Technical Note

Adaptive Autonomy Expert System in Smart Grid based on Deterministic Timed Petri Nets

Mohammad Ali Zamani
CIPCE, School of ECE, University College of Engg.,
University of Tehran
Tehran, Iran
ma.zamani@ece.ut.ac.ir

Mohammad Amin Sharifi K.
CIPCE, School of ECE, University College of Engg.,
University of Tehran
Tehran, Iran
m.sharifi@ece.ut.ac.ir

Alireza Fereidunian
ECE Dep., Power and Water University of Technology
and
CIPCE, School of ECE, University of Tehran
Tehran, Iran
fereidunian@pwut.ac.ir; arfi@ece.ut.ac.ir

Hamid Lesani
CIPCE, School of ECE, University College of Engg.,
University of Tehran
Tehran, Iran
lesani@ut.ac.ir

Caro Lucas
CIPCE, School of ECE, University College of Eng.,
University of Tehran
Tehran, Iran
lucas@ut.ac.ir

Received: September 10, 2010 – Accepted: November 17, 2010

Abstract — Interaction of humans and computer agents should be harmonized by adapting the automation level of IT systems, to maintain a high performance for the system, in the changing environmental conditions. This research presents an expert system for realization of adaptive autonomy (AA), using deterministic timed Petri nets (DTPNs), referred to as AAPNES. The design is based on the practical list of environmental conditions and superior experts' judgments. As revealed by the results, the presented AAPNES can effectively determine the proper level of automation for the changing performance shaping factors of human-automation interaction systems in the smart grid.

Keywords- adaptive autonomy; expert system; human-computer interaction (HCI), human-automation interaction (HAI); petri net; power distribution automation; smart grid; level of automation (LOA); performance shaping factors (PSFs).

This paper is dedicated to the memory of the late Professor Caro Lucas, may rest in peace.

This paper is an extended version of “AAPNES: A Petri Net Expert System Realization of Adaptive Autonomy in Smart Grid,” In Proc. of IST2010, that is recommended by the committee to be considered for publication in IJICT.
1. INTRODUCTION

IT systems, as a vital part of the smart grid, are supposed to be intelligent, optimized, and adaptive [1]-[3]. They have enabled real-time monitoring, control, and automated operation of distribution networks.

Beside all of these characteristics, it should be considered that disregarding human factors in IT systems makes them more problematic than beneficial [4]. Therefore, IT systems should be designed with an orchestrated environment for humans and computers, in order to secure a best-fit for both humans and computers capabilities, advantages and disadvantages. In fact, the cooperation of humans and computers is more productive than either humans or computers working alone [5].

Reference [6] investigates the human-automation interaction strategy in manufacturing through five empirical cases. One of these cases is a manufacturing company in telecommunication industry in which wrong automation strategies (i.e. neglecting human-automation interaction issues) has led to outcomes below the expectations [6]. On the other hand, considering both human factors and computers simultaneously leads to a high level of complexity. In order to manage this complexity, great efforts are made under different names like: man (human)-machine systems, human-computer interaction (HCI), human-automation interaction (HAI) [7], and human-centered automation (HCA) [8], [9] (as indicated in [10]).

A simple form of the HAI model was first introduced by P.M. Fitts in 1951, where only two levels of automation (manual or automate) were considered [12]. Since this primary model was no longer successful in optimizing human-automation interaction; Sheridan and Verplank introduced a ten-degree level of automation (LOA) to overcome the deficiency of Fitts' two-degree model [7], [10], [13]-[15]. Afterwards, Parasuraman, Sheridan and Wickens [7] suggested the AA concept (also known as adaptive automation or adjustable automation [7], [10], [14]-[16]), which is expected to adapt the LOAs to the environmental conditions in order to optimize human-automation systems performance in different environmental conditions [7], [10]. Subsequently, Fereidunian et al. introduced a model-based framework for realization of the AA concept [17]-[19] and suggested expert systems for the realization of the adaptive autonomy concept [20]-[22].

Although considerable amount of research effort have been dedicated to this concept, still more investigations are required to implement the HAI and the AA concepts in industry and civil services [13], Excluding military and aerospace applications, [18] and [19] report the first implementations of the AA expert systems in the civil services. The capability of the simple model introduced in [18] and [19] in tracking and simulating human experts' opinion in complicated situations is partially acceptable.

