Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 4947–4975.
DOI: 10.3934/mbe.2019250
Received: 25 January 2019
Accepted: 24 March 2019
Published: 30 May 2019

Research article

Modeling and verification of an intelligent tutoring system based on Petri net theory

Yu-Ying Wang[1], Ah-Fur Lai[2], Rong-Kuan Shen[3], Cheng-Ying Yang[2], Victor R.L. Shen[4,5,]* and Ya-Hsuan Chu[4]

[1] Department of Applied Foreign Languages, Jinwen University of Science and Technology, 99, Anzhong Rd., Xindian Dist., New Taipei City 23154, Taiwan

[2] Department of Computer Science, University of Taipei, 1, Ai-Guo West Road, Taipei City, 10048 Taiwan

[3] Department of Japanese Language and Literature, Shih Hsin University, #1 Lane17, Sec.1, Mu-Cha Rd., Taipei City, Taiwan

[4] Department of Computer Science and Information Engineering, National Taipei University, 151, University Rd., Sanxia District, New Taipei City, 237 Taiwan

[5] Department of Information Management, Chaoyang University of Technology, 168, Jifeng E. Rd., Wufeng District, Taichung City 413, Taiwan

* **Correspondence:** Email: rlshen@mail.ntpu.edu.tw; Tel: +886282598689; Fax: +886282598589.

Abstract: According to the educational regulations in Taiwan, students are required to learn English when they are at the first grade of elementary school. However, not all the students have an appropriate environment to practice English, especially, for those students whose school is not located in the city. Thus, their English abilities in speaking, reading, and listening are poor. An intelligent tutoring system is used to help the students improve their English capabilities. This paper aims to provide a convenient tutoring environment, where teachers and students do not need to prepare a lot of teaching aids. They can teach and learn English whenever in the environment. Also, it proposes a method to verify the intelligent tutoring system using Petri nets. We have built the intelligent tutoring system based on Augmented Reality (AR), Text-to-Speech (TTS), and Speech Recognition (SR). This intelligent tutoring system is divided into two parts: one for teachers and the other for students. The experimental results have indicated that using Petri nets can help users verify the intelligent tutoring system for better learning performance and operate it correctly.

Keywords: intelligent tutoring system; augmented reality; text-to-speech; speech recognition; petri net

1. Introduction

English is a common international language. Thus, most of the countries, including Taiwan, have English courses in their formal educational institutions, such as elementary school, junior high school, senior high school and university. However, most of the students in Taiwan do not have a good environment to learn English, especially, for the students whose school is not located in the city. They do not have enough qualified teachers and financial aids to afford the tutorial classes. For that reason, we have built a tutoring system that provides a convenient environment to learn English without any real teaching aids. So, teachers and students do not need to spend much money on making the teaching aids. The important thing is the students' perspective. Some students think that English is not important because they will not use it any more in the future. However, English is required by students to pass the school or university entrance examination. It is the proof that English is really important.

Currently, in many tutoring systems, some distinguished platforms, such as Moodle, Sakai and D2L, are categorized as the ones of open source. According to SCORM (Sharable Content Object Reference Model) [1], teaching materials can be input and output on these platforms. However, these platforms are too complicated to be operated by school children.

Verification is an important issue in the tutoring system development. This paper aims to propose a novel method to verify the tutoring system using Petri nets. A novel environment, to change the traditional English learning method, based on Augmented Reality (AR), Text-to-Speech (TTS), and Speech Recognition (SR) was also developed. We change the traditional flash cards to AR. Since AR has been applied to different research areas and proven to be effective, more and more researchers use AR technology as an innovative tutoring method [2–5]. When using TTS and SR, students can know the correct pronunciation and present much courage to speak. It will enhance the students' interests in learning English. Furthermore, WoPeD (Workflow Petri net Designer) was applied to prove whether this tutoring system was working well or not.

We perform the system verification because it is susceptible to errors. This is also the important case with any engineering activity. The studies in human reliability have shown that people were trained to perform a specific operation making around 1–3 errors per hour in the best case of scenarios. In any activity, or the resulting outcome of an activity, the search for potential errors should not be neglected, regardless of whether they will happen or not. The consequences of errors could cause extremely significant failures or threats for teaching and learning [6].

The rest of the paper is organized as follows: The literature review of this tutoring system is discussed in Section 2. Section 3 describes the verification method of a tutoring system by using Petri nets. The simulation results, the real operation of a tutoring system, the comparison with other existing systems, learning results, and students' feedback are presented in Section 4. Finally, conclusion and future work are remarked in Section 5.

2. Literature review

In this section, we first introduce the concepts of Petri net theory. Then the Text-to-Speech (TTS) is presented to let the tutoring system speak. The Speech Recognition (SR) is presented to let the tutoring system understand users' saying. And the Augmented Reality (AR) is introduced to let users see the virtual objects superimposed upon or composited with the real world.

2.1. Petri net

Petri net theory was proposed by Dr. Carl Adam Petri in 1962. Petri net is defined as a five-tuple $PN = (P, T, F, W, M_0)$.

$P = \{p_1, p_2, ..., p_n\}$, $n > 0$, a finite set of places.
$T = \{t_1, t_2, ..., t_m\}$, $m > 0$, a finite set of transitions.

$F \subseteq (P \times T) \cup (T \times P)$, a finite set of arcs (i.e. flow relation).

W: $F \rightarrow \{1,2,3,\ldots\}$, a weight function.
M_0: $P \rightarrow \{0,1,2,\ldots\}$, an initial marking (or state).

$P \cap T = \emptyset$, $P \cup T \neq \emptyset$.

An ordinary Petri net is a four-variable network, $PN = (P, T, F, M_0)$. F denotes an arc that connects the transition to the place or the place to the transition. Petri net is a directed, weighted, and bipartite graph, denoted as $G = (V, E)$. The place and the transition are denoted as the nodes (V). A transition is enabled to fire if it meets the conditions such as weights, where a token transits from its input place i (p_i) to its output place j (p_j). It implies that the state of a system has been changed [7,8].

An arc is denoted as E, where $G = (V, E)$ denotes places, transitions and arcs. An arc is used to describe the correlations between places and transitions. It runs in a uni-direction and shows the direction of the token's transition. Weight functions, denoted as $w(p_i, t_j)$ and $w(t_j, p_j)$, can be added to the arcs, representing the condition it needs to transit from place i (p_i) to place j (p_j).

Marking (M) is an m-vector (k_0, k_1,…, k_m), where m denotes the number of places, and $k_\mathrm{i} >= 0$ denotes the number of tokens in place p_i. Initial marking, M_0 denotes a set of tokens in the system. When a set of tokens change, it indicates that the transition is fired. It means that the action of an event, i.e. the dynamic behavior, is described in the Petri net.

A fundamental Petri net is used to describe the change of the state in a discrete event dynamic system, while a complete Petri net model includes places, transitions, and arcs. The flow of tokens is denoted as the state change of the dynamic system [9–12].

In summary, the interpretation of places, transitions, and arcs are shown in Table 1.

Table 1. Interpretation of places, transitions, and arcs.

Places (V)	Input/output data
Transitions (V)	Computation
Arcs (E)	Place→Transition: consumes input data
	Transition→place: produces output data

2.1.1. Properties of Petri Nets

The properties of standard Petri net models can be divided into behavioral and structural properties [13]. The former refer to the initial marking functions of Petri nets, whereas the latter describe the structures of Petri nets. Furthermore, a Petri net model can be described using the following properties.

The behavioral properties include content reachability, boundedness, liveness, reversibility, coverability, persistence, synchronic distance, and fairness. The structural properties include structurally liveness, controllability, structural boundedness, conservativeness, repetitiveness, and consistency. Safeness, boundedness, deadlock and liveness are the common system behaviors which can be analyzed by using Petri net theory. The safeness means a place in a Petri net is safe if the number of tokens in that place never exceeds one. The boundedness in the Petri net means the number of tokens in a place is limited. The Petri net is *K*-bounded, which means the number of tokens never exceeds *K*. Liveness means the situations are without deadlocks.

In this paper, applying two major methods in behavioral properties and a novel method are used to analyze the tutoring system, which contains reachability tree, state equations, and WoPeD.

2.1.2. Reachability tree

The reachability tree represents the reachability set in a Petri net. The reachability tree is constructed from an initial state M_0. The algorithm begins with the initial state and enters an iterative procedure for a state expansion.

