure for the QoE calculation can be efficiently performed.

Further experimental research is needed to estimate the efficiency of this approach for different types of web services and their parameters.

## 7 CONCLUSIONS

In this paper, we have discussed different self-adaptive models for automatic QoE evaluation/‘prediction’ of web services. Extending the decision tree approach we have proposed a scalable approach for learning corresponding Boolean formulae using logic circuits for their representation. Two algorithms are proposed in the

paper, namely an algorithm for deriving an initial logic network based on statistics being gathered, and an algorithm for logic network learning based on new statistical data. As the future work, we plan to perform more experiments with different web services.

## ACKNOWLEDGEMENTS

The work is partially supported by ANR PIMI project (France) and by RFBR grant № 14-08-31640 мол\_a (Russia).

## REFERENCES

- Ahmed, S., Begum, M., Hasan Siddiqui, F., Abul Kashem, M., 2012. Dynamic Web Service Discovery Model Based on Artificial Neural Network with QoS Support. *International Journal of Scientific & Engineering Research Volume 3, Issue 3, pp. 1-7.*
- Al-Masri, E., Mahmoud, Q. H. Quality of web services dataset. *url: http://www.uoguelph.ca/~qmahmoud/qws/.*
- Al-Masri, E., Mahmoud Qusay, H., 2009. Discovering the Best Web Service: A Neural Network-based Solution. *SMC 2009, pp. 4250-4255.*
- Berkeley Logic Synthesis and Verification Group, ABC: A System for Sequential Synthesis and Verification, *url: http://www.eecs.berkeley.edu/~alanmi/abc/.*
- Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D., 2004. Web services architecture. *W3C Working Group Note, W3C Technical Reports and Publications, url: http://www.w3.org/TR/ws-arch/.*
- Directory of Public SOAP Web Services, *url: http://www.service-repository.com.*
- Khirman, S., Henriksen, P., 2002. Relationship between Quality-of-Service and Quality-of-Experience for public Internet service. In *Proc. of PAM 2002.*
- Kim, H. J., Lee, D. H., Lee, J. M., Lee, K. H., Lyu, W., Choi S.G., 2008. The QoE evaluation method through the QoS-QoE correlation model. In *Proc. of NCM 08, vol. 2, pp. 719-725.*
- Kuehlmann, A. (ed.), 2003. *The Best of ICCAD. 20 Years of Excellence in Computer-Aided Design.* Kluwer Academic Publishers.
- Lin, M., Xie, J., Guo, H., Wang, H., 2005. Solving QoS-driven web service dynamic composition as fuzzy constraint satisfaction. *EEE 2005, pp. 9-14.*
- McCluskey, E. J., 1965. *Introduction to the theory of switching circuits.* McGraw-Hill Book Company, NY.
- Mitchell, T.M., 1997. *Machine learning.* McGraw Hill series in computer science, McGraw-Hill.
- Morais, A., Cavalli, A., 2012. Deliverable D2.1 – State of the art of SQM/CEM technology, tools, and



standartization. *IPNQSYS European project*, url:  
<http://projects.celtic-initiative.org/ipnqsis/IPNQSYS-D21.pdf>.

Online community for the Universal Description, Discovery, and Integration OASIS Standard, url:  
<http://uddi.xml.org>.

Pokhrel, J., Mallouli, W., Montes de Oca, E., 2013. QoS Prediction and Self-Learning Mechanisms. *Technical report on the PIMI Project*.

Torres, R., Astudillo, H., Salas, R., 2011. Self-Adaptive Fuzzy QoS-Driven Web Service Discovery. In *IEEE SCC 2011*, pp. 64-71.

Winckler, M. A., Bach, C., Bernhaupt, R., 2013. Identifying user experience dimensions for mobile incident reporting in urban contexts. *IEEE Transactions on Communications*, vol. 56, no. 2, pp. 40-82.

Zadeh, L. A., 1965. Fuzzy sets. *Information and Control*, 8 (3), pp. 338-353.