Moreover, the data-oriented expert systems introduced in [22], [23], and [24], using logistic regression, generalized linear model, and hybrid techniques, respectively, require a large amount of data to determine the proper LOA. Also, the model-oriented expert systems of [20] and [21] can partially trace the changing environmental conditions in complicated situations and update the level of automation.

This article, as a continuum of a series, presenting a Petri net (PN) based expert system, introduces a method for realization of AA concept as a discrete-event system referred as Adaptive Autonomy Petri Net Expert System (AAPNES). Functional behavior of PNs simplifies modeling of experts’ judgments in form of IF-THEN rules. In this study, a novel method is presented for translating PSFs’ judgments to initial marking for execution of the Petri net models. Moreover, rule base extracted from the experts’ judgments is categorized into two types of rules, primary rules and veto rules. These two types of rules are modeled separately; and, consequently, a practical mechanism is introduced for connecting the separate levels of modeling. Furthermore, for the veto rules, a kind of priority is extracted by experts' judgments. This priority is employed in modeling using deterministic timed Petri nets’ (DTPN) characteristics.

The remainder of the paper is organized as follows: a brief background on Petri nets, the methodology of AAPNES realization, followed by implementation and results of the system. Afterwards, a discussion is presented, in order to investigate the performance of the proposed AAPNES.

II. PETRI NETS

This section is intended to briefly introduce the main concepts of Petri nets, in order to make this paper self-explanatory. We use notation, definition, and properties of Petri nets as given in [28].

A. Petri Net Definition

A Petri net is a particular sort of bipartite directed graph including places, transitions, and directed arcs. Directed arcs connect places to transitions or transitions to places. The dynamic behavior of a Petri net is shown by flow of tokens from some places to others by firing transitions.

A Petri net is formally defined as a 5-tuple \( N = (P, T, I, O, M_0) \), where

1. \( P = \{p_1, p_2, ..., p_n\} \) is a finite set of places;
2. \( T = \{t_1, t_2, ..., t_m\} \) is a finite set of transitions, \((P \cup T) \neq \emptyset, P \cap T = \emptyset\);
3. \( I: P \times T \to N \) is an input function that defines directed arcs from places to transitions, where \( N \) is a set of nonnegative integers;
4. \( O: T \times P \to N \) is an output function that defines directed arcs from transitions to places;
5. \( M: P \to N \) is a marking describing the distribution of tokens in places. The initial marking of net is indicated by \( M_0 \).

Downloaded from journal.itrc.ac.ir at 16:37 IRDT on Sunday June 14th 2020
marking of the net changes during the execution.

B. Petri Net Graph

A Petri net graph contains two types of nodes, circles and bars (boxes), representing places and transitions, respectively. Directed arcs (arrows), labeled with their multiplicity (weight), connect places and transitions. Dots residing in the circles represent tokens in places (see Fig. 1). For the Petri net of Fig. 1:

\[
\begin{align*}
\mathcal{P} &= \{p_1, p_2, p_3\};
\mathcal{T} = \{t_1, t_2\};
I(p_1, t_1) &= 1,
I(p_2, t_1) = 0, I(p_1, t_2) = 1, I(p_2, t_2) = 2,
I(p_3, t_1) = 0; O(t_1, p_2) = 1, O(t_2, p_3) = 1,
O(t_1, p_1) = 0; M_0 = (3, 2, 0)^T; (i, j = 1, 2)
\end{align*}
\]

C. Transition Firing

The execution of a Petri net is controlled by the number and distribution of tokens (see Fig. 2). Followings are enabling rule and firing rule of a transition, which control the flow of tokens in places:

1. Enabling rule: A transition \( t \) is enabled if \( \forall p \in \mathcal{P}: M(p) \geq I(p, t) \).

2. Firing rule: The firing of an enabled transition \( t \) removes from each input place \( p \) the Number of Tokens (NOT) equal to the Weight of Arc (WOA) connecting \( p \) to \( t \); and, deposits in each output place the NOT equal to the WOA connecting \( t \) to \( p \).