The procedure includes the following:

(1) Generate the next state.

(2) Confirm that the next state is new. If the next state equal to the present state is discarded, it can avoid duplication of the state.

(3) In case of a new state, insert a new state into the reachability tree. By using the reachability tree, we can confirm no deadlocks occurring in this system. An example of reachability tree is shown in Figure 1.

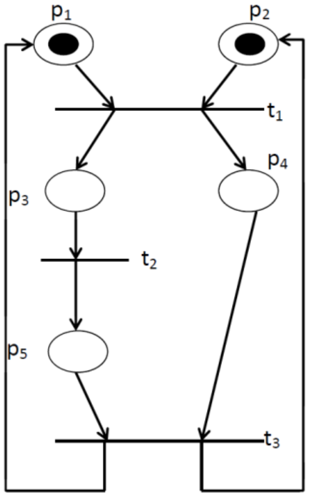


Figure 1. An example of reachability tree.

The firing sequence is denoted as t_1, t_2, $t_{3.}$

$$\begin{tikzcd} M_0 = [1\ \ 1\ \ 0\ \ 0\ \ 0] \arrow[d, "t_1"] \\ M_1 = [0\ \ 0\ \ 1\ \ 1\ \ 0] \arrow[d, "t_2"] \\ M_2 = [0\ \ 0\ \ 0\ \ 1\ \ 1] \arrow[uu, bend left=60, "t_3"] \end{tikzcd}$$

2.1.3. State equations

A state equation is based on a matrix in a Petri net. An alternative way to the (*P, T, I, O*) definition of Petri net is to define two matrices D^- and D^+, which denote the input and output functions, respectively. Each matrix contains *m* rows (one for each transition) by *n* columns (one for each place). We define $D^-[j, i] = \#(p_i, I(t_j))$ and $D^+[j, i] = \#(p_i, O(t_j))$. D^- denotes the inputs to the transition and D^+ denotes the outputs from the transition [14].

The matrix form of a Petri net $(P, T, D^{-,} D^{+)}$ is equivalent to the standard form we have used but allows the definitions to be recast in vector and matrix terms. $D = D^+ - D^-$ denotes the composite change matrix. Also, by using mathematical formula to calculate the state after a conversion, the formula is denoted as $\mu' = \mu + \chi{*}D.$

μ' denotes the next state (reachable from μ).

μ denotes the present state (in the beginning, it means initial marking).

χ denotes a transition vector.

An example of State Equations is shown in Figure 2.

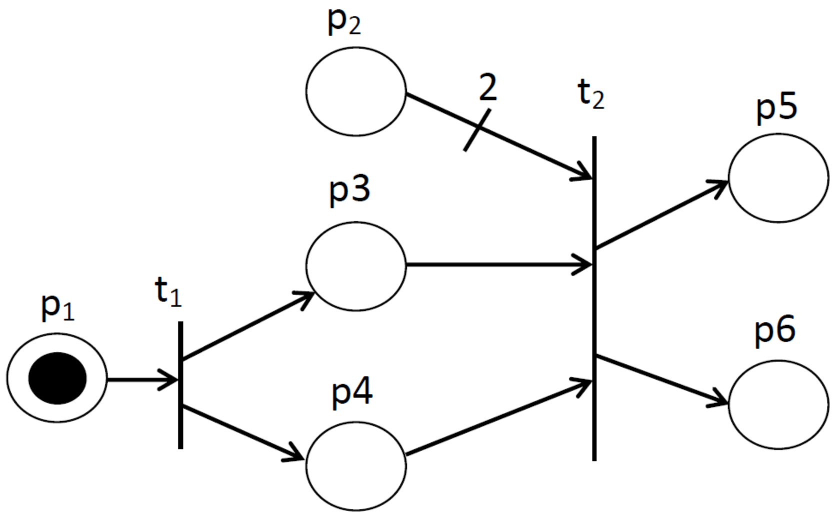


Figure 2. An example of the state equations.

$$D^{+} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad D^{-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$D = D^{+} - D^{-} = \begin{bmatrix} -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -2 & -1 & -1 & 1 & 1 \end{bmatrix}$$

$$\mu' = \mu + \chi * D$$

$$\chi = \mathrm{t}_1$$

$$\mu' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -2 & -1 & -1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\chi = \mathrm{t}_2$$

$$\mu' = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -2 & -1 & -1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -2 & -1 & -1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -2 & 0 & 0 & 1 & 1 \end{bmatrix}$$

2.1.4. WoPeD

WoPeD is used to perform the simulation. WoPeD is an open-source software system developed by the Cooperative State University Karlsruhe under the GNU Lesser General Public License (LGPL). WoPeD is used as a simulation tool. The main goal is to provide an easy-to-use software system for modeling, simulating and analyzing processes described by workflow nets [15] .The WoPeD interface is shown in Figure 3.

In WoPed, the semantical analysis is separated into two parts: one is Wizard and the other is Expert. Wizard part needs to check Workflow net property and Soundness. Expert part is separated into two sub-parts: one is Structural analysis and the other is Soundness. Structural analysis is divided into five parts: Net Statistics, Wrongly used operators, Free-choice violations, S-Components and Wellstructuredness. Soundness is divided into four parts: Workflow net property, Initial marking, Boundedness and Liveness.

The functions of WoPeD include the following:

(1) Process and Resource Editor
(2) PNML Compliance and Import/Export Formats
(3) Soundness Checker

(4) Interactive Simulator and Coverability Graph Visualization
(5) Quantitative Simulation and Capacity Planning
(6) Research Support Tool
(7) AProMoRe Repository Front-End

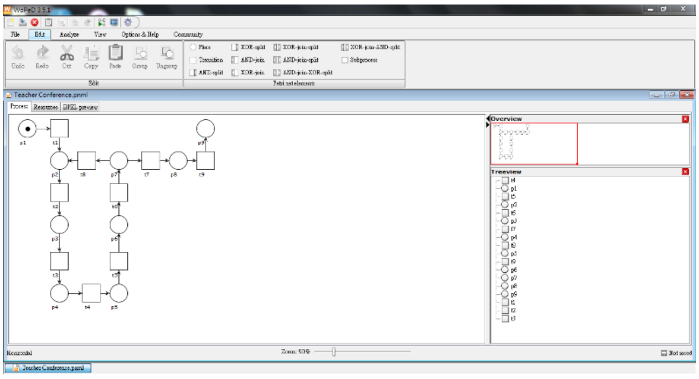


Figure 3. WoPeD interface.

2.2. Text-to-speech (TTS)

TTS is a computerized system, in which written texts are synthetically transformed into speech. This technology is often called "Speech Synthesis" [16]. The TTS system converts a normal language text into speech. However, other systems render symbolic linguistic representations like phonetic transcriptions into speech [17]. The users do not need to pre-record the voice, just key-in the words a computer system.

Recently, the TTS technology is more and more popular, used in daily life, like public broadcasting, automatically reported to the radio message of daily life, automatic earthquake alarm, emergency alert, and language listening tests or exercises, etc. Especially, the foreign language educational applications, in order to improve students' listening comprehension, by using TTS students can learn any language whenever they want and also have the standard pronunciation.

2.3. Speech recognition (SR)

The main purpose of SR in a computer is to realize the human voice. Some SR systems use "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in the accuracy increased. Systems that do not use training are called "speaker independent" [18]. Systems that use training are called "speaker dependent".

Using SR allows the users to control the computer via voice commands. The SR program can also dictate texts, so that the user can enter a text by voice. It contains a simple instruction to guide the user, and provides training functions, improving the accuracy of identification.

For now, speech recognition applications include voice dialing, voice navigation, indoor device control, voice document retrieval, simple dictation data entry, and so on. In the future, speech recognition, combined with other natural language processing technologies, such as machine translation and speech synthesis, can build more complex applications, such as voice-to-speech translation.

2.4. Augmented reality (AR)

AR allows users to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replaces it. Ideally, it would appear to the user that the virtual and real objects coexisted in the same space.