Mathematically, firing \( t \) at \( M \) yields a new marking \( M' \) determined by Eq. 2.

\[
\forall p \in \mathcal{P}: M'(p) = M(p) - I(p, t) + O(t, p)
\]

D. Reachability

A marking \( M_1 \) is said to be immediately reachable from \( M_0 \) if firing an enabled transition in \( M_0 \) results in \( M_1 \). Reachability is generalized in the way that a marking \( M_2 \) is said to be reachable from \( M_0 \) if firing a sequence of transitions in \( T \), starting from \( M_0 \), results in \( M_2 \). The set of all reachable markings of a graph \( G \) from initial marking \( M_0 \) is denoted by \( B(G, M_0) \) [29].

E. Incident Matrix and State Equation

The incidence matrix of a Petri net with \( m \) places and \( n \) transitions is \( A = [a_{i,j}]_{m \times n} \) with typical entry \( a_{i,j} = a_{i,j}^+ - a_{i,j}^- \) where \( a_{i,j}^+ = O(t_j, p_i) \) and \( a_{i,j}^- = I(p_j, t_i) \). According to firing rule, \( a_{i,j} \) represents the change in number of tokens in place \( p_j \) when transition \( t_i \) fires once.

Suppose \( M_k \) as a \( m \times 1 \) column vector which \( j^{th} \) entry denotes the NOT in place \( j \) immediately after the \( k^{th} \) firing in some firing sequence, and \( u_k \) as the \( k^{th} \)

firing vector with only one nonzero entry, a 1 in the \( i^{th} \) position for the \( i^{th} \) transition to be fired at the \( k^{th} \) firing.

The state equation for the Petri net is given in Eq. 3.

\[
M_k = M_{k-1} + A^T u_k; k = 1, 2, ...
\]

Now, suppose that the destination marking \( M_d \) is reachable from \( M_0 \) through a firing sequence \( \{u_1, u_2, ..., u_d\} \). The state equation can be generalized as Eq. 4.

\[
M_d = M_0 + \sum_{k=1}^{d} u_k
\]

F. Inhibitor Arc

An inhibitor arc connects an input place to a transition and changes the transition enabling condition in the way that there should be no tokens in each input place connected to the transition by an inhibitor arc (see Fig. 3).

G. Deterministic Timed Petri Net (DTPN)

Deterministic timed Petri net is a 6-tuple \((P, T, I, O, M_0, \tau)\), where \((P, T, I, O, M_0)\) is a Petri net, and \(\tau: T \rightarrow \mathbb{R}^+\) is a function that associates transitions with deterministic time delays. A transition \( t_i \) in a DTPN can fire at time \( \tau \) if and only if for any input place \( p \) of this transition, there have been the NOT equal to the WOA connecting \( p \) to \( t_i \) continuously for the time interval \([\tau - t_i, \tau]\), where \( t_i \) is the associated firing time of transition \( t_i \). After the transition fires, each of its output places, \( p_i \), will receive the NOT equal to the WOA connecting \( t_i \) to \( p_i \) at time \( \tau \) [28].

H. Petri Net Modeling of IF-THEN Rules

An IF-THEN rule can be modeled as a transition, in which, input places and output places represent antecedent portion and consequence portion of the
rule, respectively. This is performed as each proposition in antecedent portion is modeled as an input place and each proposition in consequence portion is modeled as an output place [30]. For instance, the following IF-THEN rule can be modeled as shown in Fig. 4.

\[ R_1: \text{if } (A \text{ or } B) \text{ and } C \text{ then } (D \text{ or } E) \text{ and } F \]  

(5)

Fig. 4. Petri net modeling of IF-THEN rule of Eq. (5)

III. METHODOLOGY

A. Problem Statement

The basic idea of this research is initiated in the Greater Tehran Electricity Distribution Company (GTEDC), from which, the practical list of Performance Shaping Factors (PSFs) – suggested to represent the environmental conditions which affect the performance of HAI system – were obtained. Also, the experts’ judgments interviews were performed in the GTEDC.

The smart grid considered in this paper is at the distribution level, which needs automation, advanced IT infrastructure for data collection, energy efficiency and better integration of distributed generation [31]. Therefore a smart grid’s goal is to integrate advanced sensing technologies, control methods, and integrated communication into the current electricity grid.

The Greater Tehran Electrical Distribution Company (GTEDC) operates the distribution network of the Greater Tehran, the main part of the Iranian capital city. It delivers electric power to the Greater Tehran metropolitan area, feeding more than 12,000 medium voltage (20 kV / 400 V) substations [19]. It is the largest electric utility in the country, providing almost 16641 [GWh / year], 32,490,000 customers, and 13,300 MV substations [19]. Like many other utilities, the GTEDC is continually making an effort to improve modeling and estimation methods [32].

The GTEDC is implementing some pilot projects to transfer the current grid to the smart grid such as MV substations automation to improve the Utility Management Automation (UMA) of the company.

Utility Management Automation (UMA), as a subsystem of smart grid [24], acts as a SCADA (supervisory control and data acquisition) system for the electric utility, in which, human operators and automation systems work collaboratively. In this research, an expert system (referred to as AAPNES) is used to adapt the autonomy level (LOA) of the UMA system to the changes in PSFs. In other words, the AAPNES controls the LOA of the UMA system.

AAPNES is implemented to one of the power distribution automation functions, referred to as feeder reconfiguration function of utility management automation (UMA-DRF). The UMA-DRF system (which has been introduced in [33], [34], and [35]), automatically restores the electric energy for the affected customers (electric power delivery load points), by reconfiguring the distribution network topology, after a failure in the distribution network. Fig. 5 shows the proposed expert system role in relation with the other subsystems of the UMA. The dashed arrow from the UMA conveys the PSFs to the AAPNES; where, the other solid line arrows command the LOA that is recommended by the AAPNES to the UMA.

The AAES framework, Adaptive Autonomy, PSF and LOA concepts are completely introduced in [17]-[19].

![Fig. 5. Position of Petri Net Adaptive Autonomy Expert System in Power Distribution System](image)

B. IF-THEN Rules for AAPNES

IF-THEN rules and representation of them in Petri net are the primary concerns on realization of the AAPNES. In this part, the general form of the extracted rules from the experts’ judgment is presented.

There are two sorts of rules that are applied in the AAPNES. Primary Rules and Veto Rules. The primary rules suggest an LOA when only one PSF changes from its normal condition. The PSFs’ values are described in [18], [19], and [21]. The primary IF-THEN rules are (Eq. 6):

\[ R_1: \text{if } PSF_1 = PSF_1, \ldots, PSF_i = PSF_i, \ldots, \text{ and } PSF_j = PSF_j, \ldots, \text{ and } PSF_n = PSF_n, \text{ then } LOA = L_1 \]

\[ R_2: \text{if } PSF_1 = PSF_1, \ldots, PSF_i = PSF_i, \ldots, \text{ and } PSF_j = PSF_j, \ldots, \text{ and } PSF_n = PSF_n, \text{ then } LOA = L_2 \]

\[ R_3: \text{if } PSF_1 = PSF_1, \ldots, PSF_i = PSF_i, \ldots, \text{ and } PSF_j = PSF_j, \ldots, \text{ and } PSF_n = PSF_n, \text{ then } LOA = L_3 \]

\[ R_4: \text{if } PSF_1 = PSF_1, \ldots, PSF_i = PSF_i, \ldots, \text{ and } PSF_j = PSF_j, \ldots, \text{ and } PSF_n = PSF_n, \text{ then } LOA = L_4 \]

\[ R_5: \text{if } PSF_1 = PSF_1, \ldots, PSF_i = PSF_i, \ldots, \text{ and } PSF_j = PSF_j, \ldots, \text{ and } PSF_n = PSF_n, \text{ then } LOA = L_5 \]

\[ R_6: \text{if } PSF_1 = PSF_1, \ldots, PSF_i = PSF_i, \ldots, \text{ and } PSF_j = PSF_j, \ldots, \text{ and } PSF_n = PSF_n, \text{ then } LOA = L_6 \]

where $PSF_j$ is the $j$th PSF, $PSF_i^j$ is the $i$th value of $j$th PSF, $L_i$ is LOA value of rule $i$ ($R_i$), and $n$ is the number of PSFs which is 6 in our application according to GTEDC’s experts [20].

The veto rules present a kind of authority in some PSFs (with particular PSFs’ value) that can change...
LOA without considering other PSFs’ values. In other words, some PSFs could veto other PSFs. For example, according to the experts opinion, if 10 faults occur in two hours (PSF4 = 10 faults in two hours), then the LOA will be 7 whatever other PSFs are. Additionally, experts’ judgments introduce a kind of priority in veto rules, that is, veto rules should be applied in order.