AR was divided into two definitions: one was defined by Ronald Azuma in 1997. AR is viewed as a system that has the following three characteristics: 1) Combining real objects and virtual objects, 2) Interactive in real time, 3) Registered in 3-D [19]; and the other was defined by Paul Milgram and Fumio Kishino in 1994. The concept of a "virtuality continuum" relates to the mixture of classes of objects presented in any particular display situation, as illustrated in Figure 4. The real environments are shown at the end of the continuum; and virtual environments are shown at the opposite end of the extremum. The former case, at the left, defines the environment solely consisting of real objects and includes for example what is observed via a conventional video display of a real-world scene. The latter case, at the right, defines the environment solely consisting of virtual objects [20].

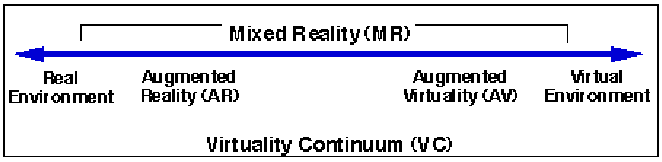


Figure 4. Simplified representation of a “virtuality continuum”.

3. The proposed system

This Section describes the flowcharts of the tutoring system and uses WoPeD to draw the Petri net models for verification.

3.1. Flowcharts of intelligent tutoring system

The tutoring system contains two parts: one part for teachers and the other for students. Figures 5 and 6 represent the flowcharts of the teacher system and the student system, respectively. The teacher system and the student system can communicate with each other.

In Figure 5, teachers can choose a picture from the file. The picture is shown at the center of the interface. The teacher can adjust the position where the picture appears, input the vocabulary or the short sentence in the *text box*, and save it into the system. Follow these steps until no more pictures are added.

In Figure 6, students can use the document that is provided by teachers. Click the *Start* button, the *Listen* button and listen to the vocabulary or the short sentences. If students cannot listen clearly, they can repeat it until they understand and speak out the vocabulary or the short sentences. Then, if the speech recognition detects that the vocabulary or the short sentence is correct, the vocabulary or the short sentence is displayed in the *message box* and the picture is displayed on the screen. Repeat these steps until all vocabularies or sentences are pronounced by the system.

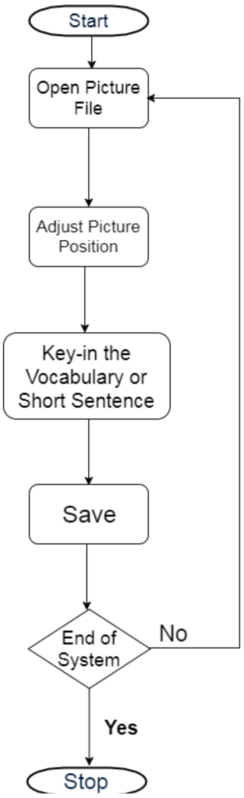


Figure 5. Flowchart of teacher system.

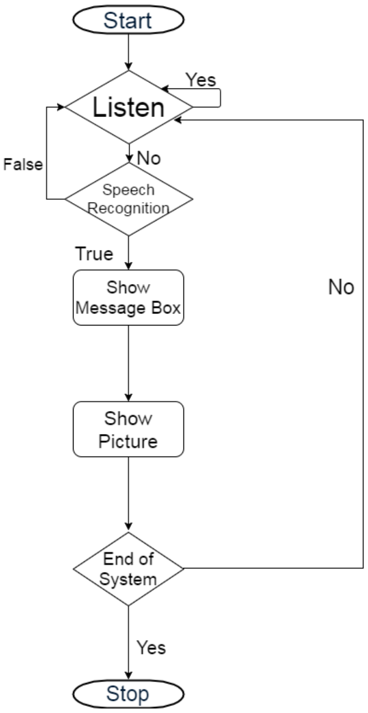


Figure 6. Flowchart of student system.

3.2. Tutoring system architecture on WoPeD

In this Sub-section, WoPeD is used to verify the workflow of the tutoring system.

We divide the tutoring system into two parts, including the WoPeD architectures of the teacher system and the student system.

3.2.1. Architecture of teacher system

The architecture of teacher system is clearly illustrated in Figure 7. p_7 denotes the end-of-system, where the teacher can choose whether to exit the system or not.

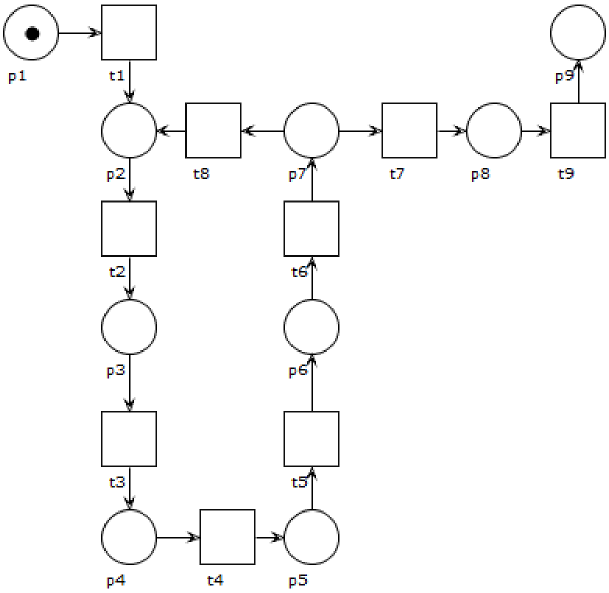


Figure 7. WoPeD for teacher system.

The teacher system with interpretation of places and transitions is shown in Tables 2 and 3, respectively.

Table 2. Interpretation of places in teacher system.

Place	Interpretation
p_1	Main
p_2	Tutoring System Available
p_3	Picture Available
p_4	Picture Position Adjusted
p_5	Vocabulary or Short Sentence Available
p_6	File Saved
p_7	End-of-System
p_8	Check Already
p_9	Stop

Table 3. Interpretation of transitions in teacher system.

Transition	Interpretation
t_1	Start
t_2	Open Picture File
t_3	Adjust Picture Position
t_4	Key-in the Vocabulary or Short Sentence
t_5	Save the Position, Picture and Vocabulary
t_6	Check End-of- System
t_7	End
t_8	No End
t_9	Choose the System

3.2.2. Architecture of student system

The student can choose to listen to the vocabulary or short sentences repeatedly. If it is not this case, the student has to pronounce the vocabulary or short sentences through speech recognition process. Figure 8 illustrates the tutoring system architecture for students.

Tables 4 and 5 describe the interpretation of places and transitions in the student system, respectively.

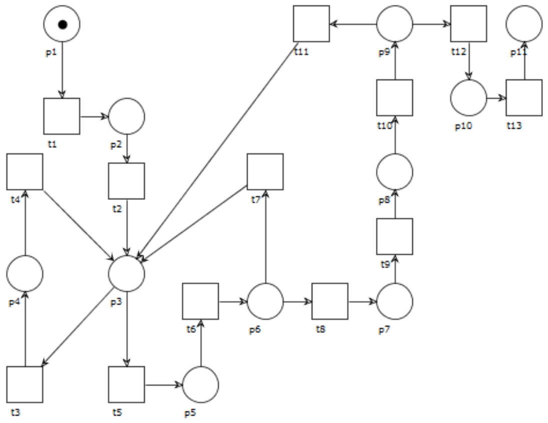


Figure 8. WoPeD for student system.

Table 4. Interpretation of places in student system.

Place	Interpretation
p_1	Main
p_2	Student System Available
p_3	Listening
p_4	*Listen* Button Clicked
p_5	*Speech Recognition* Button Clicked
p_6	Speech Recognition
p_7	The Message Box Shown

p_8	AR Available
p_9	End-of- System
p_{10}	Check Already
p_{11}	Stop

Table 5. Interpretation of transitions in student system.

Transition	Interpretation
t_1	Start
t_2	Click *Listen* Button
t_3	Choose *Listen* Button Again
t_4	Repeat Listening
t_5	Do not Choose *Listen* Button Again
t_6	Speak
t_7	Check Speech Recognition False
t_8	Check Speech Recognition True
t_9	Show the Picture
t_{10}	Check End-of-System
t_{11}	No End
t_{12}	End
t_{13}	Close the System

3.3. Intelligent tutoring system analysis using reachability tree

Applying reachability tree for verification, we need to find the initial marking. By firing transitions t_1, t_2, $t_{3,}$..., t_n , the reachability tree is generated accordingly.