IV. REALIZATION OF AAPNES

In this part the rules from previous part are customized to be applied in the AAPNES. AAPNES is the final expert system that recommends the proper LOA in presence of different PSFs. Fig. 6 shows the overall design of the expert system. As shown, the proposed expert system includes two main levels of Petri net modeling: Primary level and Veto level, corresponding to primary rules and veto rules, respectively. For the systems to operate, input PSFs, values are transformed to an initial marking for the primary level PN. Then, using reachability analysis, maximum, minimum, and average LOAs are determined from outputs of primary level PN. Afterwards, the maximum, minimum, and average LOAs from primary level are applied as inputs (initial marking) to veto level PN and combined using veto rules, in order to determine the final hybrid LOA.

In the remainder of this section, different layers of the expert system are presented in details.

![Fig. 6. Layers of expert system in determining proper LOA from input PSFs](image)

A. Primary Level Petri Net of Modeling

The primary level of modeling is constructed using primary rules from rule base (Eq. 6). To do so, each rule of Eq. 6 is modeled by a transition and each PSF is modeled by an input place for the transition. Then, each proposition \( PSF_i = PSF^x \) in antecedent is transformed to a weight for the arc (WOA) connecting the input place corresponding to \( PSF_i \) to the transition. Table 1 shows the corresponding weight of arcs (WOAs) for different PSFs’ values for each PSF. As shown, based on the effect of PSFs’ values on LOA given by experts’ opinion, WOAs are determined in a way that for each PSF, the PSF’s normal value (indicated by * in Table 1) gets WOA = 1 and for other PSF’s values, WOA increases as the abnormality of PSF’s value, compared to the normal PSF’s value, increases.

<table>
<thead>
<tr>
<th>PSF</th>
<th>PSF’s Value</th>
<th>WOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Night</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Service area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Un-crowded urban</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Crowded urban</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Rural</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Customer type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial/Industrial</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>VIP</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Number of faults per two hours</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Few*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>More</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Much more</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Network age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle-aged</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Old</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low*</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

For the consequent portion of the rule, an output place is considered for the level of automation suggested by the rule, and the suggested value of LOA determines the WOA between the transition and the output place. For instance, the third rule in the rule base is

R3: If \((PSF_1 = Day) \& (PSF_2 = Crowded urban) \& (PSF_3 = Residential) \& (PSF_4 = Few) \& (PSF_5 = New) \& (PSF_6 = Low))\) then \((LOA = 6)\)

that is modeled as shown in Fig. 7.

![Fig. 7. Rule to transition modeling method PN graph of elementary modeling](image)

Following the same procedure for other rules leads to the complete PN model of primary rules. Fig. 8 shows the primary level PN model.

To execute primary level, the input condition (PSF vector) is transformed to an initial marking according to Table 1, i.e. the same mechanism for determining WOAs from rules holds for determining NOTs in initial marking from input PSF vector. This is because each rule is actually based on an input PSF vector and its corresponding LOA. Then, using reachability analysis, all immediate markings from initial marking...
are calculated. The reason that we only consider immediate markings is that only these markings are realistic, i.e. those markings which are result of firing sequences including two or more transitions are not reliable due to ending in an unacceptable LOA. Finally, based on NOT in output place (M (LOA)) in all immediate markings from initial marking, the maximum, minimum, and average LOAs for corresponding input condition are determined. In other words, in primary level of modeling, at first, the effect of each PSF is determined individually to obtain a set of acceptable LOAs for given input PSF vector; then, these effects are combined and projected in three values of maximum, minimum, and average LOAs from derived acceptable LOAs in PN model.

![PN graph of primary modeling](image)

The most important characteristic of primary level of modeling is that the transformation of PSFs’ values to WOAs is in a way that in PN model of Fig. 5 some transitions include some others; that is; enabling conditions of some transitions include that of some other transitions. For instance, the input PSFs of ((PSF1 = Night) & (PSF2 = Crowded urban) & (PSF3 =Residential) & (PSF4 = Few) & (PSF5 = Few) & (PSF6 = Low)) corresponds to initial marking of (2,1,1,1,1,1). This initial marking enables not only transition R2 (the rule that corresponds to the given input PSFs), but also transition R1 (the rule corresponding to normal condition). Consequently, for any input PSFs deviated from normal values, all transitions corresponding to states between normal condition and the given input PSFs are enabled.