3.3.1. Teacher system

The initial marking is denoted as $M_{0.}$

The firing sequence is denoted as $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_{9.}$

The marking changed sequence is shown as follows:

$$M_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_6 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix}$$

$$M_7 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$$

$$M_8 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$M_0 = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

3.3.2. Teacher System

The initial marking is denoted as $M_{0.}$

The firing sequence is denoted as t_1, t_2, $t_3//t_4$, t_5, t_6, t_7, t_8, t_9, t_{10}, $t_{11,}$ t_{12}, $t_{13.}$

The marking changed sequence is shown as follows:

$$M_0 = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$M_1 = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$M_2 = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$M_3 = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \quad M_4 = \begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$M_5 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$M_6 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$M_7 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

$$M_8 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix}$$

$$M_9 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$$

$$M_{10} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

3.4. Tutoring system analysis using state equations

By using formula $\mu' = \mu + \chi{*}D$ and $D=D^{+}\text{-}D^{-}$ to verify the system, we check whether this tutoring system has a deadlock or not.

3.4.1. Teacher system

In the first step, we calculate the D^{+} and D^{-}, and get the result of D.

$$D^{+}=\begin{bmatrix}0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&0\\0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1\end{bmatrix}\qquad D^{-}=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&0\end{bmatrix}$$

D=D^{+}-D^{-}:

$$D=\begin{bmatrix}0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&0\\0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1\end{bmatrix}-\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&0\end{bmatrix}$$

$$=\begin{bmatrix}-1&1&0&0&0&0&0&0&0\\0&-1&1&0&0&0&0&0&0\\0&0&-1&1&0&0&0&0&0\\0&0&0&-1&1&0&0&0&0\\0&0&0&0&-1&1&0&0&0\\0&0&0&0&0&-1&1&0&0\\0&0&0&0&0&0&-1&1&0\\0&1&0&0&0&0&-1&0&0\\0&0&0&0&0&0&0&-1&1\end{bmatrix}$$

By following the D=D^{+}-D^{-}, we get the result of D, and proceed to the next step by using formula μ' = μ + χ*D. We need to calculate the next state μ', so we denote the χ as t_1 =

$$\begin{bmatrix}1&0&0&0&0&0&0&0&0\end{bmatrix}.$$

Firing t_1, we get

$$\begin{bmatrix}1&0&0&0&0&0&0&0&0\end{bmatrix}+\begin{bmatrix}1&0&0&0&0&0&0&0&0\end{bmatrix}*\begin{bmatrix}-1&1&0&0&0&0&0&0&0\\0&-1&1&0&0&0&0&0&0\\0&0&-1&1&0&0&0&0&0\\0&0&0&-1&1&0&0&0&0\\0&0&0&0&-1&1&0&0&0\\0&0&0&0&0&-1&1&0&0\\0&0&0&0&0&0&-1&1&0\\0&1&0&0&0&0&-1&0&0\\0&0&0&0&0&0&0&-1&1\end{bmatrix}=\begin{bmatrix}0&1&0&0&0&0&0&0&0\end{bmatrix}$$

Firing t_1, the next state is μ'= $\begin{bmatrix}0&1&0&0&0&0&0&0&0\end{bmatrix}$. It represents that the next state is available to $p_{2.}$ When firing *t2, t3, t4, t5, t6, t7, t8, t9* sequentially, it is verified to be deadlock free.

3.4.2. Student system

Similarly, by following the operating procedure of the teacher system, we can find the D^+ and D^- immediately and find the D.

$$D^+=\begin{bmatrix}0&1&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0&1\end{bmatrix}\qquad D^-=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&1&0\end{bmatrix}$$

$D=D^+-D^-$:

$$D=\begin{bmatrix}0&1&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0&1\end{bmatrix}-\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&1&0\end{bmatrix}$$

$$=\begin{bmatrix}-1&1&0&0&0&0&0&0&0&0&0\\0&-1&1&0&0&0&0&0&0&0&0\\0&0&-1&1&0&0&0&0&0&0&0\\0&0&1&-1&0&0&0&0&0&0&0\\0&0&-1&0&1&0&0&0&0&0&0\\0&0&0&0&-1&1&0&0&0&0&0\\0&0&1&0&0&-1&0&0&0&0&0\\0&0&0&0&0&-1&1&0&0&0&0\\0&0&0&0&0&0&-1&1&0&0&0\\0&0&0&0&0&0&0&-1&1&0&0\\0&0&1&0&0&0&0&0&-1&0&0\\0&0&0&0&0&0&0&0&-1&1&0\\0&0&0&0&0&0&0&0&0&-1&1\end{bmatrix}$$

By following the $D=D^{+}-D^{-}$, we get the result of D. We proceed to the second step by using formula $\mu' = \mu + \chi * D$. Since we need to calculate the next state μ', we denote χ as $\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$.

Firing t_1, we get

$$
\begin{aligned}
&\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \\
&* \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix} \\
&= \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}
\end{aligned}
$$

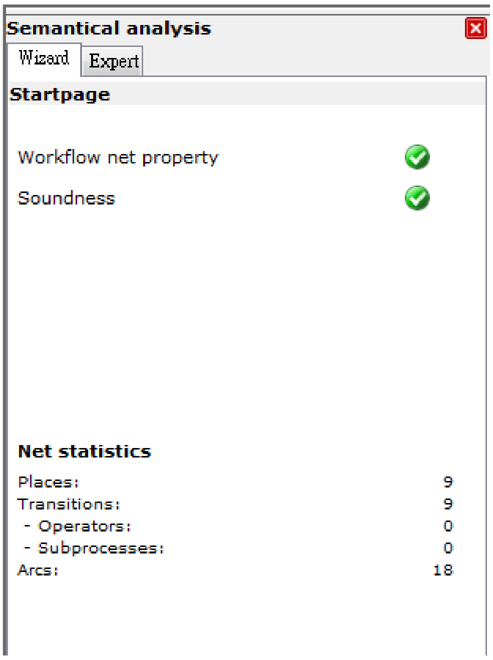
Firing t_1, the next state is $\mu' = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$. It represents that the next state is available to $p_{2.}$ When firing *t2*, *t3*, *t4*, *t5*, *t6*, *t7*, *t8*, *t9*, *t10*, *t11*, *t12*, *t13* sequentially, it is also verified to be deadlock free.

4. Experimental results

In this Section, we focus on the WoPeD simulation results. Also, we propose a deadlock example, introduce the tutoring system operating procedure, compare with other existing systems, and obtain the learning results and students' feedback.

4.1. Simulation results

In this sub-section, the simulation results of the proposed model are discussed. This system uses WoPeD based on the Petri net model. The simulation results of the teacher and the student systems are shown in Figures 9 and 10, respectively. Figure 9 shows the experimental results of the teacher system. This system contains 9 places (with one source place and one sink place), 9 transitions, and 18 arcs. The experimental results of the student system are shown in Figure 10. The student system contains 11 places (with one source place and one sink place), 13 transitions, and 26 arcs. The simulation results indicate that the proposed model can prevent the deadlock from occurring in the tutoring system.



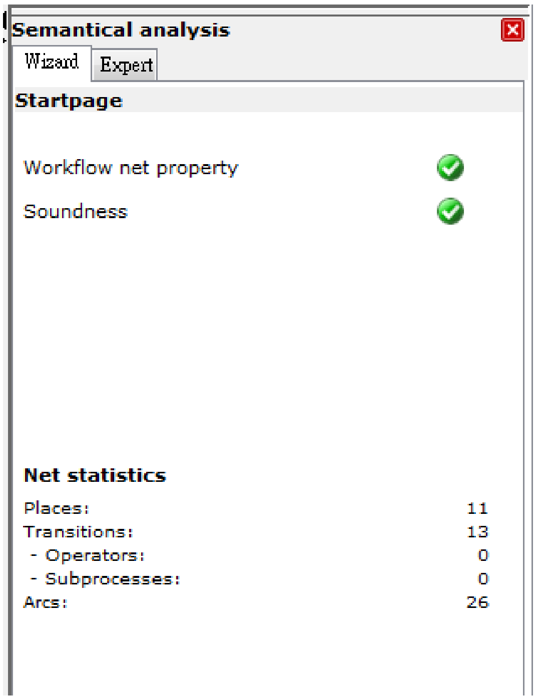
Workflow net property ✓

Soundness ✓

Net statistics

Places:	9
Transitions:	9
- Operators:	0
- Subprocesses:	0
Arcs:	18

Figure 9. Teacher semantical analysis.



Workflow net property ✓

Soundness ✓

Net statistics

Places:	11
Transitions:	13
- Operators:	0
- Subprocesses:	0
Arcs:	26

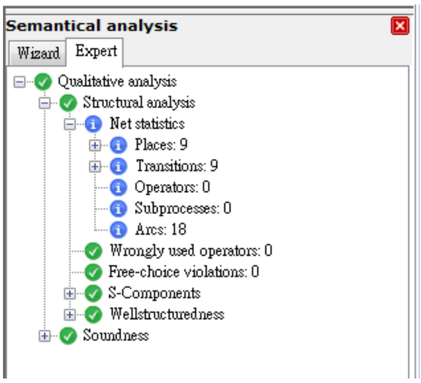
Figure 10. Student semantical analysis.

4.1.1. Teacher system

The semantical analysis contains two parts, namely, Structural analysis and Soundness analysis.

The first part is to perform the Structural analysis, which includes the Net statistics, Wrongly used operators, Free-choice violations, S-Components and Wellstructuredness. In Figure 11, Net statistics show 9 places (p_1, p_2,

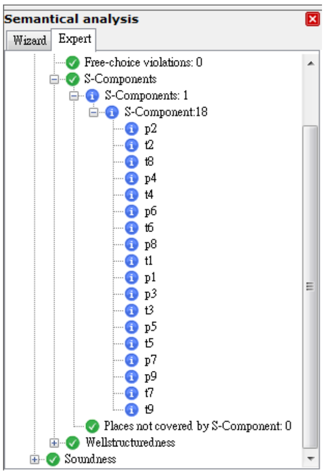
$p_{3,}, \ldots, p_{9.}$), 9 transitions (t_1, t_2, $t_{3,} \ldots, t_{9.}$), 0 operator, 0 subprocess, and 18 arcs.



```
⊟ ✓ Qualitative analysis
    ⊟ ✓ Structural analysis
        ⊟ ⓘ Net statistics
            ⊞ ⓘ Places: 9
            ⊞ ⓘ Transitions: 9
              ⓘ Operators: 0
              ⓘ Subprocesses: 0
              ⓘ Arcs: 18
          ✓ Wrongly used operators: 0
          ✓ Free-choice violations: 0
        ⊞ ✓ S-Components
        ⊞ ✓ Wellstructuredness
    ⊞ ✓ Soundness
```

Figure 11. Teacher semantical analysis for Net statistics.

The second step of checking procedure is Wrongly used operators. Following the analysis, there are no errors in this analysis. The third step is to check Free-choice violations. We do not find the violations in it, either. The fourth step shown in Figure 12 is to check S-Components. Following the analysis procedure, we have 1 set of S-Components and the S-Components shows 18, including p_1, p_2, $p_3 \ldots$, p_9, t_1, t_2, $t_{3,} \ldots$, t_9. It means that all P and T are in S-Components, and also means that no other Places are covered by S-Components.



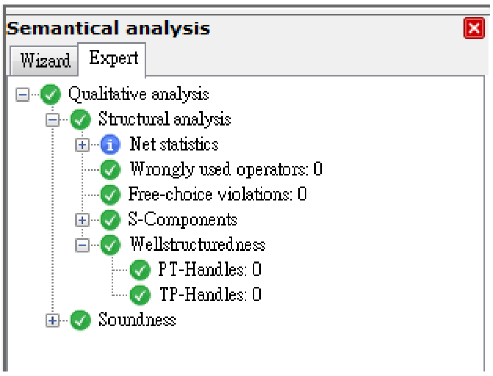
```
      ✓ Free-choice violations: 0
    ⊟ ✓ S-Components
        ⊟ ⓘ S-Components: 1
            ⊟ ⓘ S-Component:18
                ⓘ p2
                ⓘ t2
                ⓘ t8
                ⓘ p4
                ⓘ t4
                ⓘ p6
                ⓘ t6
                ⓘ p8
                ⓘ t1
                ⓘ p1
                ⓘ p3
                ⓘ t3
                ⓘ p5
                ⓘ t5
                ⓘ p7
                ⓘ p9
                ⓘ t7
                ⓘ t9
          ✓ Places not covered by S-Component: 0
    ⊞ ✓ Wellstructuredness
⊞ ✓ Soundness
```

Figure 12. Teacher semantical analysis for S-Component.

The fifth step shown in Figure 13 is to check Wellstructuredness, which includes PT-Handles and TP-Handles. Following the analysis, there are no any other errors in it.

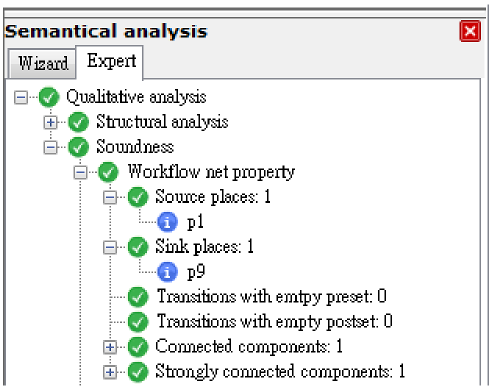
The second part is to perform the Soundness analysis which includes the Workflow net property, Initial marking, Boundedness, and Liveness. In the Workflow net property, the first step is to check the Source place, p_1; and in the second step the sink place is p_9, as shown in Figure 14. The third and four steps are transitions with empty preset and transitions with empty postset, which have no any errors, either. The fifth step is to check the Connected components. The Connected components contain the whole P and T, places (p_1, p_2, $p_{3,} \ldots$, p_9) and transitions (t_1, t_2, $t_{3,} \ldots, t_9$); totally, 18

Connected components. The sixth step is to check the Strongly connected components, and the result is the same as Connected components, as shown in Figures 15 and 16.



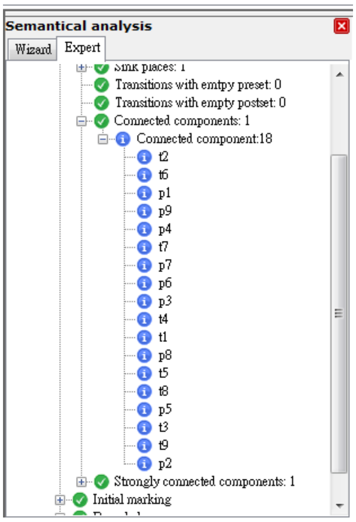
```
Semantical analysis
Wizard  Expert
⊟ ✓ Qualitative analysis
    ⊟ ✓ Structural analysis
        ⊞ i Net statistics
          ✓ Wrongly used operators: 0
          ✓ Free-choice violations: 0
        ⊞ ✓ S-Components
        ⊟ ✓ Wellstructuredness
              ✓ PT-Handles: 0
              ✓ TP-Handles: 0
    ⊞ ✓ Soundness
```

Figure 13. Teacher semantical analysis for Wellstructuredness.



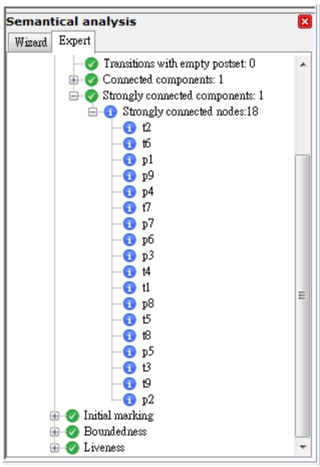
```
Semantical analysis
Wizard  Expert
⊟ ✓ Qualitative analysis
    ⊞ ✓ Structural analysis
    ⊟ ✓ Soundness
        ⊟ ✓ Workflow net property
            ⊟ ✓ Source places: 1
                  i p1
            ⊟ ✓ Sink places: 1
                  i p9
              ✓ Transitions with emtpy preset: 0
              ✓ Transitions with empty postset: 0
            ⊞ ✓ Connected components: 1
            ⊞ ✓ Strongly connected components: 1
```

Figure 14. Teacher semantical analysis for Source places and Sink places.



```
Semantical analysis
Wizard  Expert
            ⊞ ✓ Sink places: 1
              ✓ Transitions with emtpy preset: 0
              ✓ Transitions with empty postset: 0
            ⊟ ✓ Connected components: 1
                ⊟ i Connected component:18
                      i t2
                      i t6
                      i p1
                      i p9
                      i p4
                      i t7
                      i p7
                      i p6
                      i p3
                      i t4
                      i t1
                      i p8
                      i t5
                      i t8
                      i p5
                      i t3
                      i t9
                      i p2
            ⊞ ✓ Strongly connected components: 1
        ⊞ ✓ Initial marking
```