B. Veto Level Petri Net of Modeling

The veto level of modeling is constructed using veto rules from rule base. Applying veto rules to primary modeling, with the same mechanism as primary rules, weakens performance of the expert system. This is due to the fact that primary rules and veto rules provide different types of information; while primary rules determine the effect of each PSF’s value on the LOA, veto rules include special conditions where some PSFs’ values veto other PSFs’ values and determine the final LOA. Moreover, in modeling method, transformation of primary rules to transitions and input PSFs to initial marking is in a way that WOAs in transitions and NOTs in initial marking, both equal at least one; while, veto rules are independent from some PSFs, i.e. WOAs from corresponding input places are zero; thus, the initial marking from input PSFs is not applicable to veto rules.

The main structure of veto level of modeling is based on the essential characteristic of primary modeling. In primary level when a transition is enabled, all of its sub-transitions are enabled, too (transition i is a sub-transition of transition j if $WOA_{i,k} \leq WOA_{j,k}$ for $k=1,2,\ldots, 6$, where $WOA_{m,n}$ is WOA from input place corresponding to PSF, to transition m). Consequently, it can be driven that when minimum LOA form primary model equals 5, the final LOA equals maximum LOA; vice versa, when maximum LOA form primary model equals 5, the final LOA equals minimum LOA. In complicated situations, when minimum and maximum LOAs are not enough to determine the final LOA, average LOA is considered.

In order to use veto rules in veto level of modeling, we translated them into the rules applicable to results of primary modeling – maximum, minimum and average LOAs; that is, the authority given by veto rules to some PSFs’ values is transformed to the authority of some combinations of maximum, minimum and average LOAs over other combinations. This authority is modeled as priority of some transitions over others using DTPNs and inhibitor arcs for modeling (one of the main challenges of our work has been developing a method that includes tools for modeling the priority introduced by the system experts). Fig. 9 shows the veto level PN. As shown in Fig. 9, the net is a DTPN including inhibitor arcs.

![DTPN graph of veto modeling](image)

V. RESULTS AND DISCUSSIONS

Intelligence of an expert system is extensively based on including appropriate rules. Therefore, the rule base of the proposed expert system is based on superior experts’ judgments. The superior experts are experts whose superiority (in higher and more reliable expertise) has been verified according to consistency for their expert judgments [20].

In this section, the results of implementation of expert system proposed in previous part are presented;
and, also, different aspects of obtained results are discussed. Afterwards, the performance of presented AAPNES is compared with other expert systems proposed in our previous researches.

In order to evaluate the performance of the expert system, it is checked whether the system can simulate an expert opinion. To do so, the test sets are asked from a superior expert in various PSFs combinations. All feasible conditions include 324 states which are used to determine Correct Classification Rate (CCR) of the system.

Table II shows the CCR of the proposed AAPNES. As expected, it was observed that the maximum and minimum LOAs can mostly determine the proper LOA for $LOA = 7$ and $LOA = 3$, respectively; and, they show very weak performance for other levels of automation. Also, as shown in Table II, applying veto level of modeling on results of primary model highly improves the performance of the expert system.

**TABLE II. CCR OF DIFFERENT MODELING METHODS**

<table>
<thead>
<tr>
<th>Method</th>
<th>CCR %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary modeling</td>
<td>38.6</td>
</tr>
<tr>
<td>Minimum LOA</td>
<td>6.8</td>
</tr>
<tr>
<td>Average LOA</td>
<td>38.3</td>
</tr>
<tr>
<td>Veto Modeling</td>
<td>72.8</td>
</tr>
</tbody>
</table>

Moreover, veto level of modeling elevates the performance of expert system for $LOA = 3$ or 7. CCR for these levels of automation is 90%, which demonstrates very high performance of expert system for critical situations. On the other hand, this result depicts the weak performance of the system for mediate conditions. This phenomenon was also seen in experts during interviews; while they could easily determine the proper LOA for boundary conditions ($LOA = 3$ or 7), for complicated combination of input PSFs with both increasing and decreasing factors, they needed more time in order to analyze and determine the proper LOA. This may be due to the fact that the presented expert system introduces a new method for realization of AA as a discrete-event systems using PN modeling techniques which is more like humanistic decision making in terms of hierarchical reasoning. To illustrate, the presented AAPNES first determines the effect of PSFs separately, then combines them to determine to proper LOA.