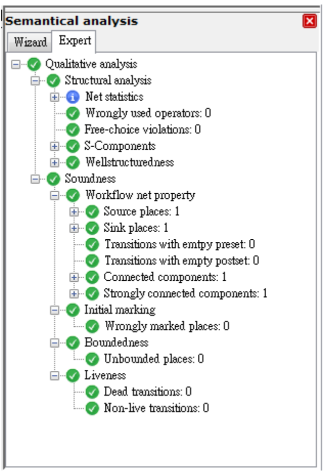
Figure 15. Teacher semantical analysis for Connected components.



```
      Transitions with empty postset: 0
    ⊞ Connected components: 1
    ⊟ Strongly connected components: 1
        ⊟ Strongly connected nodes:18
            t2
            t6
            p1
            p9
            p4
            t7
            p7
            p6
            p3
            t4
            t1
            p8
            t5
            t8
            p5
            t3
            t9
            p2
  ⊞ Initial marking
  ⊞ Boundedness
  ⊞ Liveness
```

Figure 16. Teacher semantical analysis for Strongly connected components.

In the initial marking, there are no other Wrongly marked places. In the Boundedness, there are no other Unbounded places. In Liveness, there is no any dead transition in it. Therefore, following this step to check the system, the proof without deadlocks in it is completed. The total results are shown in Figure 17.



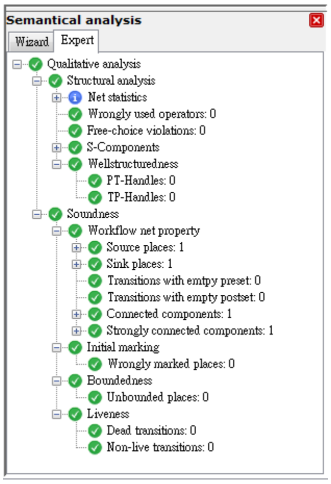
```
⊟ Qualitative analysis
    ⊟ Structural analysis
        ⊞ Net statistics
          Wrongly used operators: 0
          Free-choice violations: 0
        ⊞ S-Components
        ⊞ Wellstructuredness
    ⊟ Soundness
        ⊟ Workflow net property
            ⊞ Source places: 1
            ⊞ Sink places: 1
              Transitions with emtpy preset: 0
              Transitions with empty postset: 0
            ⊞ Connected components: 1
            ⊞ Strongly connected components: 1
        ⊟ Initial marking
              Wrongly marked places: 0
        ⊟ Boundedness
              Unbounded places: 0
        ⊟ Liveness
              Dead transitions: 0
              Non-live transitions: 0
```

Figure 17. Teacher system semantical analysis.

4.1.2. Student system

In Figure 8, based on the semantical results, the Net statistics have 11 Places, 13 Transitions and 26 Arcs, and have one S-Component. There are p_1, p_2, p_3,…, p_{11} and t_1, t_2, t_3,…, t_{13}, totally, 24 S-Components. In the Workflow net property, Source place is p_1, and Sink place is p_{11}. The connected components and strongly connected components are all ones. The Strongly connected components have 24 nodes, including p_1, p_2, p_3,…, p_{11} and t_1, t_2, t_3,…, t_{13}. The initial marking, boundedness and liveness show that there are no wrongly marked places, unbounded places, dead transitions and non-live transitions.

Thus, based on the same checking steps on the Teacher system, we can get the experimental results of the student system, which does not have deadlocks inside the system, either. The resulting semantical analysis is shown in Figure 18.



Semantical analysis

Wizard | Expert

- Qualitative analysis
  - Structural analysis
    - Net statistics
    - Wrongly used operators: 0
    - Free-choice violations: 0
    - S-Components
    - Wellstructuredness
      - PT-Handles: 0
      - TP-Handles: 0
  - Soundness
    - Workflow net property
      - Source places: 1
      - Sink places: 1
      - Transitions with emtpy preset: 0
      - Transitions with empty postset: 0
      - Connected components: 1
      - Strongly connected components: 1
    - Initial marking
      - Wrongly marked places: 0
    - Boundedness
      - Unbounded places: 0
    - Liveness
      - Dead transitions: 0
      - Non-live transitions: 0

Figure 18. Semantical analysis of student system.

4.2. Experimental results of the intelligent tutoring system

By following the procedure of Petri net verification, users can know how to operate this tutoring system correctly.

4.2.1. Teacher system operation

Figure 19 shows our teacher system with initial visual screen. Then, we click *Open File* to choose which one is the picture we need to show on the screen, as shown in Figure 20.

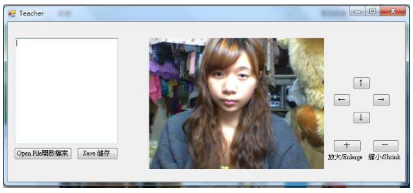


Figure 19. Initial teacher system.

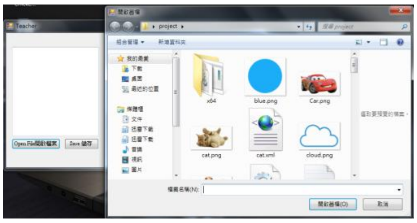


Figure 20. Teacher system-choose the picture.

Setting the position of the picture shown on the screen is to follow the center of a face, as shown in Figure 21. This function is completed by face recognition in an AR mark. The next step is to adjust the position we want. Let the *Cloud* on the top right-hand side of a face, as shown in Figure 22.

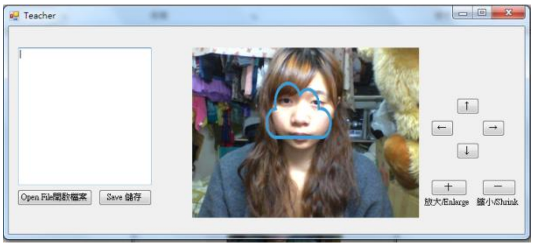


Figure 21. Teacher system-initial picture.

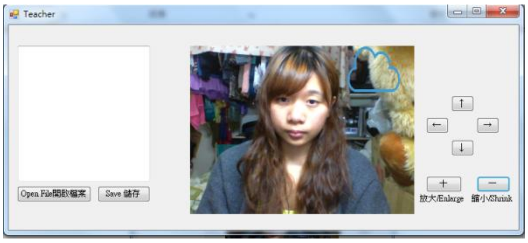


Figure 22. Teacher system-picture position adjusted.

Key-in the corresponding vocabulary in the *Text box*. For example, we key-in the *Cloud* inside, and click the *Save* button, as shown in Figure 23.

By following this procedure, we can add more vocabularies and pictures in the teacher system, as shown in Figure 24. If users finish the operating procedure, then the system is closed.

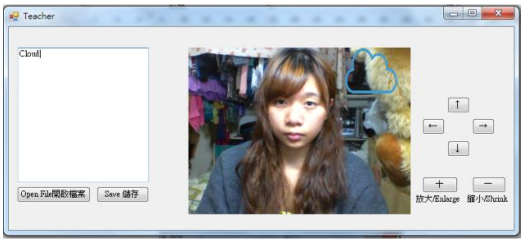

Cloud

Figure 23. Teacher system-key-in vocabulary and *Save*.

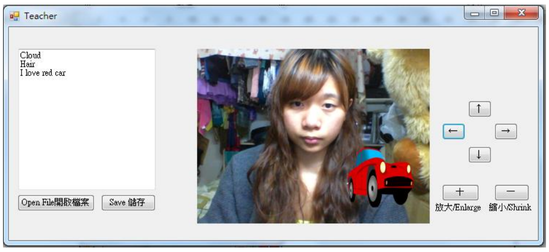

Cloud
Hair
I love red car

Figure 24. Teacher system-follow the procedure to key-in.

4.2.2. Student system operation

In the student system, we can just click the *Start* button. Figure 25 shows the student system with initial visual screen.

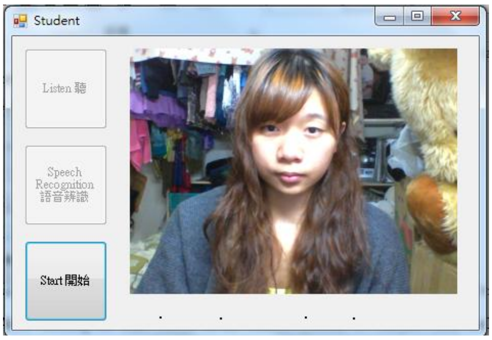


Figure 25. Initial student system.

After clicking the *Start* button in the student system, we can just click the *Listen* button to listen to the vocabulary based on the teacher system setting, as shown in Figure 26.

Users can listen repeatedly if they are not sure what the system is talking about until they can understand clearly, as shown in Figure 27. When users know what the vocabulary is, they can click the *Speech Recognition* button to speak. When the *Speak Now* button is clicked, users can speak out the vocabulary they heard, as shown in Figure 28.

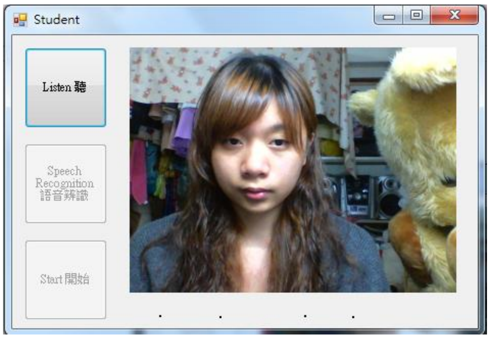


Figure 26. Student system - *Listen* clicked and listen.

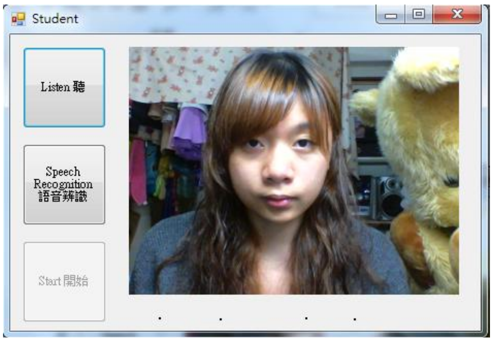


Figure 27. Student system-*Listen*.

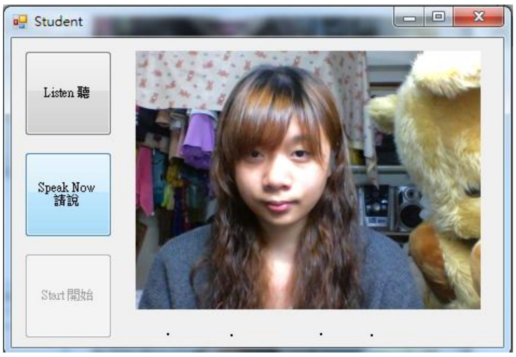


Figure 28. Student system-Speech recognition.

If users speak a correct vocabulary, the student system will show the *message box*, as shown in Figure 29. After clicking the *Sure* button, the picture is shown on the screen and the position and the big or small size is set in the teacher system, which is not changed in the student system, as shown in Figure 30. The total number of questions and those questions which users have already answered correctly are also shown on the underneath screen.

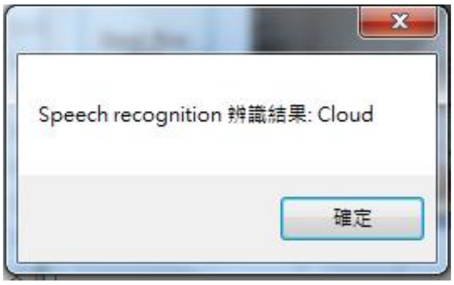
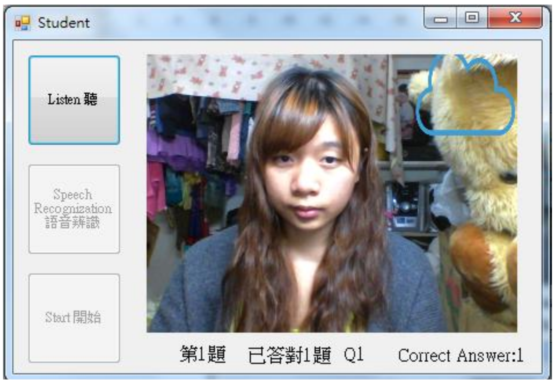

Speech recognition 辨識結果: Cloud

Figure 29. Student system-correct speech recognition.



第1題 已答對1題 Q1 Correct Answer:1

Figure 30. Student system-showing the picture.

In other cases, if the system can’t recognize users’ speaking, the system will not show anything on it, as shown in Figure 31.

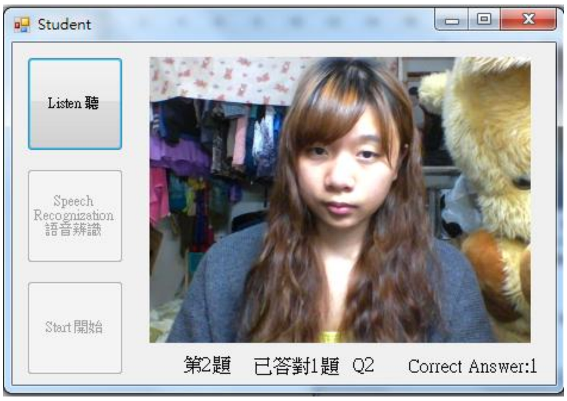


Figure 31. Student system-speech recognition error.

By following the operating procedure, we can do more and more exercises, as shown in Figure 32.

We can use the student system to give tests after the teacher taught in class. Let students practice English at home, or let the students with insufficient English ability have a good tutor at home. By using AR, TTS and speech recognition technology in this system will increase the student's interests in learning English and not shy to speak out English.

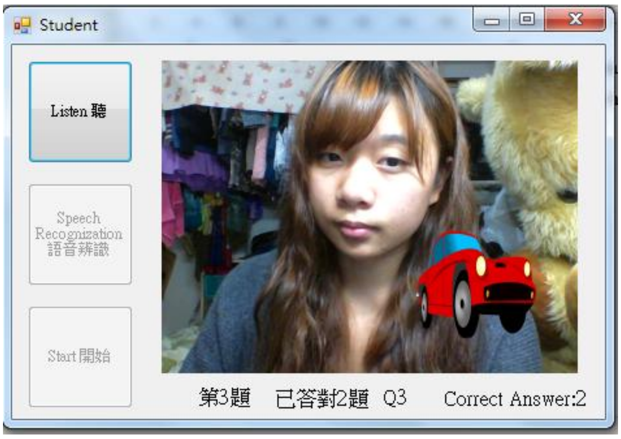


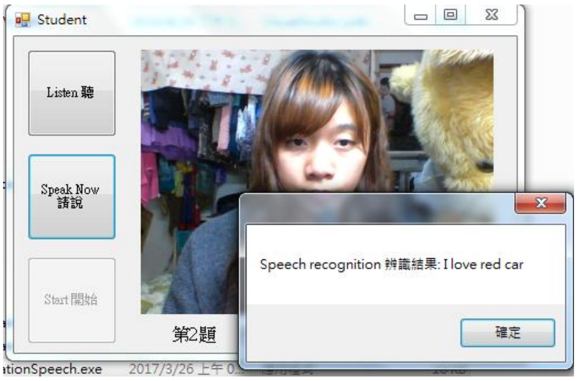
Figure 32. Student system-following the procedure.

4.2.3. Comparison with other existing systems

Comparing with other existing systems in [21–23], we can find they did not provide convenient usage for teachers and students because most of the systems have already set the vocabularies and made the QR code for the augmented reality of 3D shapes. This tutoring system is aimed to build a convenient environment for the users. Especially, the students' families cannot afford the high cost of Internet system. When teachers teach whenever they can key-in the vocabularies or short sentences immediately without making any 3D shapes first. When students want to learn the new vocabularies or the short sentences by using *teacher system* at home, they do not need to create the QR code and just need to download the picture they want and use the *student system* to practice. The most important difference is that other existing systems cannot use the short sentences to further practice but we can, as shown in Figure 33.

Table 6. The advantages and disadvantages of the systems compared.