In comparison with model-driven expert systems presented in [20] and [21], the introduced AAPNES in this article has the unique quality of employing the priority introduced in experts’ judgments in modeling. Moreover, although, expert systems of [20] and [21] can track the human expert opinion in simple combination of environmental conditions, they fail in determining proper LOA for complicated situations which leads to weak CCR. Furthermore, the expert systems of [22], [23], and [24] may show higher performance regarding CCR; however, they are all data-driven and require a large amount of data to be able to determine to proper LOA; while the proposed AAPNES is based on twenty and two general rules.

Although the presented AAPNES is successful in tracking the human experts’ opinion, it needs a medium between two levels of modeling in order to introduce the output of primary model as input to veto model. In other words, the design of presented expert system is not integrated and contains two separate levels. This problem can be solved using more detailed modeling technique in order to achieve an integrated system.

**VI. CONCLUSIONS**

An expert system was introduced for realization of the adaptive autonomy framework of [18] and [19], referred to as AAPNES. The presented AAPNES adapts the level of automation of UMA (a part of the smart grid) to the environmental conditions. The judgments of GTEDC’s experts were developed as a subjective rule base for the AAPNES.

The proposed method uses the functional characteristics of Petri nets in order to model IF-THEN rules extracted from experts’ judgments. This modeling method is applicable to input-output systems described by IF-THEN rules – inputs and outputs of these systems should have discrete values. Note that modeling details can change based on the nature of the system (describing IF-THEN rules) and designer objectives.

Simulation results illustrated that the presented AAPNES is successful in adapting the level of automation (LOA) to the environmental conditions of the automation system.

This research will be continued by introducing a more effective modeling technique.

**ACKNOWLEDGMENT**

Many thanks to the SCADA experts in the Greater Tehran Electricity Distribution Company (GTEDC), for the time they afford for interviews and to fill in our questionnaires.

Special appreciation goes to GTEDC managing board, especially to: Mr. Habib Torabi (Deputy CEO), Mr. Payam Kharazmi (CIS/CRM Manager) and Mr. A. Kazemzakhshi (Dispatching and Operation Manager) for their helpful practical assistance in preparing the experimental data for this research.

**REFERENCES**

grid/docs/At%20Vision%20for%20the%20Modern%20Grid_Final_v1_0.pdf


Alireza Fereidunian received his Ph.D. & M.Sc. from University of Tehran (School of ECE), in 2009 and 1997; where he is a Post-Doctoral Research Fellow now. Dr. Fereidunian is a faculty member at Power and Water University of Technology and serves as an independent consultant to GTEDC. His research interests include power systems automation, and application of intelligent systems, human-automation interaction, data mining, decision-support, decision-making, IT and signal processing in power systems. He is a member of IEEE (and IEEE-SMC-HCI TC member) and INCOSE (as INCOSE Iran point of contact).

Hamid Lesani is a Professor at the Center of Excellence for Control and Intelligent Processing (CIPCE), School of Electrical and Computer Engineering, University of Tehran. He received the M.S. degree in electrical power engineering from the University of Tehran, Iran, in 1975, and the Ph.D. degree in electrical engineering from the University of Dundee, U.K., in 1987. Early in his career, he served as a Faculty Member with Mazandaran University. After obtaining the Ph.D. degree, he joined the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran. His teaching and research interests are design and modeling of electrical machines and power systems. Professor Lesani is a member of IEEE (PES) and IEEE Iran Section.

Caro Lucas was a Professor and a member (as well as the Founder-Director) of Center of Excellence for Control and Intelligent Processing (CIPCE), School of Electrical and Computer Engineering, University of Tehran, as well as a Researcher at the School of Cognitive Sciences (SCS), Institute for Studies in Theoretical Physics and Mathematics (IPM). His research interests include uncertain systems, intelligent control, multi-agent systems, data mining, business intelligence, financial modeling, biological computing, computational intelligence, neural networks, and knowledge management. He was the founder of the ISRF, IPM and has assisted in founding several new research organizations and engineering disciplines in Iran. He is the holder of the patent for Speaker Independent Farsi Isolated Word Neurorrecognizer. Dr. Lucas has served as Chairman of the IEEE, Iran Section (1990-1992) and as the Chairman of several international conferences. Professor Lucas has been nominated as a member of Long-lasting Figures of Science and Culture Hall of Fame by the Iranian Academy of Engineering in 2006. Regretfully, the late Professor Caro Lucas passed away on July 8, 2010.