	Advantages	Disadvantages
System 1 [21]	Has 3-D graphic function. Has the tutor by using 3-D Avatar onto the AR marker. Has the demo of playing multimedia in the AR classroom.	Needs to prepare the teaching aids (QR Code, *AR Classroom*). Users cannot create the questions by themselves. Does not provide a convenient environment for learning English whenever and wherever.
System 2 [22]	*TeachAR* platform has two ways: AR version and non-AR version. Using speech recognition helps users speak louder to practice English. Has interactive function in non-AR version.	Needs to prepare the teaching aids (QR Code, Marker groups). Users cannot create the questions by themselves.
System 3 [23]	Has two languages in the *Speaky Notes*. Users can create the questions by themselves. Combines the eBook that contains all the information about a single word and its marker image.	Needs to use the Internet. Users need to print out and make the cards with QR code by themselves. Does not have the speech recognition to check.
Proposed System	Provides a convenient environment for users. Not only building vocabulary but also practicing short sentences. Without any teaching aids. Users can create the questions by themselves. Using speech recognition helps users to speak out.	Does not have 3-D graphic function. Does not have multimedia playing in the AR. Does not have the explanation for the vocabulary.



Speech recognition 辨識結果: I love red car

Figure 33. Student system-use the short sentences to practice.

Table 7. Functional comparisons.

Function	System 1 [21]	System 2 [22]	System 3 [23]	Proposed System
AR technology	◎	◎	◎	◎
Speech recognition		◎		◎
3-D graphic function	◎	◎		
Convenient environment without using any teaching aids				◎
Building the vocabulary			◎	◎
Explanation for the vocabulary			◎	
Has the tutor	◎			

4.2.4. Learning results and students' feedback

Feedbacks on the satisfaction degree of this tutoring system are achieved by asking the students to fill in a questionnaire on the Google Form. Each student may complete the questionnaire once. Each question has ten options that score from one to ten, and there are 21 junior high school students invited to answer the questions below.

Q1-"What overall grade would you give to this tutoring system?" (80.95%)

Q2-"The tutoring system has improved my understanding of English."(82.85%)

Q3-"The tutoring system has helped my understanding of the correctly pronouncing English."(83.80%)

Q4-"The tutoring system has helped me increase the interest in learning English."(78.09%)

Q5-"The tutoring system has helped me learn English whenever I want."(81.42%)

Q6-"The tutoring system has provided me with a convenient environment without any teaching aids." (82.85%)

Q7-"The tutoring system has helped me learn English by myself without teachers." (87.38%)

Q8-"By using this tutoring system, I won't be shy to speak out English." (84.04%)

Q9-"By using this tutoring system, I can control the speed when I am learning English." (85.95%)

Q10-"By using this tutoring system, I love this novel way to learn English than the traditional way. " (80.47%)

Q11-"I will use this tutoring system to practice my English in the future." (78.09%)

Q12-"I will invite my friends to use this tutoring system." (81%)

According to the above results, if the satisfaction level 79% is set to be a successful criterion, then those items except Q4, Q11 have achieved the successful criteria.

5. Conclusion

This paper has proposed a novel method to verify the intelligent tutoring system by applying Petri nets. We first build a workflow using WoPeD, and apply two major Petri net analysis methods to verify the intelligent tutoring system. The first method is reachability trees and the second method is state equations, which are used in WoPeD to get the simulation results. We also build a tutoring system for teachers and students based on Augmented Reality (AR), Text-to-Speech (TTS), and Speech Recognition (SR). The contributions of this paper are presented as follows:

(1) Using reachability trees of Petri net can verify the tutoring system immediately.
(2) The tutoring system can be proved to have no deadlocks by using the state equations.
(3) The Petri net model built by a simulation tool, WoPeD, is analyzed to ensure that there is no overflow and deadlock problems caused in the tutoring system.
(4) By using WoPeD, we can build the tutoring system easier.

By following the students’ feedback, this tutoring system has shown to provide a convenient environment where students can practice English at home.

In the future, we will achieve the goal with a more powerful tutoring system in order to help students learn other languages.

Acknowledgments

The authors are very grateful to the anonymous reviewers for their constructive comments which have improved the quality of this paper. Also, this work was supported by the Ministry of Science and Technology, Taiwan, under grant MOST 107- 2221- E-845- 002-MY3.

Conflict of interest

The authors have declared that no conflict of interest in this paper.

References

1. H. K. Wu, S. W. Y. Lee, H. Y. Chang, et al., Current status, opportunities and challenges of augmented reality in education, *Comput. Educ.*, **62** (2013), 41–49.
2. Y. B. Kim and S. Y. Rhee, Augmented reality on a deformable surface by considering self-occlusion and realistic illu, *J. Internet Technol.*, **12** (2011), 733–740.
3. B. Koo and T. Shon, A structural health monitoring framework using 3D visualization and augmented reality in wireless sensor networks, *J. Internet Technol.*, **11** (2010), 801–807.
4. G. Schall, E. Mendez, E. Kruijff, et al., Handheld augmented reality for underground infrastructure visualization, *Pers. Ubiquit. Comput.*, **13** (2009), 281–291.
5. K. J. L. Nevelsteen, Virtual world, defined from a technological perspective and applied to video games, mixed reality, and the Metaverse, *Comput. Animat. Virt. W.*, **29** (2017).
6. J. Luo, Q. Zhang, X. Chen, et al., Modeling and race detection of ladder diagrams via ordinary Petri nets, *IEEE T. Syst., Man. Cy. S.*, **48** (2018), 1166–1176.
7. B. J. Boom, S. Orts-Escolano, X. X. Ning, et al., Interactive light source position estimation for augmented reality with an RGB-D camera, *Comput. Animat. Virt. W.*, **28** (2017).
8. P. C. Xiong, Y. S. Fan and M. C. Zhou, A Petri net approach to analysis and composition of Web services, *IEEE T. Syst. Man. Cy. A.*, **40** (2010), 376–387.
9. Y. Ru and C. N. Hadjicostis, Bounds on the number of markings consistent with label observations in Petri nets, *IEEE T. Autom. Sci. Eng.*, **6** (2009), 334–344.
10. J. Rosell, Assembly and task planning using Petri nets: A survey, *J. Eng. Manuf.*, **218** (2004), 987–994.
11. L. Li, Y. Ru and C. N. Hadjicostis, Least-cost firing sequence estimation in labeled Petri nets, *Process of the 45th IEEE International Conference on Decision and Control*, (2006), 416–421.

12. D. M. Xiang, G. J. Liu, C. G. Yan, et al., Detecting data-flow errors based on Petri nets with data operations, *IEEE/CAA JAS*, **5** (2018), 251–260.
13. T. Murata, Petri nets: properties, analysis, and applications, *IEEE*, **77** (1989), 541–580.
14. N. Wu and M. C. Zhou, System modeling and control with resource-oriented Petri nets, *Int. J. Prod. Res.*, **49** (2011), 6585–6586.
15. WoPeD, Mar. 2017. Available from: http://woped.dhbw-karlsruhe.de/woped/.
16. F. Suro and Y. Ono, Japanese EFL learners' uses of text-to-speech technology and their learning behaviors: A pilot study, *Process of the 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, (2016), 296–301.
17. Y. Lee, S. Lee and S. H. Lee, Multifinger interaction between remote users in avatar-mediated telepresence, *Comput. Animat. Virt. W.*, **28** (2017).
18. Speaker independent connected speech recognition-fifth generation computer corporation, May 2017. Available from: http://www.fifthgen.com/speaker-independent-connected-s-r.htm.
19. F. J. Yang, N. Q. Wu, Y. Qiao, et al., Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets, *IEEE/CAA JAS*, **5** (2018), 270–280.
20. P. Milgram and A. F. Kishino, Taxonomy of mixed reality visual displays, *IEICE T. Inform. Syst.*, (1994), 1321–1329.
21. K. C. Li, C. W. Tsai, C. T. Chen, et al., The design of immersive English learning environment using augmented reality, *Process of 8th IEEE International Conference on UMEDIA*, (2015), 174–179.
22. C. S. C. Dalim, A. Dey and T. Piumsomboon, TeachAR: An interactive augmented reality tool for teaching basic English to non-native children, *Process of IEEE International Symposium on Mixed and Augmented Reality Adjunct*, (2016), 67–72.
23. F. Sorrentino, L. D. Spano and R. Scateni, Speaky notes learn languages with augmented reality, *Process of IEEE International Conference Interactive Mobile Communication Technologies and Learning (IMCL)*, (2015), 56–61.